

LNCS 7564

Agostino Cortesi
Nabendu Chaki
Khalid Saeed
Sławomir Wierzchoń (Eds.)

Computer Information Systems and Industrial Management

11th IFIP TC 8 International Conference, CISIM 2012
Venice, Italy, September 2012
Proceedings



ifip



Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Agostino Cortesi Nabendu Chaki
Khalid Saeed Sławomir Wierzchoń (Eds.)

Computer Information Systems and Industrial Management

11th IFIP TC 8 International Conference, CISIM 2012
Venice, Italy, September 26-28, 2012
Proceedings



Springer

Volume Editors

Agostino Cortesi
University Ca' Foscari of Venice, DAIS
via Torino 155, 30170 Venice, Italy
E-mail: cortesi@unive.it

Nabendu Chaki
University of Calcutta
Department of Computer Science and Engineering
92 APC Road, 700009 Kolkata, India
E-mail: nabendu@ieee.org

Khalid Saeed
AGH University of Science and Technology
Faculty of Physics and Applied Computer Science
Al. Mickiewicza 30, 30-059, Cracow, Poland
E-mail: saeed@agh.edu.pl

Sławomir Wierzchoń
Polish Academy of Sciences
Department of Artificial Intelligence
Jana Kazimierza 5, 01-248 Warsaw, Poland
E-mail: s.wierzchon@ipipan.waw.pl

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-33259-3 e-ISBN 978-3-642-33260-9
DOI 10.1007/978-3-642-33260-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012946037

CR Subject Classification (1998): H.4, C.2, H.3, I.2, D.2, C.2.4, K.6.5

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

© IFIP International Federation for Information Processing 2012

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains the papers presented at CISIM 2012, the 11th International Conference on Computer Information Systems and Industrial Management held during September 26–28, 2012, in Venice.

CISIM provides a forum for researchers from all over the world to discuss effective software support for widespread use of information systems technologies. The main focus of this edition was on data management in (possibly) untrusted networks, addressing the issue of securing computer networks so as to ensure reliable data management and delivery. The conference is supported by IFIP TC8 Information Systems.

Topics covered by CISIM include network and application security models and tools, security requirements definition and modelling, formal verification of security-relevant properties, security testing of legacy systems, data analysis, biometric security, advanced biosignal processing and modelling, biometric performance management, classification and indexing of multimedia information objects, multimedia security and protection, access control and data protection, Web security, security of SaaS in cloud computing environments, software engineering for cloud and ubiquitous computing, business process engineering and execution support, data protection in ERP systems, industrial applications: government, finance, retail, etc.

This year, 80 papers were submitted to CISIM. The volume contains 35 papers selected by the Program Committee based on anonymous reviews and discussions through EasyChair. The main selection criteria were relevance and quality. Every paper was reviewed by two to five reviewers, and the articles presented in this volume were deeply improved based on the reviewers' comments.

The CISIM program included three keynote lectures by Roy Maxion (Carnegie Mellon University, USA), Pierpaolo Degano (University of Pisa, Italy), and Young Im Cho (University of Suwon, Korea).

We would like to thank all the members of the Program Committee, and the external reviewers for their dedicated effort in the paper selection process. We thank also the Honorary Chair of the conference, Ryszard Tadeusiewicz, and the Organizing Committee Chair, Andrea Marin.

We thank SAP, our industrial sponsor; the Venice chapter of ISACA; the DAIS Department of Ca' Foscari University; the University of Calcutta; AGH Krakow; the Polish Academy of Sciences; AICA; Kairos Consulting; and Venezia Congressi.

We are also grateful to Andrei Voronkov, whose EasyChair system eased the submission and selection process, and greatly supported the compilation of the proceedings.

July 2012

Agostino Cortesi
Nabendu Chaki
Khalid Saeed
Sławomir Wierzchoń

Organization

Program Committee

Raid Al-Tahir	University of the West Indies
Adrian Atanasiu	University of Bucharest, Romania
Aditya Bagchi	Indian Statistical Institute, India
Rahma Boucetta	University of Sfax, Tunisia
Silvana Castano	University of Milan, Italy
Nabendu Chaki	University of Calcutta, India
Rituparna Chaki	West Bengal University of Technology, India
Young Im Cho	The University of Suwon, Korea
Sankhayan Choudhury	University of Calcutta, India
Agostino Cortesi	Università Ca' Foscari Venezia, Italy
Dipankar Dasgupta	The University of Memphis, USA
Pierpaolo Degano	Università di Pisa, Italy
David Feng	University of Sydney, Australia
Pietro Ferrara	ETH Zurich, Switzerland
Riccardo Focardi	Università Ca' Foscari Venezia, Italy
Aditya Ghose	University of Wollongong, Australia
Kaoru Hirota	Tokyo Institute of Technology, Japan
Sushil Jajodia	George Mason University, USA
Khalide Jbilou	Université du Littoral Côte d'Opale, France
Dong Hwa Kim	Hanbat National University, Korea
Debajyoti Mukhopadhyay	Maharastra Institute of Technology
Yuko Murayama	Iwate Prefectural University, Japan
Nishiuchi Nobuyuki	Tokyo Metropolitan University, Japan
Isabelle Perseil	Inserm, France
Marco Pistoia	IBM T.J. Watson Research Center, USA
Khalid Saeed	AGH University of Science and Technology, Krakow, Poland
Vaclav Snasel	VSB-Technical University of Ostrava, Czech Republic
Bernhard Steffen	University of Dortmund, Germany
Giancarlo Succi	Free University of Bolzano/Bozen, Italy
Ryszard Tadeusiewicz	AGH University of Science and Technology, Krakow, Poland
Heinrich Voss	Hamburg University of Technology, Germany
Slawomir Wierchoń	Polish Academy of Sciences, Poland

Additional Reviewers

Adamski, Marcin
Albanese, Massimiliano
Albarelli, Andrea
Almasi, Adela
Baranga, Andrei
Bergamasco, Filippo
Bhattacharjee, Debotosh
Bodei, Chiara
Bolosteanu, Iulia
Cai, Weidong
Chakrabarti, Amlan
Chanda, Bhabatosh
Constantinescu, Liviu
Costantini, Giulia
De Benedictis, Alessandra
Dinu, Liviu P.
Ferrari, Gian-Luigi
Grossi, Roberto
Hashizume, Ayako
Hristea, Florentina
Khodaei, Katayoun
Le, Meixing

Luccio, Fabrizio
Marin, Andrea
Montangero, Carlo
Morogan, Luciana
Mukherjee, Dipti Prasad
Murthy, C.A.
Olimid, Ruxandra
Orlando, Salvatore
Paraschiv-Munteanu, Iuliana
Rossi, Sabina
Roy, Samir
Rybnik, Mariusz
Sarkar, Anirban
Sengupta, Sabnam
Simion, Emil
Tabedzki, Marek
Tataram, Monica
Togan, Mihai
Torsello, Andrea
Xia, Yong
Zhang, Lei

Sponsors



Table of Contents

Invited Talks

Formalising Security in Ubiquitous and Cloud Scenarios	1
<i>Chiara Bodei, Pierpaolo Degano, Gian-Luigi Ferrari, Letterio Galletta, and Gianluca Mezzetti</i>	
Designing Smart Cities: Security Issues	30
<i>Young Im Cho</i>	

Security, Access Control and Intrusion Detection

Certificate-Based Encryption Scheme with General Access Structure . . .	41
<i>Tomasz Hyla and Jerzy Pejaś</i>	
Security Margin Evaluation of SHA-3 Contest Finalists through SAT-Based Attacks	56
<i>Ekawat Homsirikamol, Paweł Morawiecki, Marcin Rogawski, and Marian Srebrny</i>	
Usage Control Model Specification in XACML Policy Language: XACML Policy Engine of UCON	68
<i>Um-e-Ghazia, Rahat Masood, Muhammad Awais Shibli, and Muhammad Bilal</i>	
TIDS: Trust-based Intrusion Detection System for Wireless Ad-hoc Networks	80
<i>Novarun Deb and Nabendu Chaki</i>	
Intruder Data Classification Using GM-SOM	92
<i>Petr Gajdoš and Pavel Moravec</i>	
Method for Identification of Suitable Persons in Collaborators' Networks	101
<i>Pavla Dráždilová, Alisa Babskova, Jan Martinovič, Kateřina Slaninová, and Štěpán Minks</i>	
A Graph-Based Formalism for Controlling Access to a Digital Library Ontology	111
<i>Subhasis Dasgupta and Aditya Bagchi</i>	
Role Approach in Access Control Development with the Usage Control Concept	123
<i>Aneta Poniszewska-Maranda</i>	

Pattern Recognition and Image Processing

A New Algorithm for Rotation Detection in Iris Pattern Recognition ...	135
<i>Krzysztof Misztal, Jacek Tabor, and Khalid Saeed</i>	
Outlier Removal in 2D Leap Frog Algorithm	146
<i>Ryszard Kozera and Jacek Tchórzewski</i>	
Dynamic Signature Recognition Based on Modified Windows Technique	158
<i>Rafał Doroz and Krzysztof Wróbel</i>	
Rigid and Non-rigid Shape Matching for Mechanical Components Retrieval	168
<i>Andrea Albarelli, Filippo Bergamasco, and Andrea Torsello</i>	
Embedding of the Extended Euclidean Distance into Pattern Recognition with Higher-Order Singular Value Decomposition of Prototype Tensors	180
<i>Bogusław Cyganek</i>	

Biometric Applications

DTW and Voting-Based Lip Print Recognition System	191
<i>Piotr Porwik and Tomasz Orczyk</i>	
Advances in the Keystroke Dynamics: The Practical Impact of Database Quality	203
<i>Mariusz Rybniak, Piotr Panasiuk, Khalid Saeed, and Marcin Rogowski</i>	
Advanced Intracardiac Biosignal Processing	215
<i>Marek Penhaker, Petr Klimes, Jakub Pindor, and David Korpas</i>	
Multi-constraints Face Detect-Track System	224
<i>Hazar Mliki, Mohamed Hammami, and Hanène Ben-Abdallah</i>	
Using a Differential Pressure Sensor as Spirometer	236
<i>Martin Augustynek, Ondrej Adamec, and David Micanik</i>	

Algorithms and Data Management

Hybrid Negative Selection Approach for Anomaly Detection	242
<i>Andrzej Chmielewski and Sławomir T. Wierchoń</i>	
Spectral Clustering Based on k -Nearest Neighbor Graph	254
<i>Małgorzata Lucińska and Sławomir T. Wierchoń</i>	
A New Scale for Attribute Dependency in Large Database Systems	266
<i>Soumya Sen, Anjan Dutta, Agostino Cortesi, and Nabendu Chaki</i>	

Left-Right Oscillate Algorithm for Community Detection Used in E-Learning System	278
<i>Jan Martinovič, Pavla Dráždilová, Kateřina Slaninová, Tomáš Kocyan, and Václav Snášel</i>	
Plan and Goal Structure Reconstruction: An Automated and Incremental Method Based on Observation of a Single Agent	290
<i>Bartłomiej Józef Dzieńkowski and Urszula Markowska-Kaczmar</i>	
On Spectral Partitioning of Co-authorship Networks	302
<i>Václav Snášel, Pavel Krömer, Jan Platoš, Miloš Kudělka, and Zdeněk Horák</i>	
An Efficient Binary Playfair Algorithm Using a 4×4 Playfair Key Matrix	314
<i>Saswati Mukherjee, Matangini Chattopadhyay, Ayan Lahiri, and Samiran Chattopadhyay</i>	
Tuning of a Knowledge-Driven Harmonization Model for Tonal Music	326
<i>Mariusz Rybnik and Wladyslaw Homenda</i>	
Efficient Processing the Braille Music Notation	338
<i>Tomasz Sitarek and Wladyslaw Homenda</i>	

Networking

ETSeM: A Energy-Aware, Trust-Based, Selective Multi-path Routing Protocol	351
<i>Manali Chakraborty and Nabendu Chaki</i>	
Weighted Energy Efficient Cluster Based Routing for Wireless Sensor Networks	361
<i>Soumyabrata Saha and Rituparna Chaki</i>	

System Models and Risk Assessment

A Business Process Modeling Notation Extension for Risk Handling	374
<i>Bartosz Marcinkowski and Michal Kuciapski</i>	
Modeling Consumer Decision Making Process with Triangular Norms . . .	382
<i>Agnieszka Jastrzebska and Wladyslaw Homenda</i>	
Neural Network Modeling of a Flexible Manipulator Robot	395
<i>Rahma Boucetta and Mohamed Naceur Abdelkrim</i>	
P Systems for Traffic Flow Simulation	405
<i>Jiří Dvorský, Zbyněk Janoška, and Lukáš Vojáček</i>	

Using Extended Raster File for Real Time Traffic Information Mining 416
 Michal Radecký, Jan Martinovič, Dušan Fedorčák, Radek Tomis, and Ivo Vondrák

A Comparative Study of Risk Assessment Methods, MEHARI & CRAMM with a New Formal Model of Risk Assessment (FoMRA) in Information Systems 428
 Imed El Fray

Author Index 443

Formalising Security in Ubiquitous and Cloud Scenarios^{*}

Chiara Bodei, Pierpaolo Degano, Gian-Luigi Ferrari,
Letterio Galletta, and Gianluca Mezzetti

Dipartimento di Informatica, Università di Pisa
{chiara,degano,giangi,galletta,mezzetti}@di.unipi.it

Abstract. We survey some critical issues arising in the ubiquitous computing paradigm, in particular the interplay between context-awareness and security. We then overview a language-based approach that addresses these problems from the point of view of Formal Methods. More precisely, we briefly describe a core functional language extended with mechanisms to express adaptation to context changes, to manipulate resources and to enforce security policies. In addition, we shall outline a static analysis for guaranteeing programs to securely behave in the digital environment they are part of.

1 Introduction

We can be connected at any time and anywhere. Internet is *de facto* becoming the infrastructure providing us with wired or wireless access points for our digitally instrumented life. A great variety of activities and tasks performed by individuals are mediated, supported and affected by different heterogeneous digital systems that in turn often cooperate each other without human intervention. These digital entities can be any combination of hardware devices and software pieces or even people, and their activities can change the physical and the virtual environment where they are plugged in. For example, in a smart house, a sensor can proactively switch on the heater to regulate the temperature. An emerging line is therefore integrating these entities into an active and highly dynamic digital environment that hosts end-users, continuously interacting with it. Consequently, the digital environment assumes the form of a communication infrastructure, through which its entities can interact each other in a loosely coupled manner, and they can access resources of different kinds, e.g., local or remote, private or shared, data or programs, devices or services. The name *ubiquitous computing* is usually adopted to denote this phenomena.

Some illustrative, yet largely incomplete, cases of computational models and technologies towards the realisation of this approach have been already developed and deployed. Among these, the most significant are Service Oriented Computing, the Internet of Things and Cloud Computing. Each of them is fostered by

^{*} This work has been partially supported by IST-FP7-FET open-IP project ASCENS and Regione Autonoma Sardegna, L.R. 7/2007, project TESLA.

and addresses different aspects of the implementation and the usage of ubiquitous computing as follows.

In the Service Oriented Computing approach, applications are open-ended, heterogenous and distributed. They are built by composing software units called services, which are published, linked, and invoked on-demand by other services using standard internet-based protocols. Moreover, applications can dynamically reconfigure themselves, by re-placing the services in use with others. Finally, services are executed on heterogeneous systems and no assumptions can be taken on their running platforms. In brief, a service offers its users access remote resources, i.e. data and programs.

A further step towards ubiquitous computing is when software pervades the objects of our everyday life, e.g. webTV, cars, smartphones, ebook readers, etc. These heterogenous entities often have a limited computational power, but are capable of connecting to the internet, coordinating and interacting each other, in the so-called “plug&play” fashion. The real objects, as well as others of virtual nature (programs, services, etc.), which are connected in this way, form the Internet of Things. Objects become points where information can be collected and where some actions can be performed to process it, so changing the surrounding environment.

Cloud computing features facilities that are present on both the approaches above. Indeed, it offers through the network a hardware and software infrastructure on which end-users can run their programs on-demand. In addition, a rich variety of dynamic resources, such as networks, servers, storage, applications and services are made available. A key point is that these resources are “virtualised” so that they appear to their users as fully dedicated to them, and potentially unlimited.

The three approaches briefly surveyed above share some peculiar aspects, in that the objects they manipulate are highly dynamic, open-ended, available on-demand, heterogeneous, and always connected. At the same time, also the infrastructure hosting the digital entities needs to efficiently support connectivity, elastic re-configuration, and resource access and handling. Mechanisms are therefore in order to adapt the shape and modalities of the interactions among digital entities, making their behaviour context-aware. These can join and leave the digital environment at will, so the infrastructure must protect itself and the users by supplying security guarantees.

Many different techniques and approaches are being proposed to tackle the issues typical of the ubiquitous computing scenario. In spite of their importance, we shall completely neglect the social and legal aspects, and refer the interested reader to [96]. In this paper, we shall instead rely on *Formal Methods*, in particular from a language-based perspective. Being formal offers the mathematical bases supporting a well-established repertoire of techniques, methods and tools for a rigorous development of applications. In turn, these are to be implemented in a programming language with high-level constructs, endowed with a clear semantics.

More precisely, we shall focus on adaptivity and security. Adaptivity is the capability of digital entities to fit for a specific use or situation; in a pervasive computing scenario this is a key aspect. Security is mandatory because the apparent simplicity of use of the new technologies hides their not trivial design and implementation, that become evident only when something goes wrong. In general, the risk is exchanging simplicity for absence of attention. For example, [71] reports on an easy attack to a wireless insulin pump, that enables the intruder, who bypasses authentication, to dangerously quadruple the dose remotely.

The next section will briefly survey the security issues of context-aware systems developed in the fields of Service Oriented Computing, the Internet of Things and Cloud computing. In Section 3 we shall discuss the interplay between context-awareness and security, and in Section 4 we shall introduce our recent proposal through a running example. Section 5 briefly overviews our proposal more technically [53]. We describe a core functional language extended with mechanisms to express adaptation to context changes, to manipulate resources and to enforce security policies. In addition, we shall outline a static analysis for guaranteeing programs to safely adapt their behaviour to the changes of the digital environment they are part of, and to correctly interact with it.

2 State of the Art and Challenges

There is a very rich literature about ubiquitous computing, from different points of view, including social, political, economical, technological and scientific ones. By only considering the approaches within the last two viewpoints, a large number of technological and scientific communities grew in a mesh of mostly overlapping fields, each one with its own methodologies and tools.

Below, we focus on three branches, and related technologies, that we consider pivotal in ubiquitous computing from a developer perspective. We think that security and context-awareness are among the main concerns of ubiquitous computing, and so we mainly report on the results of the formal method community on these aspects.

2.1 Service Oriented Computing

Service Oriented Computing (SOC) is a well-established paradigm to design distributed applications [81,80,79,50]. In this paradigm, applications are built by assembling together independent computational units, called *services*. Services are stand-alone components distributed over a network, and made available through standard interaction mechanisms.

The main research challenges in SOC are described in [80]. An important aspect is that services are *open*, in that they are built with little or no knowledge about their operating environment, their clients, and further services therein invoked. Adaptivity shows up in the SOC paradigm at various levels. At the lower one, the middleware should support dynamically reconfigurable run-time architectures and dynamic connectivity. Service composition heavily depends on

which information about a service is made public; on how those services are selected that match the user's requirements; and on the actual run-time behaviour of the chosen services. Service composition demands then autonomic mechanisms also driven by business requirements. The service oriented applications, made up by loosely coupled services, also require self management features to minimise human intervention: self-configuring, self-adapting, self-healing, self-optimising, in the spirit of autonomic computation [67].

A crucial issue concerns defining and enforcing non-functional requirements of services, e.g. security and service level agreement ones. In particular, service assembly makes security imposition even harder. One reason why is that services may be offered by different providers, which only partially trust each other. On the one hand, providers have to guarantee the delivered service to respect a given security policy, in any interaction with the open operational environment, and regardless of who actually called the service. On the other hand, clients may want to protect their sensible data from the services invoked. Furthermore, security may be breached even when all the services are trusted, because of unintentional behaviour due, e.g. to design or implementation bugs, or because the composition of the services exhibits some unexpected and unwanted behaviour, e.g. leakage of information.

Web Services [8,88,94] built upon XML technologies are possibly the most illustrative and well developed example of the SOC paradigm. Indeed, a variety of XML-based technologies already exists for describing, discovering and invoking web services [45,28,12,2]. There are several standards for defining and enforcing non-functional requirements of services, e.g. WS-Security [14], WS-Trust [11] and WS-Policy [29]. The kind of security taken into account in these standards only concerns end-to-end requirements about secrecy and integrity of the messages exchanged by the parties.

Assembly of services can occur in two different flavours: *orchestration* or *choreography*. Orchestration describes the interactions from the point of view of a single service, while choreography has a global view, instead. Languages for orchestration and choreography have been proposed, e.g. BPEL4WS [12,72] and WS-CDL [70]. However these languages have no facilities to explicitly handle security of compositions, only being focussed on end-to-end security. Instead, XACML [3] gives a more structured and general approach, because it allows for declaring access control rules among objects that can be referenced in XML.

It turns out that the languages mentioned above do not describe many non-functional requirements, especially the ones concerning the emerging behaviour obtained by assembling services.

The literature on formal methods reports on many (abstract) languages for modelling services and their orchestration, see [56,27,60,73,20,77,74,97,38,34] just to cite a few; [23] is a recent detailed survey on approaches to security and related tools, especially within the process calculi framework, [76] provides a formal treatment for a subset of XACML. An approach to the secure composition of services is presented in [19,20]. Services may dynamically impose policies on resource usage, and they are composed guaranteeing that these policies will

actually be respected at run-time. The security control is done efficiently at static time, by exploiting a type and effect system and model-checking.

The problem of relating orchestration and choreography is addressed in [40,46], but without focusing on security issues.

Recently, increasing attention has been devoted to express service contracts as behavioural or session types [65]. These types synthesise the essential aspects of the interaction behaviour of services, while allowing for efficient static verification of properties of composed systems. Through session types, [91] formalises compatibility of components and [33] describes adaptation of web services. Security has also been studied using session types, e.g. by [26,17,16].

2.2 Internet of Things

In 1988, Mark Weiser described his vision about the coming age of ubiquitous computing.

“Ubiquitous computing names the third wave in computing, just now beginning. First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop. Next comes ubiquitous computing, or the age of calm technology, when technology recedes into the background of our lives.”

In this world, sensors and computers are commodities available everywhere and surrounding people anytime. This idea has given rise to what is now called the “Internet of Things” [15] or “Everyware” [58]. Due to the pervasive integration of connectivity and identification tags, real objects are represented by digital entities in the virtual environment supported by dynamic opportunistic networks [82] or by the Internet. This is the case, e.g., of a fridge, that becomes an active entity on the Internet ready to be queried for its contents.

Such an intelligent space is then made of smart things, physical and endowed with software pieces, or fully virtual, that are highly interconnected and mutually influence their behaviour.

The software of intelligent spaces has then to be aware of the surrounding environment and of the ongoing events, to reflect the idea of an active space reacting and adapting to a variety of different issues, e.g. arising from people, time, programs, sensors and other smart things. Besides being assigned a task, a device can also take on the responsibility of proactively performing some activities with or for other digital entities.

The *Ambient calculus* [41] is among the first proposals to formalise those aspects of intelligent, virtual environments that mainly pertain to the movement of (physical devices and) software processes. The processes are hosted in virtual, separated portions of the space, called *ambients*. Processes can enter and leave ambients, and be executed in them. This calculus has been used, e.g. in [36] to specify a network of devices, in particular to statically guarantee some properties

of them, e.g. for regulating rights to the provision and discovery of environmental information.

Security plays a key role in the Internet of Things, not only because any digital entity can plug in, but also because the smart things may be devices, with also specific physical dimensions. The key point here is that information and physical security became interdependent, and traditional techniques that only focus on digital security are inadequate [39]. For example, there are some papers facing these issues with suitable extensions of the Ambient calculus, among which [75,37,31,30,93], but these proposals do not address the protection of the physical layer of smart things.

In addition, since spaces are active, the digital entities composing them may easily collect sensible information about the nearby people and things. Privacy may therefore be violated, because people are not always aware that their sensible data may be collected, nor that their activities are always context-aware, either. Even worse: it is possible through data mining to disclose pieces of information, possibly confidential, so originating a tension with a tacit assumption or an explicit guarantee of privacy. For example, the analysis of behavioural patterns over big data can infer the actual identity of the individuals, violating their assumed anonymity [89].

Although some workshops and conferences are being organised on the new security topics of the Internet of Things, to the best of our knowledge little work is done in the area of formal methods, except for studies on protocols that guarantee anonymity (for brevity, we only refer the reader to the discussion on related work and to the references of [99]).

2.3 Cloud Computing

The US National Institute of Standards and Technology defines Cloud computing as follows:

Cloud computing is a model for enabling convenient, on-demand network access a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Cloud computing refers therefore to both the applications therein deployed and made available on the Internet and to the hardware infrastructures that makes this possible. The key characteristics of Cloud computing include on-demand self-service, ubiquitous network access, resource pooling. Also, Cloud systems are characterised by some peculiar adaptivity characteristics called *elasticity* and *measured services* [10]. These are related to the two different viewpoints of the Cloud that customers and providers have. Elasticity refers to the ability of the provider to adapt its hardware infrastructure with minimal effort to the requirements of different customers, by scaling up and down the resources assigned to them. Measured services indicates load and resource usage optimisation. It refers then to scalability from the provider point of view, usually achieved by

multi-tenancy, i.e. by the capability of dynamically and accurately partitioning an infrastructure shared among various consumers.

The Cloud offers different models of services: “Software as a Service”, “Platform as a Service” and “Infrastructure as a Service”, so subsuming SOC. These three kinds can be organised in a stack hierarchy where the higher ones are built on the lower ones. At the lowest level, the Cloud provider offers an infrastructure made up by virtual machines, storages, network interconnections, etc. The whole application environment is granted to the user, who takes the responsibility for it. When supplying a platform as a service, the provider gives a basic software stack, usually including the operating system and a programming environment. At the highest level, we have Software as a Service, i.e. the provider enables its users to run on-demand one of its software service.

These aspects have been recently tackled within the formal methods approach. A formal calculus for defining the architecture of virtualised systems has been proposed in [22]. Elasticity has been studied and formalised in [32] through a calculus of processes with a notion of groups. A core functional language extended with explicit primitives for computational resource usage has been proposed in [24]. A process calculus with explicit primitives to control the distributed acquisition of resources has been presented in [25].

The most relevant security issues in Cloud computing deal with access to resources, their availability and confidentiality [90]. Due to the distributed nature of the computation carried on in the Cloud, one challenge is to ensure that only authorised entities obtain access to resources. A suitable identity management infrastructure should be developed, and users and services are required to authenticate with their credentials. However such a feature may affect the level of interoperability that a Cloud needs to have, because of the management of the identity tokens and of the negotiation protocols. In addition, providers should guarantee the right level of isolation among partitioned resources within the same infrastructure, because multi-tenancy can make unstable the borders between the resources dedicated to each user. Virtualisation usually helps in controlling this phenomenon.

Moreover users should protect themselves from a possibly malicious Cloud provider, preventing him from stealing sensible data. This is usually achieved by encryption mechanisms [10] and identity management. It is worth noting that whenever a program running in the Cloud handles some encrypted data, and the attacker is the provider itself, encryption is useless if also the encoding key is on the Cloud. For a limited number of cases, this problem can be circumvented by using homomorphic encryption schemata [57,6]. There are providers that support working groups (e.g. [1]) who are aiming at making efficient homomorphic encryption so to commercially exploit it in the Cloud [78].

Note in passing that the Cloud can be misused and support attacks to security. Indeed, the great amount of computational resources made easily available can be exploited for large-scale hacking or denial of service attacks, and also to perform brute force cracking of passwords [4].

In the currently available systems, the responsibility for dealing with security issues is often shared between customer and providers. The actual balance depends on the service level at which security has to be enforced [9].

3 Adaptivity and Security

In the previous sections, we overviewed a few technological and foundational features of ubiquitous computing, focussing on the Service Oriented Computing, the Internet of Things and the Cloud paradigms, from the developers' perspective. An emergent challenge is integrating adaptivity and security issues to support programming of applications and systems in the ubiquitous setting.

Adaptivity refers to the capability of a digital entity, in particular of its software components, to dynamically modify its behaviour reacting to changes of the surrounding active space, such as the location of its use, the collection of nearby digital entities, and the hosting infrastructure [86,35]. Software must therefore be aware of its running environment, represented by a declarative and programmable *context*.

The notion of context assumes different forms in the three computational models discussed earlier. The shape of contexts in Service Oriented Computing is determined by the various directories where services are published and by the end-points where services are actually deployed and made available, as well as by other information about levels of service, etc. In the Internet of Things, the context is a (partial) representation of the active space hosting and made of the digital entities; so each entity may have its own context. In the Cloud, a context contains the description of the computational resources offered by the centralised provider, and also a measure of the available portion of each resource; note that the context has to show to the provider the way computational resources are partitioned, for multi-tenancy.

A very short survey of the approaches to context-awareness follows, essentially from the language-based viewpoint (see, e.g. SCEL [51]). Other approaches range on a large spectrum, from the more formal description logics used to representing and querying the context [43,95,59] to more concrete ones, e.g. exploiting a middleware for developing context-aware programs [84].

Another approach is Context Oriented Programming (COP), introduced by Costanza [49]. Also subsequent work [63,5,69,13] follow this paradigm to address the design and the implementation of concrete programming languages. The notion of *behavioural variation* is central to this paradigm. It is a chunk of behaviour that can be activated depending on the current working environment, i.e. of the context, so to dynamically modify the execution. Here, the context is a stack of *layers*, and a programmer can activate/deactivate layers to represent changes in the environment. This mechanism is the engine of context evolution. Usually, behavioural variations are bound to layers: activating/deactivating a layer corresponds to activating/deactivating a behavioural variation. Only a

few papers in the literature give a precise semantic description of the languages within the Context Oriented Programming paradigm. Among these, we refer the reader to [48,68,64,47,52], that however do not focus on security issues.

Security issues, instead, have been discussed in [42], even though this survey mainly considers specific context-aware applications, but not from a general formal methods viewpoint. Combining security and context-awareness requires to address two distinct and interrelated aspects. On the one side, security requirements may reduce the adaptivity of software. On the other side, new highly dynamic security mechanisms are needed to scale up to adaptive software. Such a duality has already been put forward in the literature [98,39], and we outline below two possible ways of addressing it: *securing context-aware systems* and *context-aware security*.

Securing context-aware systems aims at rephrasing the standard notions of confidentiality, integrity and availability [83] and at developing techniques for guaranteeing them [98]. Contexts may contain sensible data of the working environment (e.g. information about surrounding digital entities), and therefore to grant confidentiality this contextual information should be protected from unauthorised access. Moreover, the integrity of contextual information requires mechanisms for preventing its corruption by any entity in the environment. A trust model is needed, taking also care of the roles of entities that can vary from a context to another. Such a trust model is important also because contextual information can be inferred from the environmental one, provided by or extracted from digital entities therein, that may forge deceptive data. Since information is distributed, denial-of-service can be even more effective because it can prevent a whole group of digital entities to access relevant contextual information.

Context-aware security is dually concerned with the definition and enforcement of high-level policies that talk about, are based on, and depend on the notion of dynamic context. The policies most studied in the literature control the accesses to resources and smart things, see among the others [98,66,100]. Some e-health applications show the relevance of access control policies based on the roles attached to individuals in contexts [74,54].

Most of the work on securing context-aware systems and on context-aware security aims at implementing various features at different levels of the infrastructures, e.g. in the middleware [84] or in the interaction protocols [62]. Indeed, the basic mechanisms behind security in adaptive systems have been studied much less. Moreover, the two dual aspects of context-aware security sketched above are often tackled separately. We lack then a unifying concept of security.

Our proposal faces the challenges pointed out above, by formally endowing a programming language with linguistic primitives for context-awareness and security, provided with a clear formal semantics. We suitably extend and integrate together techniques from COP, type theory and model-checking. In particular, we develop a static technique ensuring that a program: (i) adequately reacts to context changes; (ii) accesses resources in accordance with security policies; (iii) exchanges messages, complying with specific communication protocols.

4 An Example

In this section a working example intuitively illustrates our methodology, made of the following three main ingredients:

1. a COP functional language, called ContextML [53,52], with constructs for resource manipulation and communication with external parties, and with mechanisms to declare and enforce security policies, that talk about context, including roles of entities, etc. We consider regular policies, in the style of [61], i.e. safety properties of program traces;
2. a type and effect system for ContextML. We exploit it for ensuring that programs adequately react to context changes and for computing as effect an abstract representation of the overall behaviour. This representation, in the form of *History Expressions*, describes the sequences of resource manipulation and communication with external parties in a succinct form;
3. a model check on the effects to verify that the component behaviour is correct, i.e. that the behavioural variations can always take place, that resources are manipulated in accordance with the given security policies and that the communication protocol is respected.

Consider a typical scenario of ubiquitous computing. A smartphone app remotely uses a Cloud as a repository to store and synchronise a library of ebooks. Also it can execute locally, or invoke remotely customised services. In this example we consider a simple full-text search.

A user buys ebooks online and reads them locally through the app. The purchased ebooks are stored into the remote user library and some books are kept locally in the smartphone. The two libraries may be not synchronised. The synchronisation is triggered on demand and it depends on several factors: the actual bandwidth available for connection; the free space on the device; etc.

This example shows that our programmable notion of context can represent some of the environmental information briefly discussed in Section 3. In particular, the context is used to represent the location where the full-text search is performed; the status of the device and of the resources (synchronised or not) offered by the Cloud; and the actual role of the service caller. We specify below the fragment of the app that implements the search over the user's library.

Consider the context dependent behaviour emerging because of the different energy profiles of the smartphone. We assume that there are two: one is active when the device is plugged in, the other is active when it is using its battery. These profiles are represented by two *layers*: **ACMode** and **BatMode**. The function `getBatteryProfile` returns the layer describing the current active profile depending on the value of the sensor (`plugged`):

```
fun getBatteryProfile x = if (plugged) then ACMode else BatMode
```

Layers can be activated, so modifying the context. The expression

```
with(getBatteryProfile()) in exp1 (1)
```

activates the layer obtained by calling `getBatteryProfile`. The scope of this activation is the expression exp_1 in Fig. [11\(a\)](#). In lines 2-10, there is the following *layered expression*:

```
ACMode. ⟨DO SEARCH⟩,
BatMode. ⟨DO SOMETHING ELSE⟩
```

This is the way context-dependent *behavioural variations* are declared. Roughly, a layered expression is an expression defined by cases. The cases are given by the different layers that may be active in the context, here `BatMode` and `ACMode`. Each layer has an associated expression. A *dispatching mechanism* inspects at runtime the context and selects an expression to be reduced. If the device is plugged in, then the search is performed locally, abstracted by `⟨DO SEARCH⟩`. Otherwise, something else gets done, abstracted by `⟨DO SOMETHING ELSE⟩`. Note that if the programmer neglects a case, then the program throws a runtime error being unable to adapt to the actual context.

In the code of exp_1 (Fig. [11\(b\)](#)), the function g consists of nested layered expressions describing the behavioural variations matching the different configurations of the execution environment. The code exploits context dependency to take into account also the actual location of the execution engine (remote in the Cloud at line 3, or local on the device at line 4), the synchronisation state of the library, at lines 5,6, and the active energy profile at lines 2,10. The smartphone communicates with the Cloud system over the bus through message passing primitives, at lines 7-9. The search is performed locally only if the library is fully synchronised and the smartphone is plugged in. If the device is plugged in, but the library is not fully synchronised, then the code of function g is sent to the Cloud and executed remotely by a suitable server.

Lines 7-10 specify some communications of the service, in quite a simple model. Indeed, we adopt a *top-down* approach [\[44\]](#) to describe the interactions between programs, based on a unique channel of communication, the *bus*, through which messages are exchanged. For simplicity, we assume the operational environment to give the protocol P governing the interactions.

In Fig. [11\(a\)](#) we show a fragment of the environment provided by the cloud. The service considered offers generic computational resources to the devices connected on the bus, by continuously running f . The function f listens to the bus for an incoming request from a user identified by id . Then the provider updates the billing information for the customer and waits for incoming code (a function) and an incoming layer. Finally, it executes the received function in a context extended with the received layer. Note that before executing the function, the Cloud switches its role from `Root`, with administrator rights, to the role `Usr`.

In the code of the Cloud there are two security policies φ, φ' , the scopes of which are expressed by the security framings $\varphi[\dots], \varphi'[\dots]$. Intuitively, they cause a sandboxing of the enclosed expression, to be executed under the strict monitoring of φ and φ' respectively. The policy φ specifies the infrastructural rules of the Cloud. Among the various controls, it will inhibit a `Usr` to become `Root`. Indeed, being context-aware, our policies can also express role-based or

location-based policies. Instead φ' is enforced right before running the received function g and expresses that writing on the library `write(library)` is forbidden (so only reading is allowed). In this way, we guarantee that the execution of external code does not alter the remote library.

The viable interactions on the bus are constrained by a given protocol P . We assume that the given protocol P is indeed an abstraction of the behaviour of the various parties involved in the communications. We do not address here how protocols are defined by the environment and we only check whether a program respects the given protocol.

In our example the app must send the identifier of the user, the layer that must be active and the code implementing the search. Eventually, the app receives the result of the search. This sequence of interactions is precisely expressed by the following protocol.

$$P = (\text{send}_{\tau_{id}} \text{send}_{\tau} \text{send}_{\tau'} \text{receive}_{\tau''})^*$$

The values to be sent/received are represented in the protocol by their type: τ_{id} , τ , τ' , τ'' . We will come back later on the usage of these types. The symbol $*$ means that this sequence of actions can be repeated any number of times.

Function `getBatteryProfile` returns either layer value `ACMode` or `BatMode`. So the function is assigned the type $ly_{\{\text{ACMode}, \text{BatMode}\}}$ meaning that the returned layer is one between the two mentioned above.

Function g takes the type `unit` and returns the type τ'' , assuming that the value returned by the `search` function has type τ'' . The application of g depends on the current context. For this reason we enrich the type with a set of preconditions \mathbb{P} where each precondition $v \in \mathbb{P}$ is a set of layers. In our example the precondition \mathbb{P} is

$$\mathbb{P} = \{\{\text{ACMode}, \text{IsLocal}, \text{LibrarySynced}\}, \{\text{ACMode}, \text{IsCloud}\}, \dots\}$$

In order to apply g , the context of application must contains all the layers in v , for a preconditions $v \in \mathbb{P}$. Furthermore, we annotate the type with the latent effect H . It is a history expression and represents (a safe over-approximation of) the sequences of resource manipulation or layer activations or communication actions, possibly generated by running g . To summarise the complete type of g is `unit` $\xrightarrow{\mathbb{P}|H} \tau''$.

Our type system guarantees that, if a program type-checks, the dispatching mechanism always succeeds at run-time. In our example, the expression (II) will be well-typed whenever the context in which it will be evaluated contains either `IsLocal` or `IsCloud`, and either `LibraryUnsynced` or `LibrarySynced`. The preconditions over `ACMode` and `BatMode` coming from exp_1 are ensured in (II) . This is because the type of `getBatteryProfile` guarantees that one among them will be activated in the context by the construct `with`.

Effects are then used to check whether a client complies with the policy and the interaction protocol provided by the environment. Verifying that the code of g obeys the policies φ and φ' is done by standard model-checking the effect

```

1 : fun f x =
2 :    $\varphi$ [with(Root) in
3 :     let id = receive $_{\tau_{id}}$  in
4 :       cd(/billing); write(bid_id)
5 :       cd(..lib_id)
6 :     ;
7 :   with(Usr) in
8 :     let lyr = receive $_{\tau}$  in
9 :     let g = receive $_{\tau'}$  in
10 :       $\varphi'$ [with(lyr) in
11 :        let res = g() in
12 :        send $_{\tau''}$ (res)
13 :      ]
14 :   ]; f()

```

(a)

```

1 : fun g x =
2 :   ACMode.
3 :     IsCloud.search(),
4 :     IsLocal.
5 :       LibrarySynced.search(y),
6 :       LibraryUnsynced.
7 :         send $_{\tau_{id}}$ (myid);
8 :         send $_{\tau}$ (ACMode);
9 :         send $_{\tau'}$ (g);
10 :        receive $_{\tau''}$ 
11 :   BatMode. <DO SOMETHING ELSE>

```

(b)

Fig. 1. Two fragments of a service in the Cloud (a) and of an app (b)

of g (a context-free language) against the policies (regular languages). Since in our example the app never writes and never changes the actual role to Root, the policies φ' and φ are satisfied (under the hidden assumption that the code for the BatMode case has an empty effect).

To check compliance with the protocol, we only consider communications. Thus, the effect of exp_1 becomes:

$$H_{sr} = send_{\tau_{id}} \cdot send_{\tau} \cdot send_{\tau'} \cdot receive_{\tau''}$$

Verifying whether the program correctly interacts with the Cloud system consists of checking that the histories generated by H_{sr} are a subset of those allowed by the protocol $P = (send_{\tau_{id}} send_{\tau} send_{\tau'} receive_{\tau''})^*$. This is indeed the case here.

5 ContextML

Context and adaptation features included in ContextML are borrowed from COP languages [63] discussed above. Indeed, ContextML is characterised by: (i) a declarative description of the working environment, called *context*, that is a stack of *layers*; (ii) layers describing properties about the application current environment; (iii) constructs for activating layers; (iv) mechanism for defining behavioural variations.

The resources available in the system are represented by identifiers and can be manipulated by a fixed set of actions. For simplicity, we here omit a construct for dynamically creating resources, that can be dealt with following [20, 18].

We enforce security properties by protecting expressions with policies: $\varphi[e]$. This construct is called *policy framing* [18] and it generalises the standard *sandbox* mechanism. Roughly, it means that during the evaluation of the program

e the computation performed so far must respect the policy φ in the so-called history-dependent security.

The communication model is based on a bus which allows programs to interact with the environment by message passing. The operations of writing and reading values over this bus can be seen as a simple form of asynchronous I/O. We will not specify this bus in detail, but we will consider it as an abstract entity representing the whole external environment and its interactions with programs. Therefore, ContextML programs operate in an open-ended environment.

5.1 Syntax and Semantics

Let \mathbb{N} be the naturals, Ide be a set of identifiers, LayerNames be a finite set of layer names, Policies be a set of security policies, Res be a finite set of resources (identifiers) and Act be a finite set of actions for manipulating resources. The syntax of ContextML is as follows:

$n \in \mathbb{N}$	$x, f \in \text{Ide}$	$L \in \text{LayerNames}$
$\varphi \in \text{Policies}$	$r \in \text{Res}$	$\alpha, \beta \in \text{Act}$
$v, v' ::=$	<i>values</i>	
$ n \mid \mathbf{fun} f x \Rightarrow e \mid () \mid L$		
$lexp ::=$	<i>layered expressions</i>	
$ L.e \mid L.e, lexp$		
$e, e' ::=$	<i>expressions</i>	
$ \mathbf{v}$	value	Core ML
$ \mathbf{x}$	variable	Core ML
$ e_1 e_2$	application	Core ML
$ e_1 \mathbf{op} e_2$	operation	Core ML
$ \mathbf{if} e_0 \mathbf{then} e_1 \mathbf{else} e_2$	conditional expression	Core ML
$ \mathbf{with}(e_1) \mathbf{in} e_2$	with	Context
$ lexp$	layered expressions	Context
$ \alpha(r)$	access event	Resource
$ \varphi[e]$	policy framing	Policy
$ \mathbf{send}_\tau(e)$	send	Communication
$ \mathbf{receive}_\tau$	receive	Communication

In addition, we adopt the following standard abbreviations: $\mathbf{let} x = e_1 \mathbf{in} e_2 \triangleq (\mathbf{fun} _ x \Rightarrow e_2) e_1$ and $e_1; e_2 \triangleq (\mathbf{fun} f x \Rightarrow e_2) e_1$, with x, f not free in e_2 .

The core of ML is given by the functional part, modelled on the call-by-value λ -calculus.

The primitive **with** models the evaluation of the expression e_2 in the context extended by the layer obtained by the evaluation of e_1 . Behavioural variations are defined by layered expression (*lexp*), expressions defined by cases each specifying a different behaviour depending on the actual structure of the context.

The expression $\alpha(r)$ models the invocation of the access operation α over the resource r , causing side effects. Access labels specify the kind of access operation, e.g. read, write, and so on.

A security policy framing $\varphi[e]$ defines the scope of the policy φ to be enforced during the evaluation of e . Policy framings can also be nested.

The communication is performed by **send** $_{\tau}$ and **receive** $_{\tau}$ that allow the interaction with the external environment by writing/reading values of type τ (see Subsection 5.3) to/from the bus.

Dynamic Semantics. We endow ContextML with a small-step operational semantics, only defined, as usual, for closed expressions (i.e. without free variables).

Our semantics is history dependent. Program *histories* are sequences of events that occur during program execution. Events ev indicate the observation of critical activities, such as activation (deactivation) of layers, selection of behavioural variations and program actions, like resource accesses, entering/exiting policy framings and communication. The syntax of events ev and programs histories η is the following:

$$ev ::= \langle_L \mid \rangle_L \mid \text{Disp}(L) \mid \alpha(r) \mid [\varphi \mid]_{\varphi} \mid \text{send}_{\tau} \mid \text{receive}_{\tau} \quad (2)$$

$$\eta ::= \epsilon \mid ev \mid \eta \eta \quad (\epsilon \text{ is the empty history}) \quad (3)$$

A history is a possibly empty sequence of events occurring at runtime. The event $\langle_L \mid \rangle_L$ marks that we begin (end) the evaluation of a **with** body in a context where the layer L is activated (deactivated), the event $\text{Disp}(L)$ signals that the layer L has been selected by the dispatching mechanism. The event $\alpha(r)$ marks that the action α has been performed over the resource r ; the event $[\varphi \mid]_{\varphi}$ records that we begin (end) the enforcement of the policy φ ; the event $\text{send}_{\tau}/\text{receive}_{\tau}$ indicates that we have sent/read a value of type τ over/from the bus.

A context C is a stack of active layers with two operations. The first $C - L$ removes a layer L from the context C if present, the second $L :: C$ pushes L over $C - L$. Formally:

Definition 1. We denote the empty context by $[]$ and a context with n elements with L_1 at the top, by $[L_1, \dots, L_n]$. Let $C = [L_1, \dots, L_{i-1}, L_i, L_{i+1}, \dots, L_n]$, $1 \leq i \leq n$ then

$$C - L = \begin{cases} [L_1, \dots, L_{i-1}, L_{i+1}, \dots, L_n] & \text{if } L = L_i \\ C & \text{otherwise} \end{cases}$$

$$L :: C = [L, L_1, \dots, L_n] \text{ where } [L_1, \dots, L_n] = C - L$$

The semantic rules of the core part are inherited from ML and we will omit them. Below, we will show and comment only the ones for the new constructs.

The transitions have the form $C \vdash \eta, e \rightarrow \eta', e'$, meaning that in the context C , starting from a program history η , in one evaluation step the expression e may evolve to e' , extending the history η to η' . Initial configurations have the form (ϵ, e) . We write $\eta \models \varphi$ when the history η obeys the policy φ , in a sense that will be made precise in the Subsections 5.2 and 5.4.

$$\begin{array}{c}
\text{with}_1 \frac{C \vdash \eta, e_1 \rightarrow \eta', e'_1}{C \vdash \eta, \mathbf{with}(e_1) \text{ in } e_2 \rightarrow \eta', \mathbf{with}(e'_1) \text{ in } e_2} \\
\text{with}_2 \frac{}{C \vdash \eta, \mathbf{with}(L) \text{ in } e \rightarrow \eta \llbracket_L, \mathbf{with}(\bar{L}) \text{ in } e} \\
\text{with}_3 \frac{L :: C \vdash \eta, e \rightarrow \eta', e'}{C \vdash \eta, \mathbf{with}(\bar{L}) \text{ in } e \rightarrow \eta', \mathbf{with}(\bar{L}) \text{ in } e'} \\
\text{with}_4 \frac{}{C \vdash \eta, \mathbf{with}(\bar{L}) \text{ in } v \rightarrow \eta \rrbracket_{L, v}}
\end{array}$$

The rules for **with**(e_1) **in** e_2 evaluate e_1 in order to obtain the layer L (with₁) that must be activated to eventually evaluate the expression e_2 (with₃). In addition, in the history, the events \llbracket_L and \rrbracket_L mark the beginning (with₂) and the end (with₄) of the evaluation of e_2 . Note that being within the scope of layer L activation is recorded by using \bar{L} (with₂). When the expression is just a value the **with** is removed (with₄).

$$\text{lexp} \frac{L_i = \text{Disp}(C, \{L_1, \dots, L_n\})}{C \vdash \eta, L_1.e_1, \dots, L_n.e_n \rightarrow \eta \text{Disp}(L_i), e_i}$$

In evaluating a layered expression $e = L_1.e_1, \dots, L_n.e_n$ (rule lexp), the current context is inspected top-down to select the expression e_i that corresponds to the layer L_i , selected by the dispatching mechanism illustrated below. The history is updated by appending $\text{Disp}(L_i)$ to record that the layer L_i has been selected. The dispatching mechanism is implemented by the partial function Disp , defined as

$$\text{Disp}([L'_0, L'_1, \dots, L'_m], A) = \begin{cases} L'_0 & \text{if } L'_0 \in A \\ \text{Disp}([L'_1, \dots, L'_m], A) & \text{otherwise} \end{cases}$$

It returns the first layer in the context $[L'_0, L'_1, \dots, L'_m]$ which matches one of the layers in the set A . If no layer matches, then the computation gets stuck.

$$\text{action} \frac{}{C \vdash \eta, \alpha(r) \rightarrow \eta \alpha(r), ()}$$

The rule (action) describes the evaluation of an event $\alpha(r)$ that consists in extending the current history with the event itself, and producing the unit value $()$.

$$\begin{array}{c}
\text{framing}_1 \frac{\eta^{-\Box} \models \varphi}{C \vdash \eta, \varphi[e] \rightarrow \eta[\varphi, \overline{\varphi}[e]} \\
\text{framing}_2 \frac{C \vdash \eta, e \rightarrow \eta', e' \quad \eta'^{-\Box} \models \varphi}{C \vdash \eta, \overline{\varphi}[e] \rightarrow \eta', \overline{\varphi}[e']} \\
\text{framing}_3 \frac{\eta^{-\Box} \models \varphi}{C \vdash \eta, \overline{\varphi}[v] \rightarrow \eta[\varphi, v]}
\end{array}$$

The policy framing $\varphi[e]$ enforces the policy φ on the expression e , meaning that the history must respect φ at each step of the evaluation of e and each event issued within e must be checked against φ . More precisely, each resulting history η' must obey the policy φ (in symbols $\eta'^{-\square} \models \varphi$). When e is just a value, the security policy is simply removed (framing₃). As for the **with** rules, placing a bar over φ records that the policy is active. Also here, in the history the events $[\varphi/\cdot]_{\varphi}$ record the point where we begin/end the enforcement of φ . Instead, if η' does not obey φ , then the computation gets stuck.

$$\begin{array}{c} \text{send}_1 \frac{C \vdash \eta, e \rightarrow \eta', e'}{C \vdash \eta, \mathbf{send}_{\tau}(e) \rightarrow \eta', \mathbf{send}_{\tau}(e')} \\[10pt] \text{send}_2 \frac{}{C \vdash \eta, \mathbf{send}_{\tau}(v) \rightarrow \eta \text{ send}_{\tau}, ()} \\[10pt] \text{receive} \frac{}{C \vdash \eta, \mathbf{receive}_{\tau} \rightarrow \eta \text{ receive}_{\tau}, v} \end{array}$$

The rules that govern communications reflect our notion of protocol, that abstractly represents the behaviour of the environment, showing the sequence of the pair direction/type of messages. Accordingly, our primitives carry types as tags, rather than dynamically checking the exchanged values. In particular, there is no check that the type of the received value matches the annotation of the primitive **receive**. Our static analysis in Subsection 5.4 will guarantee the correctness of this operation. More in detail, $\mathbf{send}_{\tau}(e)$ evaluates e (send_1) and sends the obtained value over the bus (send_2). In addition, the history is extended with the event send_{τ} . A $\mathbf{receive}_{\tau}$ reduces to the value v read from the bus and appends the corresponding event to the current history. This rule is similar to that used in the early semantics of the π -calculus, where we guess a name transmitted over the channel [85].

5.2 History Expressions

To statically predict the histories generated by programs at run-time, we introduce history expressions [87, 20, 18], a simple process algebra providing an abstraction over the set of histories that a program may generate. We recall here the definitions and the properties given in [18], extended to cover histories with a larger set of events ev , also endowing layer activation, dispatching and communication.

Definition 2 (History Expressions). *History expressions are defined by the following syntax:*

$H, H_1 ::=$	ϵ <i>empty</i>	$H_1 + H_2$	<i>sum</i>
	ev <i>events in (2)</i>	$H_1 \cdot H_2$	<i>sequence</i>
	h <i>recursion variable</i>	$\mu h. H$	<i>recursion</i>
	$\varphi[H]$ <i>safety framing, stands for $[\varphi \cdot H \cdot]_{\varphi}$</i>		

$$\begin{array}{c}
\frac{}{\epsilon \cdot H \xrightarrow{\epsilon} H} \qquad \frac{}{\alpha(r) \xrightarrow{\alpha(r)} \epsilon} \qquad \frac{}{\mu h.H \xrightarrow{\epsilon} H\{\mu h.H/h\}} \\
\\
\frac{H_1 \xrightarrow{\alpha(r)} H'_1}{H_1 \cdot H_2 \xrightarrow{\alpha(r)} H'_1 \cdot H_2} \qquad \frac{H \xrightarrow{\alpha(r)} H'}{H_1 + H_2 \xrightarrow{\alpha(r)} H'_1} \qquad \frac{H_2 \xrightarrow{\alpha(r)} H'_2}{H_1 + H_2 \xrightarrow{\alpha(r)} H'_2}
\end{array}$$

Fig. 2. Transition system of History Expressions

The signature defines sequentialisation, sum and recursion operations over sets of histories containing events; μh is a binder for the recursion variable h .

The following definition exploits the labelled transition system in Fig. 2.

Definition 3 (Semantics of History Expressions). *Given a closed history expression H (i.e. without free variables), its semantics $\llbracket H \rrbracket$ is the set of histories $\eta = w_1 \dots w_n$ ($w_i \in ev \cup \{\epsilon\}$, $0 \leq i \leq n$) such that $\exists H'. H \xrightarrow{w_1} \dots \xrightarrow{w_n} H'$.*

We remark that the semantics of a history expression is a prefix closed set of histories. For instance, $\mu h.(\alpha(r) + \alpha'(r) \cdot h \cdot \alpha''(r))$ comprises all the histories of the form $\alpha'(r)^n \alpha(r) \alpha''(r)^n$, with $n \geq 0$.

Back to the example in Section 4, assume that H is the history expression over-approximating the behaviour of the function g . Then, assuming $\tau = ly_{\text{ACMode}}$, the history expression of the fragment of the Cloud service (Fig. 1(b)) is

$$\begin{aligned}
&\mu h.\varphi[\\
&\quad (\text{Root} \cdot \text{receive}_{\tau_{id}} \cdot \text{cd}(/ \text{billing}) \cdot \text{write}(\text{bid_id}) \cdot \text{cd}(/ \text{lib}_{id}) \cdot)_{\text{Root}} \cdot \\
&\quad (\text{Usr} \cdot \text{receive}_{\tau} \cdot \text{receive}_{\tau'} \cdot \varphi'(\llbracket \text{ACMode} \cdot H \cdot \text{send}_{\tau'} \rrbracket_{\text{ACMode}}))_{\text{Usr}} \\
&\quad] \cdot h
\end{aligned}$$

Closed history expressions are partially ordered: $H \sqsubseteq H'$ means that the abstraction represented by H' is less precise than the one by H . The structural ordering \sqsubseteq is defined over the quotient induced by the (semantic-preserving) equational theory presented in [20] as the least relation such that $H \sqsubseteq H$ and $H \sqsubseteq H + H'$. Clearly, $H \sqsubseteq H'$ implies $\llbracket H \rrbracket \subseteq \llbracket H' \rrbracket$.

Validity of History Expressions. Given a history η , we denote with $\eta^{-\square}$ the history purged of all framings events $[\varphi,]_{\varphi}$. For instance, if $\eta = \alpha(r)[\varphi \alpha'(r)[\varphi'[\alpha''(r)]_{\varphi}]_{\varphi'}$ then $\eta^{-\square} = \alpha(r)\alpha'(r)\alpha''(r)$. For details and other examples, see [20].

Given a history η , the multiset $ap(\eta)$ collects all the policies φ still active, i.e. whose scope has been entered but not exited yet. These policies are called *active policies* and are defined as follows:

$$\begin{aligned}
ap(\epsilon) &= \{ \} & ap(\eta[\varphi] &= ap(\eta) \cup \{ \varphi \} \\
ap(\eta ev) &= ap(\eta) \quad ev \neq [\varphi,]_{\varphi} & ap(\eta)_{\varphi} &= ap(\eta) \setminus \{ \varphi \}
\end{aligned}$$

The validity of a history η ($\models \eta$ in symbols) is inductively defined as follows, assuming the notion of policy compliance $\eta \models \varphi$ of Subsection 5.4.

$$\begin{aligned} & \models \epsilon \\ & \models \eta' ev \quad \text{if } \models \eta' \text{ and } (\eta' ev)^{-\Box} \models \varphi \text{ for all } \varphi \in ap(\eta' ev) \end{aligned}$$

A history expression H is *valid* when $\models \eta$ for all $\eta \in \llbracket H \rrbracket$.

If a history is valid, also its prefixes are valid, i.e. validity is a prefix-closed property, as stated by the following lemma.

Property 1. If a history η is valid, then each prefix of η is valid.

For instance, if the policy φ amounts to “no $read(r)$ after $write(r)$ ”, the history $write(r)\varphi[read(r)write(r)]$ is not valid because $write(r)read(r) \not\models \varphi$ and remains not valid after, e.g. also $write(r)read(r)write(r) \not\models \varphi$. Instead, the history $\varphi[read(r)]write(r)$ is valid because both $\epsilon \models \varphi$, $read(r) \models \varphi$ and $read(r)write(r) \models \varphi$. The semantics of ContextML (in particular the rules for framing) ensures that the histories generated at runtime are all valid.

Property 2. If $C \vdash \epsilon, e \rightarrow \eta', e'$, then η' is valid.

5.3 ContextML Types

We briefly describe here our type and effect system for ContextML. We use it for over-approximating the program behaviour and for ensuring that the dispatching mechanism always succeeds at runtime. Here, we only give a logical presentation of our type and effect system, but we are confident that an inference algorithm can be developed, along the lines of [87]. All the technical properties that show the correctness of our type systems are detailed in [53]. Here, we only state the most intuitive results.

Our typing judgements have the form $\langle \Gamma; C \rangle \vdash e : \tau \triangleright H$. This reads as “in the type environment Γ and in the context C the expression e has type τ and effect H .” The associated effect H is a history expression representing all the possible histories that a program may generate.

Types are integers, unit, layers and functions:

$$\begin{aligned} \sigma & \in \wp(\text{LayerNames}) & \mathbb{P} & \in \wp(\wp(\text{LayerNames})) \\ \tau, \tau_1, \tau' & ::= \text{int} \mid \text{unit} \mid ly_\sigma \mid \tau_1 \xrightarrow{\mathbb{P}|H} \tau_2 \end{aligned}$$

We annotate layer types with sets of layer names σ for analysis reason. In ly_σ , σ safely over-approximates the set of layers that an expression can be reduced to at runtime. In $\tau_1 \xrightarrow{\mathbb{P}|H} \tau_2$, \mathbb{P} is a set of *preconditions* v , such that each v over-approximates the set of layers that must occur in the context to apply the function. The history expression H is the latent effect, i.e. a safe over-approximation of the sequence of events generated while evaluating the function.

The rules of our type and effect system are in Fig. 3. We show and comment in detail only the rules for the new constructs; the others are directly inherited from that of ML. For the sake of simplicity, we also omit some auxiliary rules and the rules for subeffecting and for subtyping that can be found in [53].

$$\begin{array}{c}
\text{(Tly)} \frac{}{\langle \Gamma; C \rangle \vdash L : ly_{\{L\}} \triangleright \epsilon} \\
\text{(Tfun)} \frac{\forall v \in \mathbb{P}. \quad \langle \Gamma, x : \tau_1, f : \tau_1 \xrightarrow{\mathbb{P}|H} \tau_2; C' \rangle \vdash e : \tau_2 \triangleright H \quad |C'| \subseteq v}{\langle \Gamma; C \rangle \vdash \mathbf{fun}f \ x \Rightarrow e : \tau_1 \xrightarrow{\mathbb{P}|H} \tau_2 \triangleright \epsilon} \\
\text{(Tapp)} \frac{\langle \Gamma; C \rangle \vdash e_1 : \tau_1 \xrightarrow{\mathbb{P}|H} \tau_2 \triangleright H_1 \quad \langle \Gamma; C \rangle \vdash e_2 : \tau_1 \triangleright H_2 \quad \exists v \in \mathbb{P}. v \subseteq |C|}{\langle \Gamma; C \rangle \vdash e_1 e_2 : \tau_2 \triangleright H_1 \cdot H_2 \cdot H} \\
\text{(Twith)} \frac{\langle \Gamma; C \rangle \vdash e_1 : ly_{\{L_1, \dots, L_n\}} \triangleright H' \quad \forall L_i \in \{L_1, \dots, L_n\}. \langle \Gamma; L_i :: C \rangle \vdash e_2 : \tau \triangleright H_i}{\langle \Gamma; C \rangle \vdash \mathbf{with}(e_1) \ \mathbf{in} \ e_2 : \tau \triangleright H' \cdot \sum_{L_i} \langle L_i \cdot H_i \rangle_{L_i}} \\
\text{(Tlexp)} \frac{\forall i. \langle \Gamma; C \rangle \vdash e_i : \tau \triangleright H_i \quad L_1 \in |C| \vee \dots \vee L_n \in |C|}{\langle \Gamma; C \rangle \vdash L_1.e_1, \dots, L_n.e_n : \tau \triangleright \sum_{L_i \in \{L_1, \dots, L_n\}} \text{Disp}(L_i) \cdot H_i} \\
\text{(Talpha)} \frac{}{\langle \Gamma; C \rangle \vdash \alpha(r) : \mathbf{unit} \triangleright \alpha(r)} \quad \text{(Tphi)} \frac{\langle \Gamma; C \rangle \vdash e : \tau \triangleright H}{\langle \Gamma; C \rangle \vdash \varphi[e] : \tau \triangleright [\varphi \cdot H]_\varphi} \\
\text{(Trec)} \frac{}{\langle \Gamma; C \rangle \vdash \mathbf{receive}_\tau : \tau \triangleright receive_\tau} \quad \text{(Tsend)} \frac{\langle \Gamma; C \rangle \vdash e : \tau \triangleright H}{\langle \Gamma; C \rangle \vdash \mathbf{send}_\tau(e) : \mathbf{unit} \triangleright H \cdot send_\tau}
\end{array}$$

Fig. 3. Typing rules

The rule (Tly) asserts that the type of a layer L is ly annotated with the singleton set $\{L\}$ and its effect is empty. In the rule (Tfun) we guess a set of preconditions \mathbb{P} , a type for the bound variable x and for the function f . For all preconditions $v \in \mathbb{P}$, we also guess a context C' that satisfies v , i.e. that contains all the layers in v : in symbols $|C'| \subseteq v$, where $|C'|$ denotes the set of layers active in the context C' . We determine the type of the body e under these additional assumptions. Implicitly, we require that the guessed type for f , as well as its latent effect H , match those of the resulting function. In addition, we require that the resulting type is annotated with \mathbb{P} .

The rule (Tapp) is almost standard and reveals the mechanism of function precondition. The application gets a type if there exists a precondition $v \in \mathbb{P}$ satisfied in the current context C . The effect is obtained by concatenating the ones of e_2 and e_1 and the latent effect H . To better explain the use of preconditions, consider the technical example in Fig. 4. There, the function $\mathbf{fun}f \ x \Rightarrow L_1.0$ is shown to have type $int \xrightarrow{\{L_1\}} int$ (for the sake of simplicity, we ignore the effects). This means that in order to apply the function, the layer L_1 must be active, i.e. must occur in the context.

The rule (Twith) establishes that the expression $\mathbf{with}(e_1) \ \mathbf{in} \ e_2$ has type τ , provided that the type for e_1 is ly_σ (recall that σ is a set of layers) and e_2 has type τ in the context C extended by the layers in σ . The effect is the union of the possible effects resulting from the evaluation of the body. This evaluation is carried on the different contexts obtained by extending C with one of the layers in σ . The special events $\langle L$ and \rangle_L mark the limits of layer activation.

By (Tlexp) the type of a layered expression is τ , provided that each sub-expression e_i has type τ and that at least one among the layers L_1, \dots, L_n occurs

$$\begin{array}{c}
\frac{\langle \Gamma, x : \text{int}, f : \tau \xrightarrow{\{|C'\|}\} \text{int}; C' \rangle \vdash 0 : \text{int} \quad L_1 \in C'}{\langle \Gamma, x : \text{int}, f : \tau \xrightarrow{\{|C'\|}\} \text{int}; C' \rangle \vdash L_1.0 : \text{int}} \\
\hline
\frac{\langle \Gamma; C \rangle \vdash \mathbf{fun} f \ x \Rightarrow L_1.0 : \text{int} \quad \langle \Gamma; C \rangle \vdash 3 : \text{int} \quad |C'| \subseteq |C|}{\langle \Gamma; C \rangle \vdash (\mathbf{fun} f \ x \Rightarrow L_1.0) 3 : \text{int}}
\end{array}$$

Fig. 4. Derivation of a function with precondition. We assume that $C' = [L_1]$, L_1 is active in C , $\text{LayerNames} = \{L_1\}$ and, for typesetting convenience, we ignore effects.

in C . When evaluating a layered expression one of the mentioned layers will be active in the current context so guaranteeing that layered expressions will correctly evaluate. The whole effect is the sum of sub-expressions effects H_i preceded by $\text{Disp}(L_i)$.

The rule (Talpha) gives expression $\alpha(r)$ type **unit** and effect $\alpha(r)$. In the rule (Tphi) the policy framing $\varphi[e]$ has the same type as e and $[\varphi.H]_\varphi$ as effect.

The expression $\mathbf{send}_\tau(e)$ has type **unit** and its effect is that of e extended with event send_τ . The expression $\mathbf{receive}_\tau$ has type τ and its effect is the event receive_τ . Note that the rules establish the correspondence between the type declared in the syntax and the checked type of the value sent/received. An additional check is however needed and will be carried on also taking care of the interaction protocol (Subsection 5.4).

The history expression H obtained as effect of an expression e safely over-approximates the set of histories η that may actually be generated during the execution of e . More formally:

Theorem 1 (Correctness)

If $\langle \Gamma; C \rangle \vdash e : \tau \triangleright H$ and $C \vdash \epsilon, e \rightarrow^* \eta, e'$, then $\eta \in \llbracket H \rrbracket$.

In [53] we also proved a more general result, long to state here: our type and effect system is *sound*. Its direct consequence is that a well-typed program may go wrong only because of policy violations or because the communication protocol is not respected. We take care of these two cases in the next subsection.

5.4 Model Checking

We discuss here our model-checking machinery for verifying whether a history expression is compliant with respect to a policy φ and a protocol P . To do this we will use the history expression obtained by typing the program.

Policy checking. A policy φ is actually a safety property [61], expressing that nothing bad will occur during a computation. Policies are expressed through Finite State Automata (FSA). We take a default-accept paradigm, i.e. only the unwanted behaviour is explicitly mentioned. Consequently, the language of φ is the set of *unwanted traces*, hence an accepting state is considered as offending. Let $L(\varphi)$ denote the language of φ .

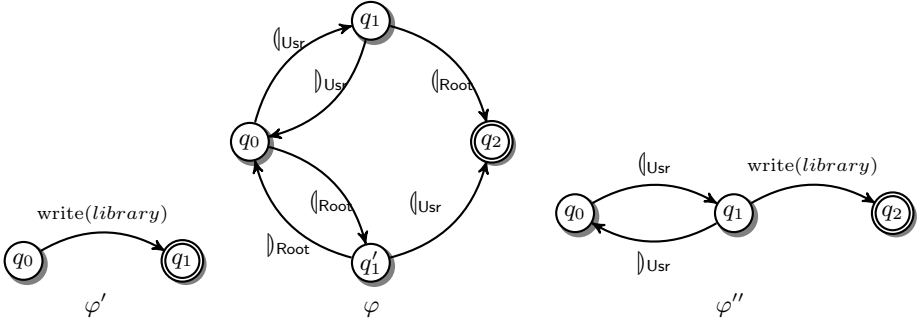


Fig. 5. Some examples of security policies. For clarity we omit the self loops in each state, that however are necessary to capture security unrelated events; they are labelled with those actions not in any outgoing edge.

The policies φ' , φ described in the example of Section 4 are in Fig. 5. The automaton for φ' expresses that writing in the library is forbidden. The one for φ describes as offending the traces that activate a layer `Root` (event $\langle Root \rangle$) while `USr` is still active (after $\langle U_{Sr} \rangle$ but before $\langle U_{Sr} \rangle$), or viceversa. The automaton for φ'' is an example of context-aware policy. In particular it states a role-based policy where writing in the library is forbidden when `USr` is impersonated. Note that also φ'' is an infrastructural policy, just as it was φ in the example of Section 4.

We now complete the definition of validity of a history η , $\eta \models \varphi$, anticipated in Section 5.2. Recall also that a history expression is valid when all its histories are valid.

Definition 4 (Policy compliance). *Let η be a history without framing events, then $\eta \models \varphi$ iff $\eta \notin L(\varphi)$.*

To check validity of a history expression we first need to solve a couple of technical problems. The first is because the semantics of a history expression may contain histories with nesting of the same policy framing. For instance, $H = \mu h. (\varphi[\alpha(r)h] + \epsilon)$ generates $[\varphi\alpha(r)[\varphi\alpha(r)]\varphi]$. This kind of nesting is *redundant* because the expressions monitored by the inner framings are already under the scope of the outermost one (in this case the second $\alpha(r)$ is already under the scope of the first φ). In addition, policies cannot predicate on the events of opening/closing a policy framing (because definition of validity in Subsection 5.2 uses $\eta^{-\square}$). More in detail, a history η has *redundant framing* whenever the active policies $ap(\eta')$ contain multiple occurrences of φ for some prefix η' of η . Redundant framings can be eliminated from a history expression without affecting the validity of its histories. For example, we can rewrite H as $H\downarrow = \varphi[\mu h. (\alpha(r)h + \epsilon)] + \epsilon$ that does not generate histories with redundant framings. Given H , there is a *regularisation* algorithm returning his regularised version $H\downarrow$ such that (i) each history in $\llbracket H\downarrow \rrbracket$ has no redundant framing, (ii) $H\downarrow$ is valid if and only if H is valid [20]. Hence, checking validity of a history expression H can be reduced to checking validity of a history expression $H\downarrow$.

The second technical step makes a local policy to speak globally, by transforming it so to trap the point of its activations. Let $\{\varphi_i\}$ be the set of all the policies φ_i occurring in H . From each φ_i it is possible to obtain a *framed automaton* φ_i^\square such that a history without redundant framings η is valid ($\models \eta$) if and only if $\eta \notin L(\bigcup \varphi_i^\square)$. The detailed construction of framed automata can be found in [20] and roughly works as follows. The framed automaton for the policy φ consists of two copies of φ . The first copy has no offending states and is used when the actions are performed while the policy is not active. The second copy is reached when the policy is activated by a framing event and has the same offending states of φ . Indeed, there are edges labelled with $[_\varphi$ from the first copy to the second and $]\varphi$ in the opposite direction. As a consequence, when a framing is activated, the offending states are reachable. Fig. 6 shows the framed automaton used to model check a simple policy φ_2 that prevents the occurrence of two consecutive actions α on the resource r . Clearly, the framed automaton only works on histories without redundant framing. Otherwise, it should record the number of nesting of policy framings to determine when the policy is active, and this is not a regular property.

Validating a regularised history expression H against the set of policies φ_i appearing therein amounts to verifying that $\llbracket H \downarrow \rrbracket \cap \bigcup L(\varphi_i^\square)$ is empty. This procedure is decidable, since $H \downarrow$ is context-free [18, 55], $\bigcup L(\varphi_i^\square)$ is regular; context-free languages are closed by intersection with regular languages; and emptiness of context-free languages is decidable.

Note that our approach fits into the standard *automata-based* model checking [92]. Therefore there is an efficient and fully automata-based method for checking validity, i.e. \models relation for a regularised history expression H [21].

Protocol compliance. We are now ready to check whether a program will well-behave when interacting with other parties through the bus. The idea is that the environment specifies P , and only accepts a user to join that follows P during the communication. We take a protocol P to be a sequence S of $send_\tau$ and $receive_\tau$ actions, possibly repeated (in symbols S^*), that designs the coordination interactions, as defined below:

$$P ::= S \mid S^*$$

$$S ::= \epsilon \mid send_\tau.S \mid receive_\tau.S$$

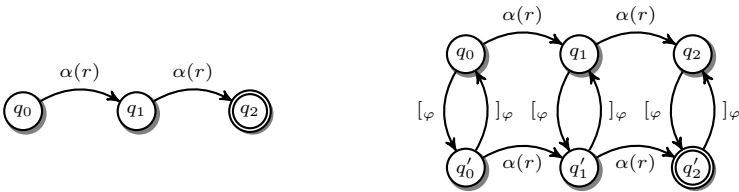


Fig. 6. On the left: a policy φ_2 that expresses that two consecutive actions α on r are forbidden. On the right: the framed automaton obtained from φ_2 .

A protocol P specifies the regular set of allowed interaction histories. We require a program to interact with the bus following the protocol, but we do not force the program to do the whole specified interaction. The language $L(P)$ of P turns out to be a prefix-closed set of histories, obtained by considering all the prefixes of the sequences defined by P . Then we only require that all the histories generated by a program (projected so that only $send_\tau$ and $receive_\tau$ appear) belong to $L(P)$.

Let H^{sr} be a projected history expression where all non $send_\tau, receive_\tau$ events have been removed. Then we define compliance to be:

Definition 5 (Protocol compliance). *Let e be an expression such that $\langle \Gamma, C \rangle \vdash e : \tau \triangleright H$, then e is compliant with P if $\llbracket H^{sr} \rrbracket \subseteq L(P)$.*

As for policy compliance, also protocol compliance can be established by using a decidable model checking procedure.

Note that, in our model, protocol compliance cannot be expressed only through the security policies introduced above. As a matter of fact, we have to check that H^{sr} does not include forbidden communication patterns, and this is a requirement much similar to a default-accept policy. Furthermore, we also need to check that some communication pattern in compliance with P *must* be done, cf. the check on the protocol compliance of the e-book reader program made in the example of Section 4.

References

1. Cloud cryptography group at Microsoft Research, <http://research.microsoft.com/en-us/projects/cryptocloud/>
2. UDDI technical white paper. Tech. rep., W3C (2000)
3. eXtensible Access Control Markup Language (XACML) Version 2.0. Tech. rep., OASIS (2005)
4. The future of cloud computing. Tech. rep., European Commision, Information Society and Media (2010)
5. Achermann, F., Lumpe, M., Schneider, J., Nierstrasz, O.: PICCOLA—a small composition language. In: Formal Methods for Distributed Processing, pp. 403–426. Cambridge University Press (2001)
6. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Order preserving encryption for numeric data. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, SIGMOD 2004, pp. 563–574. ACM, New York (2004), <http://doi.acm.org/10.1145/1007568.1007632>
7. Al-Neyadi, F., Abawajy, J.H.: Context-Based E-Health System Access Control Mechanism. In: Park, J.H., Zhan, J., Lee, C., Wang, G., Kim, T.-H., Yeo, S.-S. (eds.) ISA 2009. CCIS, vol. 36, pp. 68–77. Springer, Heidelberg (2009)
8. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services: Concepts, Architectures and Applications. Springer (2004)
9. Amazon.com Inc.: Aws customer agreement, <http://aws.amazon.com/agreement/>
10. Amazon.com Inc.: Overview of Amazon Web Services (2010), <http://aws.amazon.com/whitepapers/>
11. Anderson, S., et al.: Web Services Trust Language (WS-Trust) (2005)

12. Andrews, T., et al.: Business Process Execution Language for Web Services (BPEL4WS), Version 1.1 (2003)
13. Appeltauer, M., Hirschfeld, R., Haupt, M., Masuhara, H.: ContextJ: Context-oriented programming with java. *Computer Software* 28(1) (2011)
14. Atkinson, B., et al.: Web Services Security (WS-Security) (2002)
15. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. *Computer Networks* 54(15), 2787–2805 (2010)
16. Barbanera, F., Bugliesi, M., Dezani-Ciancaglini, M., Sassone, V.: Space-aware ambients and processes. *Theor. Comput. Sci.* 373(1-2), 41–69 (2007)
17. Barbanera, F., Dezani-Ciancaglini, M., Salvo, I., Sassone, V.: A type inference algorithm for secure ambients. *Electr. Notes Theor. Comput. Sci.* 62, 83–101 (2001)
18. Bartoletti, M., Degano, P., Ferrari, G.L.: Planning and verifying service composition. *Journal of Computer Security* 17(5), 799–837 (2009)
19. Bartoletti, M., Degano, P., Ferrari, G.L., Zunino, R.: Semantics-based design for secure web services. *IEEE Trans. Software Eng.* 34(1), 33–49 (2008)
20. Bartoletti, M., Degano, P., Ferrari, G.L., Zunino, R.: Local policies for resource usage analysis. *ACM Trans. Program. Lang. Syst.* 31(6) (2009)
21. Bartoletti, M., Zunino, R.: LocUsT: a tool for checking usage policies. *Tech. Rep. TR-08-07, Dip. Informatica, Univ. Pisa* (2008)
22. Bhargavan, K., Gordon, A.D., Narasamya, I.: Service Combinators for Farming Virtual Machines. In: Lea, D., Zavattaro, G. (eds.) *COORDINATION 2008. LNCS*, vol. 5052, pp. 33–49. Springer, Heidelberg (2008)
23. Blanchet, B.: Security Protocol Verification: Symbolic and Computational Models. In: Degano, P., Guttman, J.D. (eds.) *POST 2012. LNCS*, vol. 7215, pp. 3–29. Springer, Heidelberg (2012)
24. Bodei, C., Dinh, V.D., Ferrari, G.L.: Safer in the clouds (extended abstract). In: Bliudze, S., Bruni, R., Grohmann, D., Silva, A. (eds.) *ICE. EPTCS*, vol. 38, pp. 45–49 (2010)
25. Bodei, C., Dinh, V.D., Ferrari, G.L.: Predicting global usages of resources endowed with local policies. In: Mousavi, M.R., Ravara, A. (eds.) *FOCLASA. EPTCS*, vol. 58, pp. 49–64 (2011)
26. Bonelli, E., Compagnoni, A., Gunter, E.: Typechecking safe process synchronization. In: *Proc. Foundations of Global Ubiquitous Computing. ENTCS*, vol. 138(1) (2005)
27. Boreale, M., Bruni, R., Caires, L., De Nicola, R., Lanese, I., Loret, M., Martins, F., Montanari, U., Ravara, A., Sangiorgi, D., Vasconcelos, V., Zavattaro, G.: SCC: A Service Centered Calculus. In: Bravetti, M., Núñez, M., Zavattaro, G. (eds.) *WS-FM 2006. LNCS*, vol. 4184, pp. 38–57. Springer, Heidelberg (2006)
28. Box, D., et al.: Simple Object Access Protocol (SOAP) 1.1. *WRC Note* (2000)
29. Box, D., et al.: Web Services Policy Framework (WS-Policy) (2002)
30. Braghin, C., Cortesi, A.: Flow-sensitive leakage analysis in mobile ambients. *Electr. Notes Theor. Comput. Sci.* 128(5), 17–25 (2005)
31. Braghin, C., Cortesi, A., Focardi, R.: Security boundaries in mobile ambients. *Computer Languages, Systems & Structures* 28(1), 101–127 (2002), <http://www.sciencedirect.com/science/article/pii/S0096055102000097>
32. Bravetti, M., Di Giusto, C., Pérez, J.A., Zavattaro, G.: Adaptable Processes (Extended Abstract). In: Bruni, R., Dingel, J. (eds.) *FMOODS/FORTE 2011. LNCS*, vol. 6722, pp. 90–105. Springer, Heidelberg (2011)
33. Brogi, A., Canal, C., Pimentel, E.: Behavioural Types and Component Adaptation. In: Rattray, C., Maharaj, S., Shankland, C. (eds.) *AMAST 2004. LNCS*, vol. 3116, pp. 42–56. Springer, Heidelberg (2004)

34. Bruni, R.: Calculi for Service-Oriented Computing. In: Bernardo, M., Padovani, L., Zavattaro, G. (eds.) SFM 2009. LNCS, vol. 5569, pp. 1–41. Springer, Heidelberg (2009)
35. Bruni, R., Corradini, A., Gadducci, F., Lluch Lafuente, A., Vandin, A.: A Conceptual Framework for Adaptation. In: de Lara, J., Zisman, A. (eds.) FASE 2012. LNCS, vol. 7212, pp. 240–254. Springer, Heidelberg (2012)
36. Bucur, D., Nielsen, M.: Secure Data Flow in a Calculus for Context Awareness. In: Degano, P., De Nicola, R., Meseguer, J. (eds.) Montanari Festschrift. LNCS, vol. 5065, pp. 439–456. Springer, Heidelberg (2008)
37. Bugliesi, M., Castagna, G., Crafa, S.: Reasoning about Security in Mobile Ambients. In: Larsen, K.G., Nielsen, M. (eds.) CONCUR 2001. LNCS, vol. 2154, pp. 102–120. Springer, Heidelberg (2001)
38. Caires, L., De Nicola, R., Pugliese, R., Vasconcelos, V.T., Zavattaro, G.: Core Calculi for Service-Oriented Computing. In: Wirsing, M., Hölzl, M. (eds.) SENSORIA 2011. LNCS, vol. 6582, pp. 153–188. Springer, Heidelberg (2011)
39. Campbell, R., Al-Muhtadi, J., Naldurg, P., Sampemane, G., Mickunas, M.D.: Towards Security and Privacy for Pervasive Computing. In: Okada, M., Babu, C. S., Scedrov, A., Tokuda, H. (eds.) ISSS 2002. LNCS, vol. 2609, pp. 1–15. Springer, Heidelberg (2003), <http://dl.acm.org/citation.cfm?id=1765533.1765535>
40. Carbone, M., Honda, K., Yoshida, N.: Structured Communication-Centred Programming for Web Services. In: De Nicola, R. (ed.) ESOP 2007. LNCS, vol. 4421, pp. 2–17. Springer, Heidelberg (2007)
41. Cardelli, L., Gordon, A.D.: Mobile Ambients. In: Nivat, M. (ed.) FOS-SACS 1998. LNCS, vol. 1378, pp. 140–155. Springer, Heidelberg (1998), <http://dx.doi.org/10.1007/BFb0053547>
42. Chen, G., Kotz, D.: A survey of context-aware mobile computing research. Tech. rep., Dartmouth College, Hanover, NH, USA (2000)
43. Chen, H., Finin, T., Joshi, A.: An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review* 18(03), 197–207 (2003)
44. Cheng, B.H.C., de Lemos, R., Giese, H., Inverardi, P., Magee, J., Andersson, J., Becker, B., Bencomo, N., Brun, Y., Cukic, B., Di Marzo Serugendo, G., Dustdar, S., Finkelstein, A., Gacek, C., Geijs, K., Grassi, V., Karsai, G., Kienle, H.M., Kramer, J., Litoiu, M., Malek, S., Mirandola, R., Müller, H.A., Park, S., Shaw, M., Tichy, M., Tivoli, M., Weyns, D., Whittle, J.: Software Engineering for Self-Adaptive Systems: A Research Roadmap. In: Cheng, B.H.C., de Lemos, R., Giese, H., Inverardi, P., Magee, J. (eds.) *Software Engineering for Self-Adaptive Systems*. LNCS, vol. 5525, pp. 1–26. Springer, Heidelberg (2009)
45. Chinnici, R., Gudgina, M., Moreau, J., Weerawarana, S.: Web Service Description Language (WSDL), Version 1.2 (2002)
46. Ciancia, V., Ferrari, G.L., Guanciale, R., Strollo, D.: Event based choreography. *Sci. Comput. Program.* 75(10), 848–878 (2010)
47. Clarke, D., Costanza, P., Tanter, E.: How should context-escaping closures proceed? In: *International Workshop on Context-Oriented Programming, COP 2009*, pp. 1:1–1:6. ACM, New York (2009), <http://doi.acm.org/10.1145/1562112.1562113>
48. Clarke, D., Sergey, I.: A semantics for context-oriented programming with layers. In: *International Workshop on Context-Oriented Programming, COP 2009*, pp. 10:1–10:6. ACM, New York (2009), <http://doi.acm.org/10.1145/1562112.1562122>
49. Costanza, P.: Language constructs for context-oriented programming. In: *Proceedings of the Dynamic Languages Symposium*, pp. 1–10. ACM Press (2005)

50. Curbera, F., Khalaf, R., Mukhi, N., Tai, S., Weerawarane, S.: The next step in web services. *Communications of the ACM* 46(10) (2003)
51. De Nicola, R., Ferrari, G., Loretto, M., Pugliese, R.: A language-based approach to autonomic computing. In: *FMCO 2011. LNCS*. Springer (to appear, 2012)
52. Degano, P., Ferrari, G.L., Galletta, L., Mezzetti, G.: Typing context-dependent behavioural variations. In: *PLACES 2012 (vol. to appear in EPTCS 2012)*
53. Degano, P., Ferrari, G.-L., Galletta, L., Mezzetti, G.: Types for Coordinating Secure Behavioural Variations. In: Sirjani, M. (ed.) *COORDINATION 2012. LNCS*, vol. 7274, pp. 261–276. Springer, Heidelberg (2012)
54. Deng, M., Cock, D.D., Preneel, B.: Towards a cross-context identity management framework in e-health. *Online Information Review* 33(3), 422–442 (2009)
55. Esparza, J.: Decidability of model checking for infinite-state concurrent systems. *Acta Inf.* 34(2), 85–107 (1997)
56. Ferrari, G., Guanciale, R., Strollo, D.: JSCL: A Middleware for Service Coordination. In: Najm, E., Pradat-Peyre, J.-F., Donzeau-Gouge, V.V. (eds.) *FORTE 2006. LNCS*, vol. 4229, pp. 46–60. Springer, Heidelberg (2006)
57. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pp. 169–178. ACM (2009)
58. Greenfield, A.: *Everyware: The dawning age of ubiquitous computing*. Peachpit Press (2006)
59. Gu, T., Wang, X., Pung, H., Zhang, D.: An ontology-based context model in intelligent environments. In: *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, vol. 2004, pp. 270–275 (2004)
60. Guidi, C., Lucchi, R., Gorrieri, R., Busi, N., Zavattaro, G.: SOCK: A Calculus for Service Oriented Computing. In: Dan, A., Lamersdorf, W. (eds.) *ICSOC 2006. LNCS*, vol. 4294, pp. 327–338. Springer, Heidelberg (2006)
61. Hamlen, K.W., Morrisett, J.G., Schneider, F.B.: Computability classes for enforcement mechanisms. *ACM Trans. on Programming Languages and Systems* 28(1), 175–205 (2006)
62. Heer, T., Garcia-Morchon, O., Hummen, R., Keoh, S., Kumar, S., Wehrle, K.: Security challenges in the ip-based internet of things. *Wireless Personal Communications*, 1–16 (2011)
63. Hirschfeld, R., Costanza, P., Nierstrasz, O.: Context-oriented programming. *Journal of Object Technology* 7(3), 125–151 (2008)
64. Hirschfeld, R., Igarashi, A., Masuhara, H.: ContextFJ: a minimal core calculus for context-oriented programming. In: *Proceedings of the 10th International Workshop on Foundations of Aspect-oriented Languages, FOAL 2011*, pp. 19–23. ACM, New York (2011), <http://doi.acm.org/10.1145/1960510.1960515>
65. Honda, K., Vasconcelos, V.T., Kubo, M.: Language Primitives and Type Discipline for Structured Communication-Based Programming. In: Hankin, C. (ed.) *ESOP 1998. LNCS*, vol. 1381, pp. 122–138. Springer, Heidelberg (1998)
66. Hulsebosch, R., Salden, A., Bargh, M., Ebben, P., Reitsma, J.: Context sensitive access control. In: *Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies*, pp. 111–119. ACM (2005)
67. IBM: An architectural blueprint for autonomic computing. Tech. rep. (2005)
68. Igarashi, A., Pierce, B.C., Wadler, P.: Featherweight java: a minimal core calculus for Java and GJ. *ACM Trans. Program. Lang. Syst.* 23(3), 396–450 (2001)

69. Kamina, T., Aotani, T., Masuhara, H.: Eventcj: a context-oriented programming language with declarative event-based context transition. In: Proceedings of the Tenth International Conference on Aspect-oriented Software Development, AOSD 2011, pp. 253–264. ACM, New York (2011), <http://doi.acm.org/10.1145/1960275.1960305>
70. Kavantzaz, N., et al.: Web Service Coreography Description Language, <http://www.w3.org/TR/ws-cdl-10/>
71. Kelly, L.: The security threats of technology ubiquity, <http://www.computerweekly.com/feature/The-security-threats-of-technology-ubiquity>
72. Khalaf, R., Mukhi, N., Weerawarana, S.: Service oriented composition in BPEL4WS. In: Proc. WWW (2003)
73. Lapadula, A., Pugliese, R., Tiezzi, F.: A Calculus for Orchestration of Web Services. In: De Nicola, R. (ed.) ESOP 2007. LNCS, vol. 4421, pp. 33–47. Springer, Heidelberg (2007)
74. Lazovik, A., Aiello, M., Gennari, R.: Encoding Requests to Web Service Compositions as Constraints. In: van Beek, P. (ed.) CP 2005. LNCS, vol. 3709, pp. 782–786. Springer, Heidelberg (2005)
75. Levi, F., Sangiorgi, D.: Mobile safe ambients. ACM Trans. Program. Lang. Syst. 25(1), 1–69 (2003)
76. Masi, M., Pugliese, R., Tiezzi, F.: Formalisation and Implementation of the XACML Access Control Mechanism. In: Barthe, G., Livshits, B., Scandariato, R. (eds.) ESSoS 2012. LNCS, vol. 7159, pp. 60–74. Springer, Heidelberg (2012)
77. Misra, J.: A programming model for the orchestration of web services. In: 2nd International Conference on Software Engineering and Formal Methods, SEFM 2004 (2004)
78. Naehrig, M., Lauter, K., Vaikuntanathan, V.: Can homomorphic encryption be practical? In: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, pp. 113–124. ACM (2011)
79. Papazoglou, M.P.: Service-oriented computing: Concepts, characteristics and directions. In: WISE (2003)
80. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: a research roadmap. Int. J. Cooperative Inf. Syst. 17(2), 223–255 (2008)
81. Papazoglou, M., Georgakopoulos, D.: Special issue on service oriented computing. Communications of the ACM 46(10) (2003)
82. Pelusi, L., Passarella, A., Conti, M.: Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. IEEE Communications Magazine 44(11), 134–141 (2006)
83. Pfleeger, C., Pfleeger, S.: Security in computing. Prentice Hall (2003)
84. Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R., Nahrstedt, K.: Gaia: a middleware platform for active spaces. ACM SIGMOBILE Mobile Computing and Communications Review 6(4), 65–67 (2002)
85. Sangiorgi, D., Walker, D.: The Pi-Calculus - a theory of mobile processes. Cambridge University Press (2001)
86. Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: Proceedings of the Workshop on Mobile Computing Systems and Applications, pp. 85–90. IEEE Computer Society (1994)
87. Skalka, C., Smith, S., Horn, D.V.: Types and trace effects of higher order programs. Journal of Functional Programming 18(2), 179–249 (2008)
88. Stal, M.: Web services: Beyond component-based computing. Communications of the ACM 55(10) (2002)

89. Sweeney, L., et al.: k-anonymity: A model for protecting privacy. *International Journal of Uncertainty Fuzziness and Knowledge Based Systems* 10(5), 557–570 (2002)
90. Takabi, H., Joshi, J.B.D., Ahn, G.J.: Security and privacy challenges in cloud computing environments. *IEEE Security & Privacy* 8(6), 24–31 (2010)
91. Vallecillo, A., Vansconcelos, V., Ravara, A.: Typing the behaviours of objects and components using session types. In: *Proc. of FOCLASA* (2002)
92. Vardi, M.Y., Wolper, P.: An automata-theoretic approach to automatic program verification (preliminary report). In: *LICS*, pp. 332–344. IEEE Computer Society (1986)
93. Vitek, J., Castagna, G.: Seal: A Framework for Secure Mobile Computations. In: Bal, H.E., Belkhouche, B., Cardelli, L. (eds.) *ICCL 1998 Workshop*. LNCS, vol. 1686, pp. 47–77. Springer, Heidelberg (1999)
94. Vogels, W.: Web services are not distributed objects. *IEEE Internet Computing* 7(6) (2003)
95. Wang, X.H., Zhang, D.Q., Gu, T., Pung, H.K.: Ontology based context modeling and reasoning using owl. In: *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pp. 18–22. IEEE (2004)
96. Weber, R.H.: Internet of things-new security and privacy challenges. *Computer Law & Security Review* 26(1), 23–30 (2010)
97. Wirsing, M., Hözl, M.M. (eds.): *SENSORIA*. LNCS, vol. 6582. Springer, Heidelberg (2011)
98. Wrona, K., Gomez, L.: Context-aware security and secure context-awareness in ubiquitous computing environments. In: *XXI Autumn Meeting of Polish Information Processing Society* (2005)
99. Yang, M., Sassone, V., Hamadou, S.: A Game-Theoretic Analysis of Cooperation in Anonymity Networks. In: Degano, P., Guttman, J.D. (eds.) *POST 2012*. LNCS, vol. 7215, pp. 269–289. Springer, Heidelberg (2012)
100. Zhang, G., Parashar, M.: Dynamic context-aware access control for grid applications. In: *Proceedings of the Fourth International Workshop on Grid Computing*, pp. 101–108. IEEE (2003)

Designing Smart Cities: Security Issues

Young Im Cho

Dept. of Computer Science, The University of Suwon, Korea
ycho@suwon.ac.kr

Abstract. Smart city means an intelligent city called post-ubiquitous city. However, to be a smart city, there are some issues to consider carefully. The first is the security issues having platform and intelligent surveillance function and analysis function as well as inference. The second is the governance issues between government and cities which are developing. And the third is the service issues to be realized in the cities. In this paper, we are mainly concentrated on the first security issue. Here we propose a speech recognition technology in emergency situation for secure cities. For the emergency detection in general CCTVs(closed circuit television) environment of our daily life, the monitoring by only images through CCTVs information occurs some problems especially in emergency state. Therefore for detecting emergency state dynamically through CCTVs as well as resolving some problems, we propose a detection and recognition method for emergency and non-emergency speech by Gaussian Mixture Models(GMM). The proposed method determines whether input speech is emergency or non-emergency speech by global GMM firstly. If this is an emergency speech, then local GMM is performed secondly to classify the type of emergency speech. The proposed method is tested and verified by emergency and non-emergency speeches in various environmental conditions. Also, we discuss about the platform issues having analysis and inference function of big data in smart city.

Keywords: Smart city, security, big data, speech recognition, platform.

1 Introduction

In 1988, Mark Weiser introduced a new phrase to describe a new paradigm of computing. That phrase was ubiquitous computing and it refers to 'smart' and networked devices embedded within our environments. Now we are living in the era of ubiquitous computing. Ubiquitous computing is roughly the opposite of virtual reality. Usually, virtual reality puts people inside a computer-generated world, but ubiquitous computing forces the computer to live out here in the world with people. Virtual reality is primarily a horse power problem, but ubiquitous computing is a very difficult integration of human factors, computer science, engineering, and social sciences. Ubiquitous computing is a post-desktop model of human-computer interaction in which information processing has been thoroughly integrated into everyday objects and activities. The core technology of ubiquitous computing is an autonomic collaboration model for ubiquitous fusion services in ubiquitous

computing environment. To do some action in ubiquitous computing, many elements coordinate to complete such action. Ubiquitous networking in ubiquitous computing is completed by the collaboration of many types of networks, city technologies, city sociology and governance[1,2,3].

Ubiquitous city is a city with a virtual overlay on top of its physical construct that allows people to interact with their environment everywhere and in real time for their convenient better quality life.

Nowadays, the pervasiveness of smart phones, tablets and laptops has led to instant communication via the web, and that has profoundly changed human behavior. We make a blog, tweet and share status messages to a global audience with little cost. Information and social or otherwise is overabundant and cheap. The internet or the ubiquitous platform has been with us for a while and all of that is really obvious.

In short, smart city is a ubiquitous city having high intelligent capability with a lot of smart device and software. Nowadays smart city is more popular than ubiquitous city. Smart city is not a 20th century city with a few computers and screens thrown at it. It is not a just conventional city with its citizens carrying mobile phones. So, it needs a different planning or designing concept.

Actually, there are a lot of different typed-data to be analyzed especially in smart city. Because many smart devices and software produce a lot of data with or without our intention. So, it is very important to deal with security strategy for abundant data in smart city.

In this paper, we will survey and discuss some important security issues when we designing smart cities at this paper.

2 The Concept of Smart City

In computerized generation, industrial society paradigm was popular. That means that making some products by industry was so important at that time. After that as time goes by, in 'e' generation and 'u' generation, information society paradigm is more popular. That means that information is very important thing[3].

Main keywords in new paradigm are the power of people, collective intelligence and new ecosystem. The power of people use mobile device and social media increasing the power of people in life. Collective intelligence is the creation of value by collective intelligence as well as collective power. New ecosystem is the creation of additional value by new ecosystem using open as well as sharing.

Actually, the goal of ubiquitous computing is to make the safety, security and peace of mind society. The key technology to be a ubiquitous society is ubiquitous sensor network, in short USN. There are many application areas that use USN, such as home control, health medical care, welfare, hospital, school or education, traffic, culture, sports, environments, tracking of produce, anti-disaster, city security, etc.

Nowadays smart society is dawning. If we disregard this changes or needs, we will degenerate from information ages. Smart paradigm is just an extension of ubiquitous paradigm, so called post-ubiquitous age.

There are four types of viewpoints in ubiquitous computing. In the aspect of computing paradigm, the technology is changed from static community to automatic

community. In the aspect of technical principle, the technology is changed from situation aware to self growing. However, in the aspect of application domain, the technology is changed from healthcare to environment preservation. The fourth aspect concerns the application area, the technology is changed from home or building to society[1,3].

Smart city can be identified (and ranked) along six main axes or dimensions[4,5]. These axes are smart economy, smart mobility, smart environment, smart people, smart living, and smart governance. These six axes connect with traditional regional and neoclassical theories of urban growth and development. In particular, the axes are based -respectively- on theories of regional competitiveness, transport and ICT economics, natural resources, human and social capital, quality of life, and participation of citizens in the governance of cities.

A city can be defined as 'smart' when investments in human and social capital and traditional and modern ICT communication infrastructure fuel sustainable economic development and a high quality of life, with a wise management of natural resources, through participatory governance.

Why the smart city is necessary? It is for maximizing the utilization and functionality of space as well as increasing the value of the city. Also, it is for providing a better quality of life. For that linking the city internationally is very important. Nowadays the environment of our life is very important, so reducing many kinds of pollution is another goal of smart city.

Among many application areas of ubiquitous, smart city is the constructed city by ubiquitous technologies and paradigm. Usually many ubiquitous services and hybrid technologies using RFID, USN, IPv6, sensing devices, monitoring, auto-control, real-time management etc. make smart city.

However, there are three important issues to be s smart city as of now. Firstly, the most important issue is the research about the infra structure of smart city such as platform, security, service scenario etc. The second issue is the research about the paradigm of smart city such as role play between government and local government to perform smart city etc. The third issue is the research about the consulting of smart city such as the best service model according to many types of organs, and business model, and so on.

In short, smart city is a combination of safety and security for the sake of peaceful society. Now many services of ICT mobile devices are provided with personalized manners. For completion of real ubiquitous space through smart city, the standard model and some platforms should be a prerequisite condition for success.

In this paper, our research is mainly focused on the first issue such as security issue and platform in designing smart city.

3 Security Issues

3.1 Speech Recognition Issue

In smart city, how to implement secure city is important issue. To give a smart capability in smart city, it is usually used CCTV(closed circuit television) for

monitoring the city. However, many CCTVs catch or sense a lot of image information from the defined area rather than speech sound. However, even though they sense speech sound, it is hard to manage the emergency state or situation. For example, if somebody is in some emergency situation, he or she shouts as loud as possible. In this case, because the CCTVs only catch the images from the restricted or defined area, in case of the emergency speech or sound is a far from the CCTVs, it is hard to manage the emergency situation quickly. But if the sound is not an emergency sound, it is no problem. Only emergency sound is to be managed as soon as possible for secure city. To realize this concept is necessary for secure smart cities.

However, the difficult for speech recognition is the background noise. As the noise is the main cause for decreasing the performance, the place or environment is very important in speech recognition[6]. Here, we propose a simulated classification method for emergency and non-emergency speech.

For the purpose that, we use GMM(Gaussian Mixture Model). GMM is among the most statistically mature method for clustering (though they are also used intensively for density estimation). We introduce the concept of clustering, and see how one form of clustering in which we assume that individual data points are generated by first choosing one of a set of multivariate Gaussians and then sampling from them can be a well-defined computational operation. We see how to learn such a thing from data, and we discover that an optimization approach. This optimization method is called Expectation Maximization (EM). We spend some time giving a few high level explanations and demonstrations of EM, which turns out to be valuable for many other algorithms beyond Gaussian Mixture[7,8,9].

Therefore the proposed method determines whether input speech is emergency or non-emergency speech by global GMM. If the sensing speech is classified as an emergency speech, then local GMM is performed to recognize the type of emergency speech. By the simulation result, the proposed method is well classified whether the speech has noise or not([Fig.3]).

One of the key factors in the speech recognition is quite different from the controlled environment of the speech laboratory. However, the surrounding noises are a particularly difficult problem in the real speech recognition. The difference in the controlled environment and the real environment in speech recognition comes into play in three distinct processes: signal process, feature space process, and model process. Of these three processes, the difference is most evident in the signal process[10]. Here, the noise in the speech data after the signal process is filtered by a novel digital filtering system. A FIR(Finite Impulse Response) filter [6,10] is first used to separate the speech region and the noise region, and then a Wiener filter is used to improve the overall speech recognition ([Fig.1]).

Generally, the speech recognition system is configured by six stages. In stage 1, voice data are inputted by converting the audio signals into the electrical digital signals. In stage 2, the voice signals are separated from the surrounding noises. In stage 3, useful traits in speech recognition are extracted by using a speech recognition model. In stage 4, a standard speech pattern database is formed by speech recognition training. In stage 5, the new voice data are compared to the standard speech pattern a base, and the closest match is searched. In the final stage of 6, the matched result is put to use through the user interface.

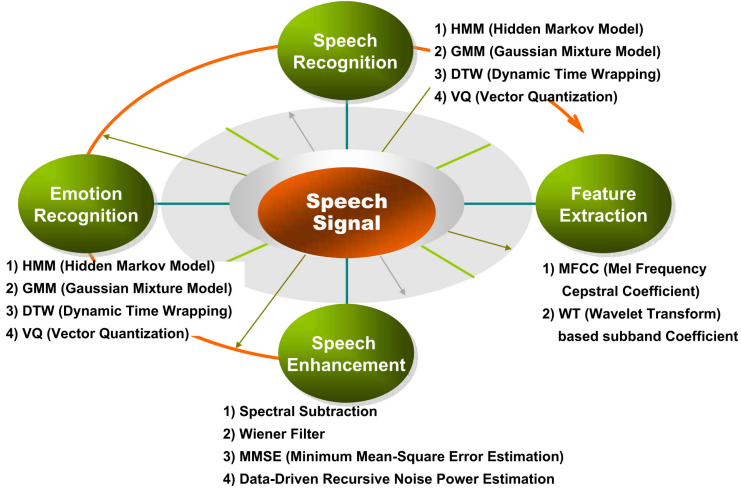


Fig. 1. Speech recognition summary

In the preprocessing (noise elimination) stage of 2, analog audio signals from a CCTVs or sensors are digitized and then fed to the digital filter. The digital filter, which is widely used and proven, selects the pass-band and filter out the stop-band.

Depending on the presence of feedback processes, the digital filter is divided into IIR(Infinite Impulse Response) and FIR filters. The latter is known to be less error-prone. For noise elimination, the Wiener and Kalman filters[4] are widely used. In emergency situations that require accurate interpretation of a rather brief voice data, the Wiener filter is usually preferred. The general model-based Wiener filtering process can be expressed as follows:

$$\hat{s}(t) = g(t) * (s(t) + n(t)) \quad (1)$$

Where, $\hat{s}(t)$ is the speech to be recognized, $s(t)$ is the speech data containing noise, $n(t)$ is the noise, and $g(t)$ is the Wiener filter. In Eq. (1), $\hat{s}(t)$ is being sought. In it, an estimate of $n(t)$ is derived from $s(t)$, and then the approximate value of $\hat{s}(t)$ is obtained by using $n(t)$. In order to achieve a better approximation of $\hat{s}(t)$, the GMM as expressed below in Eq.(2) is used. It expresses mathematically the general characteristic of speech data.

$$P(s) = \sum_k^K p(k) N(s; \mu_k; \sum_k k) \quad (2)$$

Based on Eq. (2), the model-based Wiener filter is designed per following steps: In the inputted current frame, the noise region is determined by a statistically-based VAD. In the noise region found, the noise model is renewed to the previous value. In the preprocess-WF block, a temporally noise-free clean speech is estimated using the

decision-directed Wiener filter. Using the estimated values from the previous step, the Gaussian post probabilities of the GMM are calculated. In the final, the probabilities are used to estimate the noise-free clean speech. The estimated noise-free speech and the noise model are used to design the final Wiener filter. The current frame is processed using the Wiener filter designed, and the noise-free clean speech is obtained. Then the above five steps are repeated for the next frame.

Because of the reason of the performance, we propose a fast recognition filtering method for emergency detection. The basic concept is to selectively use the audio signal being transmitted from the CCTVs'. That is, from the transmitted signal, only the audio energy spectrum that is relevant to the speech is to be selected, digitized, and saved for further analysis. A high-performance FIR Wiener filter can be used to digitally filter out the unwanted portion of the audio signal, prior to actual speech recognition.

Fig.2 is the proposed detection and recognition algorithm of emergency speech. After speech sound is detected, noise filtering and speech detection algorithms are applied. And then, through the feature extraction and similarity method by GMM we can decide whether the sound is emergency sound or not. After determination, we can apply the sound to emergency models in DB, and then finally we decide the sound and display on the dedicated screen.

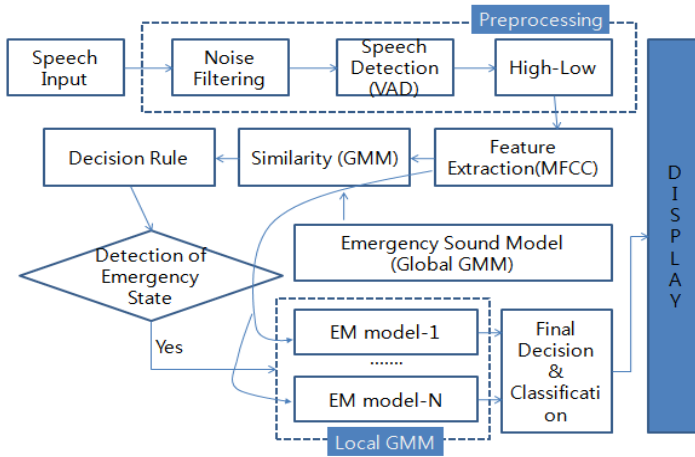


Fig. 2. The proposed detection and recognition process for emergency speech

In this paper we construct the global GMM and local GMM. Global GMM is for detection whether the sound is emergency sound or not. Global GMM is constructed based on the feature extraction of emergency sound. If the sound is in emergency sound, the type or the speech of the emergency sound is determined in local GMM process.

Finally, Fig.3 is shown the results of our method.

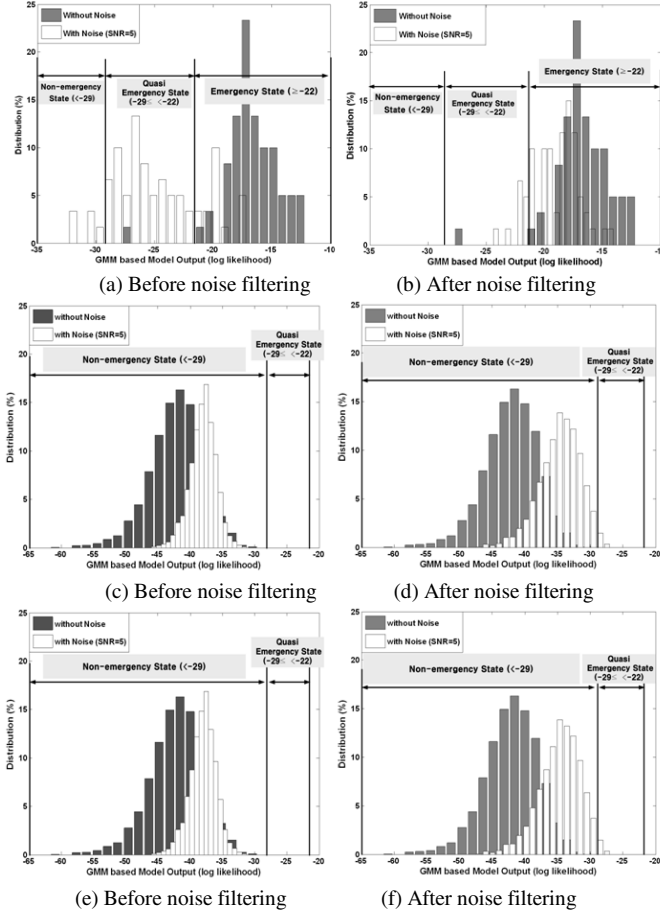


Fig. 3. The estimation results of GMM (upper: in case of white noise, middle : in case of having general noise, lower : in case of having automobile noise)

3.2 Platform Issue

Smart city has to have the following functions such as instrumented, interconnected and intelligent functions. To realize those functions, smart city has to have a standard protocol(platform) to manage many kinds of activities. Platform is the structure and law for easy transaction among groups. Usually the platform of a smart city is as important as services. Platform is a system structure for information service of special area. The role of platform is for communication between side A and side B. For the purpose of that, they have to have platform having architecture and some rule for protocol. If not, it is hard to communicate each other as well as share some information. The goal of smart city is for collaboration. To reach this goal, the platform should prepare first of all.

Fig.4 is a smart city platform which is developed by Korea government at 2009[11] firstly. But it is updated as of now. In Fig.4, the number 1 is for infra interface, number 2 is for external service interfaces. The number 3 is for core utility services, so security functions are included here. The number 4 is for workflow management, so situation awareness or event management functions are implemented here. The number 5 is for information hub, and finally number 6 is for integrated databases. From the platform, we can realize that the number 3 and 4 modules have lots of important or core functions to be a smart city.

Now intelligent surveillance technologies are developed in the platform. So, the framework of platform is like Fig.5. Smart platform should have analysis capability of many kinds of data or situation. So inference capability should include here.

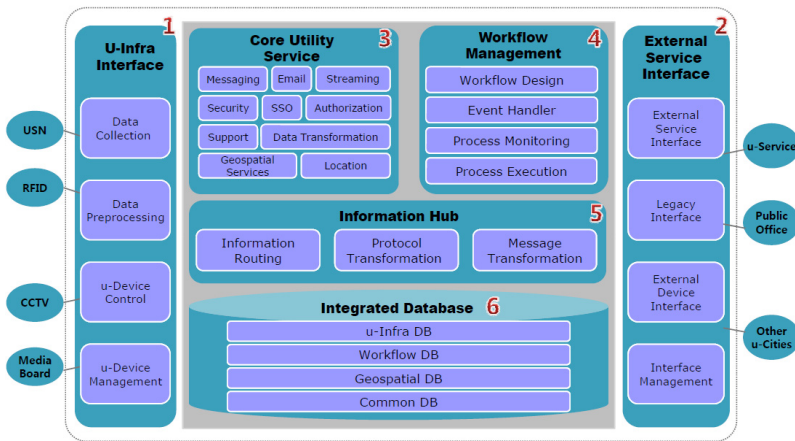


Fig. 4. Smart city platform

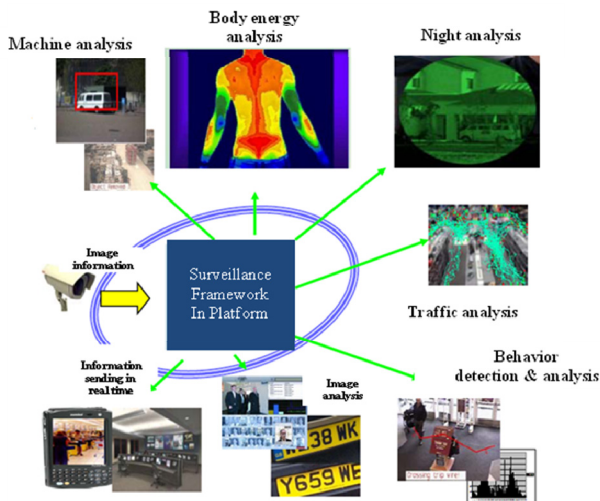


Fig. 5. Surveillance framework in platform

In information technology, the ‘big data’ is a loosely-defined term used to describe data sets so large and complex that they become awkward to work with using on-hand database management tools. Difficulties include capture, storage, search, sharing, analysis, and visualization. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data[12].

Data from citizens, systems, and general things are the single most scalable resources available to smart city stakeholders today. This big data is constantly captured through sensors and from open data sources. More and more data services for city officials, utility services, and citizens become available, which allows efficient access and use of big data, a necessary requirement for smart cities.

Advances in digital sensors, communications, computation, and storage have created huge collections of data, capturing information of value to business, science, government, and society[13]. To many, the term ‘big data’ refers to algorithms and software programs that help companies or researchers make discoveries and unearth trends by allowing them to visualize and analyze information better. But it has another meaning too. Big data literally means big data, dizzying amounts of customer records, sound recordings, images, text messages, Facebook comments and technical information that has to be stored, retrieved and understood in its proper context to be any good to anyone.

Historically, data analytics software hasn’t had the capability to take a large data set and use all of it—or at least most of it—to compile a complete analysis for a query. Instead, it has relied on representative samplings, or subsets, of the information to render these reports, even though analyzing more information produces more accurate results.

Our approach is changing with the emergence of new big data analytics engines, such as Apache Hadoop, LexisNexis’HPCC Systems. These new platforms are causing ‘the disappearing role of summarization’. So, we propose a comprehensive personalized information retrieval platform to support the big data analysis and process in Fig.6.

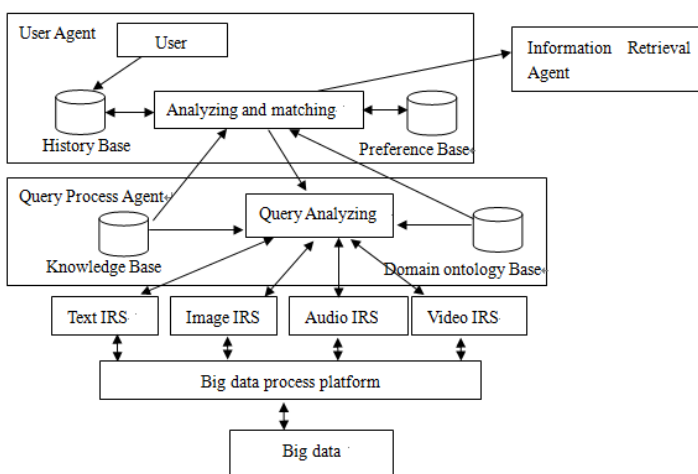


Fig. 6. Comprehensive personalized information retrieval platform

There are two main functions of user agent: first, it interacts with users and accepts the users' retrieval query and returns the retrieval results back to users; second, it analyses the received user's query and matching the preference base of each user, then returns the estimated proposals back to the user for choosing the best one and finally sends the refined query to the Information Retrieval Agent. The main function of the Query Process Agent is to analyze which domain the query may belong to according to the domain ontology base and knowledge base. For different type of data, we use Map/Reduce function to analyze and process data. Map/Reduce computing model, reasoning algorithm was designed to process mass data. Query was also transformed into Finite Semantic Graph, and semantic matched with full graph. Different kinds of data should use different Map/Reduce function, and find the optimal Map/Reduce function.

4 Conclusion

Urban performance currently depends not only on the city's endowment of hard infrastructure, but also, and increasingly so, on the availability and quality of knowledge communication and social infrastructure. The latter form of capital is decisive for urban competitiveness. It is against this background that the concept of the 'smart city' has been introduced as a strategic device to encompass modern urban production factors in a common framework and to highlight the growing importance of ICT technologies, social and environmental capital in profiling the competitiveness of cities. The significance of these two assets - social and environmental capital - itself goes a long way to distinguish smart cities from their more technology-laden counterparts, drawing a clear line between them and what goes under the name of either digital or intelligent cities.

In this paper, we deal with the speech recognition issue firstly. This is for the recognition of emergency or non-emergency sound to detect the emergency situation in smart city. Unlike the controlled environment where a speech recognition system can easily filter out the extraneous noises, it is rather difficult in the real environment where sensors, such as CCTVs, collect abundant noises from various human, mechanical, and natural sources. The success of speech recognition in the real environment thus depends critically on how well these noises are filtered. Just as important, the processing time for noise filtering needs to be reduced, as time is the most critical element in emergency situations. Thus, effective noise filtering combined with fast processing time is considered to be the essence of speech recognition. Towards these goals, an improved speech recognition system is proposed in this work. The system has the FIR and Wiener filters as the key elements and effectively filters out the extraneous noises and produces clean noise-free speech data in a reasonable time. Here, we proposed the classification method between emergency sound and non-emergency sound by GMM. This method can do that if some sound is in emergency sound, the sound can be recognized by local GMM algorithm. From the simulation results, we can conclude that the recognition ratio is more than 80%, so that this method can apply in real situation.

And secondly we discuss with platform issue having the functions for not only analysis but also inference. Big data is another inference issue. We discuss the structure of big data platform here.

References

- [1] Cho, Y.I.: U-Society and U-Convergence Paradigm. In: 2008 Asia Pacific Woman Leaders Forum for Science & Technology, September 3, pp. 183–204 (2008)
- [2] Roman, M., et al.: A Middleware Infrastructure for Active Spaces. IEEE Pervasive Computing (2002)
- [3] Cho, Y.I.: Practical Approaches for Ubiquitous Convergence System. In: Joint 4th International Conference on Soft Computing and Intelligent Systems and 9th International Symposium on Advanced Intelligent Systems, September 17–21 (2008)
- [4] Caragliu, A., Del Bo, C., Nijkamp, P.: Smart cities in Europe, Serie Research Memoranda 0048 (VU University Amsterdam, Faculty of Economics, Business Administration and Econometrics) (2009), <http://ideas.repec.org/p/dgr/vuarem/2009-48.html>
- [5] Giffinger, R., Fertner, C., Kramar, H., Kalasek, R., Pichler-Milanovic, N., Meijers, E.: Smart cities – Ranking of European medium-sized cities. Centre of Regional Science, Vienna (2009), <http://www.smart-cities.eu/>, http://www.smart-cities.eu/download/smart_cities_final_report.pdf (retrieved November 11, 2009)
- [6] Boll, S.F.: Suppression of acoustic noise in speech using spectral subtraction. IEEE Trans. ASSP 37(2), 113–120 (1979)
- [7] Doclo, S., Dong, R., Klasen, T.J., Wouters, J., Haykin, S., Moonen, M.: Extension of the multi-channel Wiener filter with ICTD cues for noise reduction in binaural hearing aids. Applications of Signal Processing to Audio and Acoustics 16(16), 70–73 (2005)
- [8] Erkelens, J.S., Heusdens, R.: Tracking of nonstationary noise based on data-driven recursive noise power estimation. IEEE Trans. Audio, Speech and Language Processing 16(6), 1112–1123 (2008)
- [9] Alpayd, E.: Soft vector quantization and the EM algorithm. Neural Networks 11(3), 467–477 (1998)
- [10] Dhanalakshmi, P., Palanivel, S., Ramalingam, V.: Classification of audio signals using AANN and GMM. Applied Soft Computing 11(1), 716–723 (2011)
- [11] Soo, Y.J.: U-City platform. In: International Conference on U-City, August 27–28 (2009)
- [12] White, T.: Hadoop: The Definitive Guide, 1st edn. O'Reilly Media (2009)
- [13] Spink, A., Jansen, B.J., Blakely, C., Koshman, S.: A study of results overlap and uniqueness among major Web search engines. Information Processing and Management 42, 1379–1391 (2006)

Certificate-Based Encryption Scheme with General Access Structure

Tomasz Hyla and Jerzy Pejaś

West Pomeranian University of Technology in Szczecin
Faculty of Computer Science and Information Technology, Poland
{thyla,jpejas}@wi.zut.edu.pl

Abstract. The protection of sensitive information is very important, but also a difficult task. It usually requires a centralised access policy management and control system. However, such solution is often not acceptable in the era of users' mobility. In the paper we propose a certificate-based group-oriented encryption scheme with an effective secret sharing scheme based on general access structure. The special design of the scheme ensures that the shared secret (encryption key information), a collection of shareholders, and the access structure can be dynamically changed without the need to update the long-term keys and shares owned by shareholders. It is also possible to delegate the access rights to another member of the qualified subgroup or to a new entity from outside the current access structure.

Keywords: Information protection, general access structure, cryptographic access control, certificate-based cryptosystems.

1 Introduction

The user and information mobility is an important feature of IT systems, which have to be considered during the design of the mechanisms for protection of sensitive information. The mobility enables creation of many new and exciting applications, and makes life much easier for mobile workers as well. Mobile devices, i.e. laptops, tablets and smartphones, often contain sensitive information, e.g. personal data, address books, files with valuable information (e.g. contracts, orders, projects). The information is usually downloaded from the network, where it can be stored by third parties.

The ability to download information from the network is crucial for the working comfort of the mobile user: regardless of where the user is, the information is always at hand. On the other hand the information stored in network by third parties subject to the risks and vulnerabilities associated with information security, i.e. anonymity, information retrieval, loss, theft and interception.

One method to reduce some of these risks is to store the information in an encrypted form. However, such solution limits the users' ability to selectively share their encrypted information at a fine-grained level. We need to control the access to the information, but the access control mechanisms should allow granting access according to a number of different constraints depending on the user privileges.

The most effective solution of the user and information mobility problem can be achieved by using cryptographic access control mechanisms. These mechanisms allow to store the information in the network in an encrypted form and to be decrypted only by authorised users.

Cryptographic access control mechanisms are typically implemented in two stages. At the first stage the information is encrypted (according to some pre-defined access control policy) and is made available on a public server. At the second stage the encrypted information can be collected by any entity. However, the information can be read only by an entity that meets the requirements specified in the access policy related to the encrypted information. A group-oriented cryptosystem, where a group of participants cooperatively decrypt the ciphertext, is the solution to this kind of task.

1.1 Related Works

The concept of a group-oriented cryptosystem was first introduced by Y. Desmedt in [1] and is based on cooperation of designated authorized subsets of participants (an access structure). In group-oriented cryptography a sender firstly determines an access structure suitable to a receiving group of users and then sends an encrypted information or stores it in some localisation. Only authorized subsets of users in the group can cooperatively recover the message.

Many group-oriented decryption schemes are based on a traditional certificate-based PKI, on an identity-based public key cryptography (ID-PKC) or on a certificateless public key cryptography (CL-PKC). However, the need for PKI supporting certificates is considered as the main drawback for deployment and management of the traditional PKI. On other hand, the main disadvantages of ID-PKC and CL-PKC are the lack of authentication of TAs and end users. C. Gentry in [2] introduced the solution that comes naturally. This solution combines the merits of traditional public key infrastructure (PKI) and identity-based cryptography. Primarily it was used for encryption and was called certificate-based encryption, but it was quickly generalised for certificate-based signature schemes [3].

There are many works on ID-based threshold decryption scheme [4, 5] that combines ID-based cryptography with threshold decryption. Considerable less effort is devoted to ID-based group-oriented decryption scheme with general access structure [6-8]. This is due to the greater popularity of threshold secret sharing methods and their simplicity in the case of a large number of subgroups belonging to the access structure. However, to realise the selective access control to information, i.e. to solve the problem of the user and information mobility, the ID-based group-oriented decryption schemes with general access structure are more suitable.

1.2 Our Contributions

In this work we firstly contribute the definition, formalization and generic feasibility of group encryption. Next, we construct a new certificate and ID-based group-oriented decryption scheme with general access structure (CIBE-GAS), and investigate its related practical and theoretical properties. The CIBE-GAS scheme is

more suitable, comparing to threshold secret sharing methods, when the same access rights to decrypt data should be selectively assigned to all participants belonging to the same well defined group of users.

The proposed certificate-based encryption scheme with general access structure (CIBE-GAS, Section 3) combines three different ideas: the secret sharing scheme [9], publicly available evidence of being a member of a particular group [10] and Sakai-Kasahara IBE (SK-IBE) scheme [11] with technique introduced by Fujisaki and Okamoto [12]. Such approach allows to achieve the new group encryption scheme with following features:

- (a) the originator is not required to know the structure of qualified subsets, members of which are authorised to decrypt the information; he simply encrypts it, no designated group having in mind, and then decides who should be able to decrypt it (the value C_5 by Eq. (14) can be calculated at any time);
- (b) there is no need to designate a specific recipient of encrypted information - each member within a qualified subset can decrypt it (Section 3, Decryption algorithm); moreover, a sender can temporarily remove some subgroups from having access rights to encrypted information, i.e. a sender can arbitrarily select the recipients by overlaying the appropriate filter on the access structure;
- (c) the CIBE-GAS scheme is the certificate and ID-based encryption scheme (Section 3); it means, compared to the certificateless schemes, that partial key created by TA is published as a certificate and allows simplifying the user's identity verification.

Furthermore, the CIBE-GAS scheme has special construction of the public component $k_{i,j}$ (Section 3, Eq. (7)), which (a) protects the scheme against dishonest shareholders and unauthorised changes of the secret values being in possession of all users, and (b) allows any shareholder $u_i \in U$ to check if he is a member of an authorised group A_j . This component allows also any member of qualified subset to delegate his rights to any entity, which belongs or doesn't belong to the set of all users (Section 4).

We proved that the CIBE-GAS scheme is correct and secure against chosen-plaintext attacks IND-CID-GO-CPA (Section 5). The proposed encryption scheme was implemented and tested using a freely available PBC library written by Ben Lynn [13].

1.3 Paper Organisation

The paper is organized as follows. In Section 2, the bilinear maps and their properties are reviewed. Then, we present the *Discrete Logarithm* problem and its variations, on which our scheme is based. This Section introduces also some basic definitions of secret sharing schemes with general access structures, which works under certificate and ID-based scenarios. In Section 3 we present the group-oriented encryption scheme CIBE-GAS with general access structures. Section 4 presents an extension to the CIBE-GAS scheme allowing delegations to be specified from an authorized user to any another user. The analyses and discussions concerning the proposed scheme

are given in Section 5. Section 6 shows a practical implementation of the group-oriented decryption scheme and summarizes the results of tests. Finally, conclusions are presented.

2 Preliminaries

2.1 Bilinear Groups and Security Assumptions

Below, we summarise some concepts of bilinear pairings using notations similar to those presented by Al-Riyami, S., et al. [14].

Definition 1. Let $(G_1, +)$ and (G_2, \cdot) be two cyclic groups of some prime order $q > 2^k$ for security parameter $k \in \mathbb{N}$. The bilinear pairing is given as $\hat{e}: G_1 \times G_1 \rightarrow G_2$ and must satisfy the following three properties:

1. **Bilinearity:** $\hat{e}(aP, bQ) = \hat{e}(abP, Q) = \hat{e}(P, abQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_1$ and all $a, b \in \mathbb{Z}_q^*$; this can be restated in the following way: for $P, Q, R \in G_1$, $\hat{e}(P+Q, R) = \hat{e}(P, R)\hat{e}(Q, R)$ and $\hat{e}(P, Q+R) = \hat{e}(P, Q)\hat{e}(P, R)$.
2. **Non-degeneracy:** some $P, Q \in G_1$ exists such that $\hat{e}(P, Q) \neq I_{G_2}$; in other words, if P and Q are two generators of G_1 , then $\hat{e}(P, Q)$ is a generator of G_2 .
3. **Computability:** given $P, Q \in G_1$, an efficient algorithm computing $\hat{e}(P, Q)$ exists.

To construct the bilinear pairing we can use, for example, the Weil or Tate pairings associated with an elliptic curve over a finite field.

2.2 Security Assumptions and Hard Problems

A few definitions presented below are important for security reduction techniques used to design the CIBE-GAS scheme and prove its security. We start from three classical hard problems: the DL (Discrete Logarithm) and BDH (Bilinear Diffie-Hellman) problems. Then we describe a problem presented in [15], called the k -BDHI (Bilinear Diffie-Hellman Inversion) problem.

Assumption 1 (DL problem). For $P, Q \in G_1^*$ finding an integer k , which satisfies $Q = kP$ is hard.

Assumption 2 (BDH problem [16]). For $P, Q \in G_1^*$ and given (P, aQ, bQ, cQ) , where $a, b, c \in \mathbb{Z}_q^*$, computing $\hat{e}(P, Q)^{abc}$ is hard.

Assumption 3 (k -BDHI problem [15]). For an integer k , $x \in_R \mathbb{Z}_q^*$, $P, Q \in G_1^*$ and given $(P, xQ, x^2Q, \dots, x^kQ)$, computing $\hat{e}(P, Q)^{1/x}$ is hard.

Obviously, if it is possible to solve DL problem then it is also possible to solve BDH problem (given Q, aQ, bQ and cQ we can take discrete logarithms to obtain a, b

and b , which allows computing $\hat{e}(P, Q)^{abc}$. It is not known whether the k -BDHI problem, for $k > 1$, is equivalent to BDH [15]. However, solving the k -BDHI problem is no more difficult than calculating discrete logarithms in G_I .

2.3 General Access Structure

An access structure is a rule that defines how to share a secret, or more widely, who has an access to particular assets in IT system. Access structures can be classified into structures with and without threshold [17]. Although threshold access structures are frequently used (e.g. the most familiar examples are (n, n) and (t, n) secret sharing schemes given by Shamir or by Asmuth-Bloom), the non-threshold structures are more versatile. It is especially visible when the sender of the information defines special decryption rules that have to be met by the document recipient (e.g., the recipient should belong to a specific users' group).

Let us assume that $U = \{u_1, u_2, \dots, u_n\}$ is a set of n participants. The set $\Gamma = \{A \in 2^U : \text{a set of shareholders, which are designated to reconstruct the secret}\}$ is an access structure of U , if the secret can be reconstructed by any set $A \in \Gamma$. All sets in access structure Γ are called authorized or qualified subsets. A desirable feature of each access structure is its monotonicity. It means that every set containing a subset of privileged entities is also a collection of the privileged entities. The set of all minimal subsets $C \in \Gamma$ is called the access structure basis Γ_0 (or alternatively, the minimal access structure) and is expressed mathematically by the following relation:

$$\Gamma \supseteq \Gamma_0 = \{C \in \Gamma : \forall_{B \subset C} B \notin \Gamma\} \quad (1)$$

Due to the monotonicity of the set Γ , the access structure basis Γ_0 may be always extended to the set Γ by including all supersets generated from the sets of Γ_0 .

The access structure $\Gamma_{(t, n)}$ of the threshold scheme (t, n) is defined as follows:

$$\Gamma_{(t, n)} = \{A \in 2^U : |A| \geq t\} \quad (2)$$

It is easy to notice, that in case of the access structure $\Gamma_{(t, n)}$ of the threshold scheme (t, n) , all users have the same privileges and credentials. G.J. Simmons in [18] generalized a secret threshold sharing scheme (t, n) and gave the definition of hierarchical (multilevel) and compartmented threshold secret sharing. In such approach, in contrast to the classical threshold secret sharing, trust is not uniformly distributed among the members of the qualified subsets. It means that participants are divided into several subsets and only participants belonging to the same subset play the equivalent roles.

We say that the structure is useful, when it is possible to implement the access structure Γ . An example of the access structures realization is the approach proposed by Benaloh-Leichter [19]. However, the application of access structures for the

construction of group-oriented decryption scheme is effective only when it is possible to reuse shares being in possession of participants. The discussion how to meet this requirement is presented in the work [9, 10, 20, 21].

3 Full Certificate-Based Encryption Scheme with General Access Structure

Assume that there are given: n -element set containing all shareholders $U = \{u_1, u_2, \dots, u_n\}$, m -element access structure $\Gamma = \{A_1, A_2, \dots, A_m\}$, dealer $D \notin U$ and combiner $Com \in U$. Then proposed Certificate-Based Encryption scheme with General Access Structure (CIBE-GAS) consists of eight algorithms: **Setup**, **SetSecretValue**, **CertGen**, **SetPublicKey**, **ShareDistribution**, **Encryption**, **SubDecryption** and **Decryption**.

The **ShareDistribution** algorithm is based on ideas taken from [9, 10] and allows to generate shares and evidences used during a message decryption (the **SubDecryption** and **Decryption** algorithms). In turn, group **Encryption** and **Decryption** algorithms with general access structure are built on basis of the non-group SK-IBE scheme [11]. A detailed description of all algorithms of CIBE-GAS scheme is presented below.

Setup. For cyclic additive group $(G_1, +)$ and cyclic multiplicative group (G_2, \times) of the same prime order q a trusted authority TA chooses randomly its main key $s \in {}_R Z_q^*$, defines a bilinear pairing \hat{e} and generates encryption scheme parameters *params*:

$$\hat{e} : G_1 \times G_1 \rightarrow G_2 \quad (3)$$

$$params = \{G_1, G_2, \hat{e}, q, P, P_0, H_1, H_2, H_3, H_4, H_5, H_6\} \quad (4)$$

where P is a primitive element of G_1 , $P_0 = sP$ is a public key, $H_1 : \{0, 1\}^* \times G_1 \times G_1 \rightarrow G_1^*$, $H_2 : \{0, 1\}^* \times G_1 \times G_1 \rightarrow Z_q^*$, $H_3 : G_2 \times \{0, 1\}^* \rightarrow Z_q^*$, $H_4 : \{0, 1\}^p \times \{0, 1\}^p \rightarrow Z_q^*$, $H_5 : G_2 \rightarrow \{0, 1\}^p$ and $H_6 : \{0, 1\}^p \rightarrow \{0, 1\}^p$ are secure hash functions. Last two hash functions are used to transform a message M of p bits: a cryptographic hash function H_5 hashes elements of G_2 into a form that can be combined with the plaintext message M , which is a bit string of length p .

SetSecretValue. Every shareholder $u_i \in U$ with an identity ID_i chooses a random number $s_i \in {}_R Z_q^*$ ($i=1, \dots, n$), calculates $X_i = s_i P$, $Y_i = s_i P_0$ and sends them to TA. The dealer $D \notin U$ performs similar actions: chooses secret $s_d \in {}_R Z_q^*$, calculates $X_d = s_d P$ and $Y_d = s_d P_0$.

CertGen. TA checks equation $\hat{e}(X_i, P) = \hat{e}(Y_i, P_0)$ for every shareholder identity ID_i ($i=1, \dots, n$). If test results are positive, then TA calculates iteratively for $i=1, \dots, n$ hash values $Q_i = H_1(ID_i, Pk_i)$, where $Pk_i = (X_i, Y_i)$, and participant's certificate $Cert_i = sQ_i$. In similar way dealer's certificate $Cert_d = sQ_d$ is calculated, where $Q_d = H_1(ID_d, Pk_d)$ and $Pk_d = (X_d, Y_d)$. TA publishes all issued certificates.

SetPublicKey. Every shareholder with an identity ID_i tests authenticity of received certificate $Cert_i$ using equation $\hat{e}(Cert_i, P) = \hat{e}(Q_i, P_0)$. If the verification passes, then the shareholder $u_i \in U$ ($i=1, \dots, n$) publishes his or her public keys $Pk_i = (X_i, Y_i)$. The dealer proceeds similarly and publishes his or her public key $Pk_d = (X_d, Y_d)$.

ShareDistribution. The dealer $D \notin U$ tests public keys of all shareholders $u_i \in U$, verifying equations $\hat{e}(Cert_i, X_i) = \hat{e}(Q_i, Y_i)$ ($i=1, \dots, n$). If test results are positive, then the dealer:

(a) for $i=1, \dots, n$ calculates values

$$h'_i = \hat{e}(Cert_d + Cert_i, Y_i)^{s_d} = \hat{e}(Cert_d + Cert_i, Y_d)^{s_i} \quad (5)$$

$$h''_i = \hat{e}(Cert_i, Y_i)^{s_d} = \hat{e}(Cert_i, Y_d)^{s_i} \quad (6)$$

(b) chooses $m = |\Gamma|$ different values $d_j \in_R Z_q \setminus \{1\}$, ($i=1, \dots, m$); these values should unambiguously identify qualified subsets of an access structure $\Gamma = \{A_1, A_2, \dots, A_m\}$;

(c) chooses secret $y \in_R Z_q^*$ and two random numbers $\alpha, \beta \in_R Z_q^*$; keeps the number α secret and then constructs first-degree polynomial $f(x) = y + \alpha x$;

(d) calculates $f(1)$ and

$$\gamma_j = f(d_j) - \sum_{u_{i_j} \in A_j} H_3(h'_{i_j}, d_j \beta) \quad (7)$$

for each subset $A_j = \{u_{1_j}, u_{2_j}, \dots\} \in \Gamma$, $j=1, \dots, m$;

(e) for every shareholder $u_i \in A_j$ ($i=1, \dots, n$; $j=1, \dots, m$) calculates the evidence in the form:

$$k_{i,j} = \frac{(H_3(h'_i, d_j \beta) - y^{-1} H_3(h''_i, d_j \beta))}{s_d + H_2(ID_d, Pk_d)} X \quad (8)$$

(f) publishes β , $f(1)$, $Y = yP$, $Y_{-1} = y^{-1}P$, $(d_j, \gamma_j, k_{i,j})$ for $j=1, \dots, m$ and $i=1, \dots, n$; it should be noted that every shareholder $u_i \in U$ might verify whether his secret value s_i is related with parameters published by TA and the dealer:

$$\begin{aligned}
& \hat{e}(H_2(ID_d, Pk_d)P + X_d, s_i^{-1}k_{i,j}) = \\
& \hat{e}(P, H_3(\hat{e}(Cert_d + Cert_i, Y_d)^{s_i}, d_j\beta))P - \\
& - H_3(\hat{e}(Cert_i, Y_d)^{s_i}, d_j\beta)Y_{-1}
\end{aligned} \tag{9}$$

This verification can be repeated for each qualified group, in which a shareholder $u_i \in U$ is a member. Moreover, special construction of the evidence $k_{i,j}$ protects from dishonest shareholders, preventing from unauthorised changes of the secret value s_i as well as value of $k_{i,j}$.

Encryption. To encrypt the message $M \in \{0, 1\}^p$ the dealer D selects a random value $\sigma \in \{0, 1\}^p$ and:

- (a) calculates $r = H_4(\sigma, M)$;
- (b) sets the ciphertext $C = (C_1, C_2, C_3, C_4, C_5, C_6)$ as follows:

$$C_1 = r(H_2(ID_d, Pk_d)P + X_d) \tag{10}$$

$$C_2 = \sigma \oplus H_5(\hat{e}(P, Y)^r) \tag{11}$$

$$C_3 = M \oplus H_6(\sigma) \tag{12}$$

$$C_4 = \hat{e}(P, f(I)P)^r \tag{13}$$

$$C_5 = \{v_k = \hat{e}(P, \gamma_k P)^r, \forall k \in F \subseteq 2^m\} \tag{14}$$

$$C_6 = rY_{-1} \tag{15}$$

The set F in C_5 plays the role of the filter, which superimposed on the access structure Γ allows decrypting information only by privileged groups, which indexes belong to F .

SubDecryption. Every shareholder from the privileged subset $u_{ij} \in A_j \in \Gamma$ ($j \in F$) partially decrypts ciphertext $C = (C_1, C_2, C_3, C_4, C_5, C_6)$ using his share s_{ij} and returns to the combiner the following value:

$$\delta_{ij,j} = \hat{e}(C_1, s_{ij}^{-1}k_{ij,j})\hat{e}(P, H_3(\hat{e}(Cert_{ij}, Y_d)^{s_{ij}}, d_j\beta))C_6) \tag{16}$$

Decryption. Let us assume further that one of privileged shareholders, e.g. $u_{kj} \in A_j, k \in \{1, \dots, |A_j|\}$, will play the combiner role. To decrypt the ciphertext $C = (C_1, C_2, C_3, C_4, C_5, C_6)$, the combiner $Com = u_{kj} \in A_j$ from (belonging to?) any authorised group performs the following steps:

- (a) gathers all partial values $\delta_{I,j}, \dots, \delta_{Com-l,j}, \delta_{Com,j}, \delta_{Com+l,j}, \dots, \delta_{|A_j|_j,j}$ and calculates

$$\Delta = \Delta_{1,d_j-l}^{\frac{d_j}{d_j-l}} \cdot \Delta_{2,d_j-l}^{\frac{-l}{d_j-l}} \quad (17)$$

where $v_j \in C_5$ and

$$\begin{aligned} \Delta_1 &= C_4 \\ \Delta_2 &= v_j \cdot \delta_{Com,j} \prod_{u_{ij} \in A_j \setminus Com} \delta_{i,j} \end{aligned} \quad (18)$$

- (b) calculates

$$\sigma = C_2 \oplus H_5(\Delta) \quad (19)$$

- (c) calculates

$$M = C_3 \oplus H_6(\sigma) \quad (20)$$

- (d) recovers $r = H_4(\sigma, M)$;

- (e) if $C_1 \neq r(H_2(ID_d, Pk_d)P + X_d)$, then raises an error condition and exits; otherwise sets the plaintext to M .

Thus the plaintext M can be obtained from the ciphertext $C = (C_1, C_2, C_3, C_4, C_5, C_6)$ and the combiner can decide if the decrypted ciphertext is correct.

4 Rights Delegation

The CIBE-GAS scheme allows to decide who can have access to the information (i.e. allows to describe each member of the access structure, who is able to gather enough number of partial values $\delta_{I,j}, \dots, \delta_{|A_j|_j,j}$ ($j=1, \dots, m$)). Moreover, we can easily

introduce delegation operation to the proposed scheme.

Assume that delegation rule is implemented as follows: (a) the user $u_i \in U$ requests the dealer to delegate his right to any entity u_p (belonging or not belonging to set U) to be a member of a group A_j ; the entity u_p cannot forward this right further, (b) dealer issues to the entity u_p evidence $k_{p,j}$ and publish it. The evidence $k_{p,j}$ has the following form:

$$k_{p,j} = \frac{(H_3(h'_i, d_j \beta) - y^{-l} H_3(h''_p, d_j \beta))}{s_d + H_2(ID_d, Pk_d)} X_p \quad (21)$$

where $h_p'' = \hat{e}(Cert_p, Y_p)^{f_d} = \hat{e}(Cert_p, Y_d)^{f_p}$, and $Cert_p$ is the certificate of entity u_p .

The entity u_p can calculate his partial share using owned by himself evidence $k_{p,j}$ and the ciphertext $C = (C_1, C_2, C_3, C_4, C_5, C_6)$:

$$\begin{aligned} \delta_{p,j} &= \hat{e}(C_1, s_p^{-1} k_{p,j}) \hat{e}(P, H_3(\hat{e}(Cert_p, Y_d)^{f_p}, d_j \beta) C_5) = \\ &\hat{e}(rP, H_3(\hat{e}(Cert_d + Cert_i, Y_d)^{f_i}, d_j \beta) P) = \delta_{i,j} \end{aligned} \quad (22)$$

which is equal to the share $\delta_{i,j}$ of entity $u_i \in U$ (compare the proof of Theorem 1). It follows, that the entity u_p indeed represents the entity $u_i \in U$. Hence, the entity can not only be a provider (on behalf of the entity $u_i \in U$) of the share $\delta_{i,j}$, but might play a combiner role and decipher encrypted message.

5 Analysis and Discussion

5.1 Correctness

Assume that the ciphertext $C = (C_1, C_2, C_3, C_4, C_5, C_6)$ was generated using the **Encryption** algorithm. Then, according to properties of bilinear parings, the following theorem shows that the users in some access instance A_j can cooperate to recover the message M .

Theorem 1. Any user $u_{k_j} = Com \in A_j, k \in \{1, \dots, |A_j|\}$ (i.e. the combiner), which is the member of authorised subgroup A_j referenced by index j belonging to the set F (see Eq. (14)) can decrypt the message M encrypted in the equation (12).

Proof. From Eq. (16), for each $u_{i_j} \in A_j, i \in \{1, \dots, |A_j|\}$ we have:

$$\begin{aligned} \delta_{i_j,j} &= \hat{e}(C_1, s_{i_j}^{-1} k_{i_j,j}) \hat{e}(P, H_3(\hat{e}(Cert_{i_j}, Y_d)^{f_{i_j}}, d_j \beta) C_6) = \\ &= \hat{e} \left(r(H_2(ID_d, Pk_d) P + X_d), s_{i_j}^{-1} \frac{(H_3(h'_{i_j}, d_j \beta) - y^{-1} H_3(h''_{i_j}, d_j \beta))}{s_d + H_2(ID_d, Pk_d)} X_i \right) \cdot \\ &\quad \cdot \hat{e}(P, H_3(h''_{i_j}, d_j \beta) C_6), \quad \text{by Eqs.(8) and (10)} \\ &= \hat{e}(rP, (H_3(h'_{i_j}, d_j \beta) - y^{-1} H_3(h''_{i_j}, d_j \beta)) P) \hat{e}(P, H_3(h''_{i_j}, d_j \beta) r y^{-1} P), \text{ by Eq.(15)} \\ &= \hat{e}(rP, H_3(\hat{e}(Cert_d + Cert_{i_j}, Y_d)^{f_{i_j}}, d_j \beta) P) \quad \text{by Eq. (5).} \end{aligned}$$

With this partially decrypted ciphertext $\delta_{i_j,j}, i \in \{1, \dots, |A_j|\}$ from all participants of authorised subset A_j the combiner $Com \in A_j, k \in \{1, \dots, |A_j|\}$ can get:

$$\Delta_2 = v_j \cdot \delta_{Com,j} \prod_{u_{i_j} \in A_j \setminus \{Com\}} \delta_{i_j,j},$$

$$\begin{aligned}
&= \hat{e}(P, \gamma_j P)^r \cdot \prod_{u_{ij} \in A_j} \hat{e}(rP, H_3(h'_{u_{ij}}, d_j \beta)P) \quad \text{by Eq. (14) and calculated } \delta_{ij,j} \\
&= \hat{e}\left(rP, \left(\gamma_j + \prod_{u_{ij} \in A_j} H_3(h'_{u_{ij}}, d_j \beta)\right)P\right) = \hat{e}(P, rf(d_j)P), \quad \text{by Eq. (7)}
\end{aligned}$$

The obtained value of $\Delta_j = C_4$ and the reconstructed value Δ_2 are related with two points on line $f(x) = y + \alpha x$, and allow to make an implicit interpolation (using Lagrange's polynomial interpolation) of the secret y :

$$\begin{aligned}
\Delta &= \Delta_1 \frac{d_j}{d_j - l} \cdot \Delta_2 \frac{-l}{d_j - l} = \\
&= \hat{e}(P, rP) \frac{d_j}{d_j - l} f(l) + \frac{-l}{d_j - l} f(d_j) = \hat{e}(P, rP)^y = \hat{e}(P, Y)
\end{aligned} \tag{23}$$

Thus the plaintext M can be obtained from Eqs (11) and (12) as follows:

$$\sigma' = C_2 \oplus H_5(\Delta) = \sigma \oplus H_5(\hat{e}(P, Y)^r) \oplus H_5(\Delta), \quad \text{by Eq. (11)}$$

$$= \sigma \oplus H_5(\Delta) \oplus H_5(\Delta) = \sigma, \quad \text{by Eq. (23)}$$

$$M' = C_3 \oplus H_6(\sigma'),$$

$$= M_3 \oplus H_6(\sigma) \oplus H_6(\sigma') = M \quad \text{by Eq. (12).}$$

If $C_1 \neq r(H_2(ID_d, Pk_d)P + X_d)$, the message M' calculated by (20) is the message M .

This ends the proof. \square

5.2 Security Analysis

The CIBE-GAS is the secret sharing group-oriented decryption scheme based on Sakai and Kasahara non-group IBE (SK-IBE) scheme [11]. L. Chen and Z. Cheng prove in [16] that the SK-IBE scheme is secure against chosen-plaintext attacks (IND-ID-CCA) in the random oracle model. They prove also that the security of SK-IBE can be reduced to the hardness of the k -BDHI problem.

In CIBE-GAS scheme an adversary can obtain public information related to all participants, i.e. $k_{i,j}$, $Pk_i = (X_i, Y_i)$, $Cert_i$ ($i=1, \dots, n$; $j=1, \dots, m$). In notice board service are also available other parameters like β , $f(1)$, $Y = yP$, $Y_{-1} = y^{-1}P$ and (d_j, γ_j) for each group of participants ($j=1, \dots, m$). However, this information doesn't affect the security of the scheme, while the hardness of the CIBE-GAS scheme is reduced from the k -BDHI problem to the DL problem.

Theorem 2. The proposed CIBE-GAS scheme is secure against chosen-plaintext attacks IND-CID-GO-CPA in the standard model, assuming that (1) the hash function H_3 is collision-resistant and (2) the DL assumption holds in group G_1 .

Proof (sketch). A group-oriented certificate and ID-based cryptosystem is secure against chosen-plaintext attacks IND-CID-GO-CPA if no polynomially bounded adversary has a non-negligible advantage against the cryptosystem in the game like this defined in [12]. In this game “an adversary makes an attack on an authorised

subset A_j , which the member number is $|A_j|$. We allow the adversary to possess the most advantageous conditions that he could obtain the private keys of any $|A_j|-1$ participants in the authorised subset, and furthermore, he could also obtain the private keys of any number of participants other than those in the authorised subset A_j ” [8].

Assume that the participant $u_{k_j} \in A_j$ and his secret key s_{k_j} are the adversary’s targets of attacks. Then for the key \bar{s}_{k_j} chosen by the adversary and from (18) follows:

$$\begin{aligned} \Delta_2 = & v_j \cdot \delta_{u_{k_j}, j} \prod_{u_{i_j} \in A_j \setminus \{u_{k_j}\}} \delta_{i_j, j} = \hat{e} \left(rP, \left(\gamma_j + \sum_{u_{i_j} \in A_j \setminus \{u_{k_j}\}} H_3(h'_{u_{i_j}}, d_j \beta) \right) P \right) \cdot \\ & \cdot \hat{e} \left(rP, \bar{s}_{k_j}^{-1} s_{k_j} H_3(h'_{u_{i_j}}, d_j \beta) P + \right. \\ & \left. y^{-1} \left(H_3(\hat{e}(Cert_{k_j}, \bar{s}_{k_j} Y_d), d_j \beta) - \bar{s}_{k_j}^{-1} s_{k_j} H_3(\hat{e}(Cert_{k_j}, s_{k_j} Y_d), d_j \beta) \right) P \right) \end{aligned}$$

The value of Δ_2 will be valid only if $\bar{s}_{k_j}^{-1} s_{k_j} = 1 \in Z_q^*$. This means that the adversary will succeed in solving the DL problem for $P, X_{k_j} \in G_1^*$. Hence, if the CIBE-GAS scheme is not secure against an IND-CID-GO-CPA adversary, then the corresponding DL assumption is flawed.

6 Implementation and Practical Performance

The scheme was implemented using the PBC library created and maintained by Benn Lynn [13]. The library consists of API, which is abstract enough, so only basic understanding of bilinear pairings is required from a programmer. The test operating system was Ubuntu 11.04 running on a virtual machine on Windows 7 host system. The host system machine was Intel Xeon W3520@2,67 GHz with 4GBRAM. We have run several tests that confirmed that our scheme is correct (the decrypted message was equal to the original message). In our tests we have used “Type A” pairing which are constructed on the curve $y^2 = x^3 + x$ over the field F_q for a 512-bit prime $q \equiv 3 \pmod{4}$, and thus the pairing result are in F_{q^2} (see details on PBC library specification [10]).

The second purpose of the tests was to verify algorithms’ performance. The primary code analysis has shown that the most time consuming are encryption and decryption algorithms. The encryption time depends on the number of shareholders and on the cardinality $\#A$ of the qualified subset A . The encryption time of a sample case ($\Gamma = \{A_1, \dots, A_{l_d}\}$ with six shareholders and average cardinality $\#A$ of the A equals three) is around 168ms.

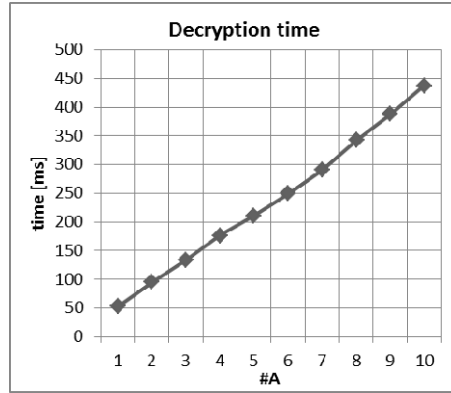


Fig. 1. Decryption time

The decryption time depends on the cardinality $\#A$ of the set A . The tests results for decryption for the different cardinality $\#A$ of the qualified subsets A are presented in the Fig.1. The results show that decryption time increases linearly with increasing the cardinality $\#A$ of the set A . It is also noteworthy that the most time consuming operation is paring calculation. The time for pairing calculation without pre-processing is around 4ms.

7 Summary

The proposed encryption scheme CIBE-GAS provides sensitive information protection in accordance with any general access structure. In this scheme we use the idea of a secret sharing scheme with general access structure given by Sang, Y., et al. [9] and the idea of a dynamic encryption scheme presented by Long, Y, et al. [10] (with modifications by Kitae, K., et al [21]).

We proved that our group-oriented certificate and ID-based cryptosystem is secure against chosen-plaintext attacks IND-CID-GO-CPA. Furthermore, the scheme allows sending the message to any authorised subsets with prior knowledge of their structure. This fine-grained access control mechanism allows a dealer alone (without the cooperation with other members of the subgroup) or each member of an authorised subgroup (in cooperation with other members of this subgroup) to decrypt sensitive information.

The access control to encrypted sensitive information is under the dealer's sole control (the dealer plays the role of an originator, which has the power to determine who is able to access the information). Especially, this means that the dealer can define different subsets of the access structure, can add members to authorised subgroups or remove members from these subgroups, and may delegate the access rights to another member of the subgroup or to a new entity not belonging to the current access structure. These properties make the scheme a suitable choice for practical applications and make it more flexible and well applied in the field of sensitive information security.

Finally, we provided an implementation of our system using the Pairing Based Cryptography (PBC) library. The test results obtained for different authorized subgroups are promising and show that our system performs well in practice.

Acknowledgment. This scientific research work is supported by NCBiR of Poland (grant No O N206 001340) in 2011-2012.

References

1. Desmedt, Y.: Society and Group Oriented Cryptography: A New Concept. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 120–127. Springer, Heidelberg (1988)
2. Gentry, C.: Certificate-based Encryption and the Certificate Revocation Problem. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 272–293. Springer, Heidelberg (2003)
3. Kang, B.G., Park, J.H., Hahn, S.G.: A Certificate-Based Signature Scheme. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 99–111. Springer, Heidelberg (2004)
4. Baek, J., Zheng, Y.: Identity-Based Threshold Decryption. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 262–276. Springer, Heidelberg (2004)
5. Long, Y., Chen, K., Liu, S.: ID-based threshold decryption secure against adaptive chosen-ciphertext attack. *Computers and Electrical Engineering* 33(3), 166–176 (2007)
6. Chang, T.-Y.: An ID-based group-oriented decryption scheme secure against adaptive chosen-ciphertext attacks. *Computer Communications* 32(17), 1829–1836 (2009)
7. Liu, H., Xie, W., Yu, J., Zhang, P., Liu, S.: A general threshold encryption scheme based on new secret sharing measure. In: 6th IEEE Conference on Industrial Electronics and Applications (ICIEA), June 21–23, pp. 2235–2239 (2011)
8. Xu, C., Zhou, J., Xiao, G.: General Group Oriented ID-Based Cryptosystems with Chosen Plaintext Security. *International Journal of Network Security* 6(1), 1–5 (2008)
9. Sang, Y., Zeng, J., Li, Z., You, L.: A Secret Sharing Scheme with General Access Structures and its Applications. *International Journal of Advancements in Computing Technology* 3(4), 121–128 (2011)
10. Long, Y., Chen, K.-F.: Construction of Dynamic Threshold Decryption Scheme from Pairing. *International Journal of Network Security* 2(2), 111–113 (2006)
11. Sakai, R., Kasahara, M.: ID based cryptosystems with pairing on elliptic curve. *Cryptology ePrint Archive*, Report 2003/054
12. Fujisaki, E., Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999)
13. Lynn, B.: PBC Library Specification, <http://crypto.stanford.edu/pbc/> (retrieved 2012)
14. Al-Riyami, S.S., Paterson, K.G.: Certificateless Public Key Cryptography. In: Lai, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003)
15. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
16. Chen, L., Cheng, Z.: Security Proof of Sakai-Kasahara's Identity-Based Encryption Scheme. In: Smart, N.P. (ed.) *Cryptography and Coding* 2005. LNCS, vol. 3796, pp. 442–459. Springer, Heidelberg (2005)

17. Daza, V., Herranz, J., Morillo, P., Ràfols, C.: Extensions of access structures and their cryptographic applications. *Applicable Algebra in Engineering, Communication and Computing* 21(4), 257–284 (2010)
18. Simmons, G.J.: How to (Really) Share a Secret. In: Goldwasser, S. (ed.) *CRYPTO 1988*. LNCS, vol. 403, pp. 390–448. Springer, Heidelberg (1990)
19. Benaloh, J., Leichter, J.: Generalized Secret Sharing and Monotone Functions. In: Goldwasser, S. (ed.) *CRYPTO 1988*. LNCS, vol. 403, pp. 27–35. Springer, Heidelberg (1990)
20. Zheng, Y., Hardjono, T., Seberry, J.: Reusing shares in secret sharing schemes. *Computer Journal* 37(3), 199–205 (1994)
21. Kitae, K., Lim, S., Yie, I., Kim, K.: Cryptanalysis of a Dynamic Threshold Decryption Scheme. *Communications of the Korean Mathematical Society* 24(1), 153–159 (2009)

Security Margin Evaluation of SHA-3 Contest Finalists through SAT-Based Attacks

Ekawat Homsirikamol², Paweł Morawiecki¹,
Marcin Rogawski², and Marian Srebrny^{1,3}

¹ Section of Informatics, University of Commerce, Kielce, Poland

² Cryptographic Engineering Research Group, George Mason University, USA

³ Institute of Computer Science, Polish Academy of Sciences, Poland

Abstract. In 2007, the U.S. National Institute of Standards and Technology (NIST) announced a public contest aiming at the selection of a new standard for a cryptographic hash function. In this paper, the security margin of five SHA-3 finalists is evaluated with an assumption that attacks launched on finalists should be practically verified. A method of attacks is called logical cryptanalysis where the original task is expressed as a SATisfiability problem. To simplify the most arduous stages of this type of cryptanalysis and helps to mount the attacks in a uniform way a new toolkit is used. In the context of SAT-based attacks, it has been shown that all the finalists have substantially bigger security margin than the current standards SHA-256 and SHA-1.

Keywords: Cryptographic hash algorithm, SHA-3 competition, algebraic cryptanalysis, logical cryptanalysis, SATisfiability solvers.

1 Introduction

In 2007, the U.S. National Institute of Standards and Technology (NIST) announced a public contest aiming at the selection of a new standard for a cryptographic hash function. The main motivation behind starting the contest has been the security flaws identified in SHA-1 standard in 2005. Similarities between SHA-1 and the most recent standard SHA-2 are worrisome and NIST decided that a new, stronger hash function is needed. 51 functions were accepted to the first round of the contest and in July 2009 among those functions 14 were selected to the second round. At the end of 2010 five finalists were announced: BLAKE [15], Groestl [21], JH [28], Keccak [13], and Skein [2]. The winning algorithm will be named ‘SHA-3’ and most likely will be selected in the second half of 2012.

Security, performance in software, performance in hardware and flexibility are the four primary criteria normally used in evaluation of candidates. Out of these four criteria, security is the most important criterion, yet it is also the most difficult to evaluate and quantify. There are two primary ways of estimating the security margin of a given cryptosystem. The first one is to compare the complexities of the best attack on the full cryptosystem. The problem with this

approach is that for many modern designs there is no known successful attack on the full cryptosystem. Security margin would be the same for all algorithms where there is nothing better than the exhaustive search if this approach is used. This is not different for SHA-3 contest where no known attacks on the full functions, except JH, have been reported. For JH there is a preimage attack [19] but its time complexity is nearly equal to the exhaustive search and memory accesses are over the exhaustive search bound. Therefore estimating the security margin using this approach tells us very little or nothing about differences between the candidates in terms of their security level. The second approach of how to measure the security margin is to compare the number of broken rounds to the total number of rounds in a given cryptosystem. As a vast majority of modern ciphers and hash functions (in particular, the SHA-3 contest finalists) has an iterative design, this approach can be applied naturally. However, there is also a problem with comparing security levels calculated this way. For example, there is an attack on 7-round Keccak-512 with complexity 2^{507} [3] and there is an attack on 3-round Groestl-512 with complexity 2^{192} [23]. The first attack reaches relatively more rounds (29%) but with higher complexity whereas the second attack has lower complexity but breaks fewer rounds (21%). Both attacks are completely non-practical. It is very unclear how such results help to judge which function is more secure.

In this paper we follow the second approach of measuring the security margin but with an additional restriction. We assume that the attacks must have practical complexities, i.e., can be practically verified. It is very similar to the line of research recently presented in [56]. This restriction puts the attacks in more ‘real life’ scenarios which is especially important for SHA-3 standard. So far a large amount of cryptanalysis has been conducted on the finalists, however the majority of papers focuses on maximizing the number of broken rounds which leads to extremely high data and time complexity. These theoretical attacks have great importance but the lack of practical approach is evident. We hope that our work helps to fill this gap to some extent.

The method of our analysis is a SAT-based attack. SAT was the first known NP-complete problem, as proved by Stephen Cook in 1971 [7]. A SAT solver decides whether a given propositional (boolean) formula has a satisfying valuation. Finding a satisfying valuation is infeasible in general, but many SAT instances can be solved surprisingly efficiently. There are many competing algorithms for it and many implementations, most of them have been developed over the last two decades as highly optimized versions of the DPLL procedure [10] and [11].

SAT solvers can be used to solve instances typically described in the Conjunctive Normal Form (CNF) into which any decision problem can be translated. Modern SAT solvers use highly tuned algorithms and data structures to find a solution to a given problem coded in this very simple form. To solve your problem: (1) translate the problem to SAT (in such a way that a satisfying valuation represents a solution to the problem); (2) run your favorite SAT solver to find a solution. The first connection between SAT and crypto dates back to [8], where a suggestion appeared to use cryptoformulae as hard benchmarks for propositional

satisfiability checkers. The first application of SAT solvers in cryptanalysis was due to Massacci et al. [18] called logical cryptanalysis. They ran a SAT solver on DES key search, and then also for faking an RSA signature for a given message by finding the e -th roots of a (digitalized) message m modulo n , in [12]. Courtois and Pieprzyk [9] presented an approach to code in SAT their algebraic cryptanalysis with some gigantic systems of low degree equations designed as potential procedures for breaking some ciphers. Soos et al. [26] proposed enhancing a SAT solver with some special-purpose algebraic procedures, such as Gaussian elimination. Mironov and Zhang [20] showed an application of a SAT solver supporting a non-automatic part of the attack [27] on SHA-1.

In this work we use SAT-based attacks to evaluate security margin of the 256-bit variant SHA-3 contest finalists and also compare them to the current standards, in particular SHA-256. We show that all five finalists have a big security margin against these kind of attacks and are substantially more secure than SHA-1 and SHA-256. We also report some interesting results on particular functions or its building blocks. Preimage and collision attacks were successfully mounted against 2-round Keccak. At the time of publication, this is the best known practical preimage attack on reduced Keccak. A pseudo-collisions on 6-round Skein-512-256 was also found. For the comparison, the Skein's authors reached 8 rounds but they found only pseudo-near-collision [2]. In the attacks we use our toolkit which is a combination of the existing tools and some newly developed parts. The toolkit helps in mounting the attacks in a uniform way and it can be easily used for cryptanalysis not only of hash functions but also of other cryptographic primitives such as block or stream ciphers.

2 Methodology of our SAT-Based Attacks

2.1 A Toolkit for CNF Formula Generation

One of the key steps in attacking cryptographic primitives with SAT solvers is CNF (conjunctive normal form) formula generation. A CNF is a conjunction of clauses, i.e., of disjunctions of literals, where a literal is a boolean valued variable or its negation. Thus, a formula is presented to a SAT solver as one big 'AND' of 'ORs'. A cryptographic primitive (or a segment of it) which is the target of a SAT based attack has to be completely described by such a formula. Generating it is a non-trivial task and usually very laborious. There are many ways to obtain a final CNF and the output results differ in the number of clauses, the average size of clauses and the number of literals. Recently we have developed a new toolkit which greatly simplifies the generation of CNF.

Usually a cryptanalyst needs to put a considerable effort into creating a final CNF. It involves writing a separate program dedicated only to the cryptographic primitive under consideration. To make it efficient, some minimizing algorithms (Karnaugh maps, Quine-McCluskey algorithm or Espresso algorithm) have to be used [17]. These have to be implemented in the program, or the intermediate results are sent to an external tool (e.g., Espresso minimizer) and then the minimized form is sent back to the main program. Implementing all of these

procedures requires a good deal of programming skills, some knowledge of logic synthesis algorithms and careful insight into the details of the primitive's operation. As a result, obtaining CNF might become the most tedious and error-prone part of any attack. It could be especially discouraging for researchers who start their work from scratch and do not want to spend too much time on writing thousands lines of code.

To avoid those disadvantages we have recently proposed a new toolkit consisting basically of two applications. The first of them is Quartus II — a software tool released by Altera for analysis and synthesis of HDL (Hardware Description Language) designs, which enables the developers to compile their designs and configure the target devices (usually FPGAs). We use a free-of-charge version Quartus II Web Edition which provides all the features that we need. The second application, written by us, converts boolean equations (generated by Quartus) to CNF encoded in DIMACS format (standard format for today's SAT solvers). The complete process of CNF generation includes the following steps:

1. Code the target cryptographic primitive in HDL;
2. Compile and synthesize the code in Quartus;
3. Generate boolean equations using Quartus inbuilt tool;
4. Convert generated equations to CNF by our converter.

Steps 2, 3, and 4 are done automatically. Using this method the only effort a researcher has to put is to write a code in HDL. Normally programming and 'thinking' in HDL is a bit different from typical high-level languages like Java or C. However it is not the case here. For our needs, programming in HDL looks exactly the same as it would be done in high-level languages. There is no need to care about typical HDL specific issues like proper expressing of concurrency or clocking. It is because we are not going to implement anything in a FPGA device. All we need is to obtain a system of boolean equations which completely describes the primitive we wish to attack. Once the boolean equations are generated by the Quartus inbuilt tool, the equations are converted into CNF by the separate application. The conversion implemented in our application is based on the boolean laws (commutativity, associativity, distributivity, identity, De Morgan's laws) and there are no complex algorithms involved.

It must be noted that Quartus programming environment gives us two important features which may help to create a possibly compact CNF. It minimizes the functions up to 6 variables using Karnaugh maps. Additionally, all final equations have at most 5 variables (4 inputs, 1 output). It is because Quartus is dedicated to FPGA devices which are built out of 'logic cells', each with 4 inputs/1 output. (There are also FPGAs with different parameters; e.g., 5/2. But we chose 4/1 architecture in all the experiments.) This feature is helpful when dealing with linear ANF equations with many variables (also referred as 'long XOR equations'). A simple conversion of such an equation to CNF gives an exponential number of clauses; an equation in n -variables corresponds to 2^{n-1} clauses in CNF. A common way of dealing with this problem is to introduce new variables and cut the original equation into a few shorter ones.

Example 1. Let us consider an equation with 5 variables:

$$a + b + c + d + e = 0$$

A CNF corresponding to this equation consists of 2^{5-1} clauses with 5 literals in each clause. However, introducing two new variables, we can rewrite it as a system of three equations:

$$a + b + x = 0$$

$$c + d + y = 0$$

$$e + x + y = 0$$

A CNF corresponding to this system of equations would consist of $2^2 + 2^2 + 2^2 = 12$ clauses.

Quartus automatically introduces new variables and cuts long equations to satisfy the requirements for FPGA architecture. Consequently a researcher needs not be worried that the CNF would be much affected by very long XOR equations (which may be a part of the original cryptographic primitive's description).

To the best of our knowledge, there are only two other tools which provide similar functionality to our toolkit — automate the CNF generation and help to mount the uniform SAT-based attacks. First is the solution proposed in [16] where the main idea is to change the behaviour of all the arithmetic and logical operators that the algorithm uses, in such a way that each operator produces a propositional formula corresponding to the operation performed. It is obtained by using C++ implementation and a feature of the C++ language called operator overloading. Authors tested their method on MD4 and MD5 functions. The proposed method can be applied to other crypto primitives but it is not clear how it would deal with more complex operations, e.g. an S-box described as a look-up table. The second tool is called Grain of Salt [25] and it incorporates some algorithms to optimize a generated CNF. However it can be only used with a family of stream ciphers.

In comparison to these two tools, our proposal is the most flexible. It can be used with many different cryptographic primitives (hash functions, block and stream ciphers) and it does not limit an input description to only simple boolean operations. The toolkit handles XOR equations efficiently and also takes an advantage of logic synthesis algorithms which help to provide more compact CNF.

2.2 Our SAT-Based Attack

All the attacks reported in the paper have a very similar form and consist of the following steps.

1. Generate the CNF formula by our toolkit;
2. Fix the hash and padding bits in the formula;
3. Run a SAT solver on the generated CNF.

The above scheme is used to mount a preimage attack, i.e., for a given hash value h , we try to find a message m such that $h = f(m)$. CryptoMiniSAT2, gold medalist from recent SAT competitions [24], is selected as our SAT solver. In the preliminary experiments, we also tried other state-of-art SAT solvers (Lingeling [4], Glucose [1]) but overall CryptoMiniSAT2 solves our formulas faster.

We attack functions with 256-bit hash. When constructing a CNF coding a hash function, one has to decide the size of the message (how many message blocks are taken as an input to the function). It is easier for a SAT solver to tackle with a single message block because coding each next message block would make a formula twice as big. However, each of the five finalists has a different way of padding the message. If only one message block is allowed, BLAKE-256 can take maximally 446 bits of message which are padded to get a 512-bit block. On the other hand, Keccak-256 can take as many as 1086 bits of message in a single block. To avoid the situation where one formula has much more message bits to search for by a SAT solver than the other formula, message is fixed to 446 bits (maximum value for BLAKE-256 with one message block processed, other finalists allow more).

To find a second preimage or a collision, only a small adjustment to the aforementioned attack is required. Once the preimage is found, we run SAT solver on exactly the same formula but with one message bit fixed to the opposite value of that from the preimage (rest of the message bits are left unknown). It turns out that in a very similar time the SAT solver is able to solve such slightly modified formula, providing a second preimage and a collision. The second preimages/collisions are expected because with a size of the message fixed to 446 bits we have 446 to 256 bits mapping.

3 Results

We have conducted the preimage attack described in Section 2.2 on the five finalists and also on the two standards SHA-256 and SHA-1. As a SAT solver we used CryptoMiniSat2, 2.9.0 version, with the parameters *gaussuntil=0* and *restart=static*. These settings were suggested by the author of CryptoMiniSat2. The experiments were carried out on Intel Core i7 2.67 GHz with 8 Gb RAM. Starting with 1-round variants of the functions, the SAT solver was run to solve the given formula and gave us the preimage. The time limit for each experiment was set to 30 hours. If the solution was found, we added one more round, encoded in CNF and gave it to the solver. The attack stopped when the time limit was exceeded or memory ran out. Table 1 shows the results. The second column contains the number of broken rounds in our preimage attack and the third column shows the security margin calculated as a quotient of the number of broken rounds and the total number of rounds. For clarity, we are reporting our preimage attack but, as explained above, it can be easily modified to get a second preimage or a collision. Therefore the numbers from Table 1 remain valid for all three types of attacks.

All the SHA-3 contest finalists have substantially bigger security margin than SHA-256 and SHA-1 standards. On the other hand, the finalists differ slightly (maximally 7%) and all have the security margin over 90%. For Groestl we were not able to attack even a 1-round variant, nor a simplified Groestl with the output transformation replaced by a simple truncation. The only successful attack on Groestl (or rather part of it) is the attack on the output transformation in the 1-round variant of Groestl. The output transformation is not a complete round but giving 100% of security margin would not be fair neither. Therefore we try to estimate ‘a weight’ of the output transformation. Essentially all the operations (equations) in Groestl compression function come from two very similar permutations (P and Q). The output transformation is built on the P permutation only so it can be treated as a half-operation of the compression function. Hence the attack on the output transformation in a 1-round variant of Groestl is shown in Table 1 as half the round.

All the reported attacks on the finalists took just a few seconds. Only for 16-round SHA-256 the attack lasted longer — one hour. Despite the fact a conservative time limit (30 hours) was set for this type of experiments, it did not help to extend the attack to reach one more round. It seems that the time of the attack grows superexponentially in the number of rounds. The same behaviour was observed by Rivest et al. when they tested MD6 function with their SAT-based analysis [22]. For MD6 with 256-bit hash size, they reached 10 rounds which gives 90% of security margin. For a reader interested in estimating the asymptotic complexity of our attacks, we report that it would be very difficult mainly because Altera does not reveal details of algorithms used in Quartus.

Table 1. Security margin comparison calculated from the results of our preimage attacks on round-reduced hash functions

Function	No. of rounds	Security margin
SHA-1	21	74% (21/80)
SHA-256	16	75% (16/64)
Keccak-256	2	92% (2/24)
BLAKE-256	1	93% (1/14)
Groestl-256	0.5*	95% (0,5/10)
JH-256	2	96% (2/42)
Skein-512-256	1	99% (1/72)

* Only output transformation broken. It is estimated as an equivalent to one half-operation of the Groestl compression function.

It is interesting to see if the parameters of CNF formula, that is the number of variables and clauses, could be a good metric for measuring the hardness of the formula and consequently the security margin. Table 2 shows the numbers of variables and clauses for full hash functions. The values are rounded to the nearest thousand. For SHA-1, SHA-256, BLAKE, and Skein, we have generated the complete formula with our toolkit. For the other functions, we have extrapolated the numbers from round-reduced variants as the toolkit had some memory

problems with those huge instances (over 1 million of clauses). As every round in the given function is basically the same (consists of the same type of equations), the linear extrapolation is straightforward. For the examined functions, the CNF formula parameters could be a good metric for measuring the hardness of the formula but only to some extent. Indeed, the smallest formulas (SHA-1 and SHA-256) have the lowest security margin but, for example, BLAKE and Keccak have nearly the same security level while Keccak formula is more than twice as big.

Table 2. The parameters of our CNF formulas coding hash functions

Function	Variables	Clauses
SHA-1	29 000	200 000
SHA-256	61 000	400 000
BLAKE-256	57 000	422 000
Keccak-256	88 000	1 075 000
Skein-512-256	148 000	1 041 000
JH-256	169 600	1 998 000
Groestl-256	279 000	3 568 000

Besides the attacks on (round-reduced) hash functions, we have also mounted the attacks on the compression functions — main building blocks of hash functions. First we tried the preimage attack on a given compression function and if it did not succeed we attacked the function in a scenario where an adversary can choose IV (initial value) and get a pseudo-preimage. Table 3 summarizes these attacks. Similarly as for the results from Table 1, the numbers remain valid for all three types of attacks (a preimage, a second preimage and a collision attack). Among the finalists our best attack was on 6-round Skein-512-256 compression function for which we found pseudo-collisions. For comparison, the Skein’s authors reached 8 rounds but they found only a pseudo-near-collision. For Groestl compression function we were not able to mount any successful attack. Keccak has a completely different design from MD hash function family — there is no a typical compression function taking IV and a block of a message. Therefore in Table 3 we do not report any result for these two functions.

It is very difficult to give a good and clear answer which designs or features of a hash function are harder for SAT solvers. One observation we have made is that those designs which use S-boxes (JH and Groestl) have the biggest CNF formula and are among the hardest for SAT solvers. Equations describing an S-box are more complex than equations describing addition or boolean AND. Consequently, the corresponding CNF formula for the S-box is also more complex with greater number of variables and clauses than in the case of other typical operations. Before we give an example, let us first take a closer look at the addition operation. This operation is used in SHA-1, SHA-256, BLAKE, and Skein. In our toolkit the addition of two words is described by the following equations (a full adder equations):

Table 3. Attacks on the compression functions

Function	Type of attack	No. of rounds	Security margin
SHA-1	preimage	21	74% (21/80)
SHA-256	preimage	16	75% (16/64)
BLAKE-256	preimage	1	93% (1/14)
JH-256	preimage	2	96% (2/42)
Skein-512-256	pseudo-preimage	6	92% (6/72)

$$S_i = A_i \oplus B_i \oplus C_{i-1}$$

$$C_i = (A_i \cdot B_i) \oplus (C_{i-1} \cdot (A_i \oplus B_i))$$

S_i is the i -th bit of the sum of two i -bit words A and B. C_i is the i -th carry output.

Now let us compare the CNF sizes of the addition operation and AES S-box used in Groestl. A CNF of 32-bit addition has 411 clauses and 124 variables while AES S-box given to our toolkit as a look-up table gives a CNF with 4800 clauses and 900 variables. We also experimented with an alternative description of AES S-box expressed as boolean logic equations, instead of a look-up table [14]. This description reduces the CNF size approximately by half but still it is a degree of order greater than the CNF from the 32-bit addition operation.

We have also observed that there is no clear limit in size of CNF formulas beyond which a SAT solver fails. For example the CNF of 2-round JH with 59 thousand clauses is solved within seconds whereas the CNF of 2-round Skein with 27 thousand clauses was not solved having 30 hours of time limit. What exactly causes the difference between hardness of formulas is a good point for further research.

4 Conclusion

The security margin of the five finalists of the SHA-3 contest using our SAT-based cryptanalysis has been evaluated in this paper. A new toolkit which greatly simplifies the most tedious stages of this type of analysis and helps to mount the attacks in a uniform way has been proposed and developed. Our toolkit is more flexible than the existing tools and can be applied to various cryptographic primitives. Based on our methodology, we have shown that all the finalists have substantially bigger security margin than the current standards SHA-256. We stress that ‘bigger security margin’ we refer only to the context of our SAT-based analysis. Using other techniques (e.g., linear cryptanalysis) could lead to a different conclusion.

As a side effect of our security margin evaluation, we have also carried out some attacks on compression functions and reported some new state-of-the-art results. For example, we have found pseudo-collisions for 6-round Skein-512-256 compression function.

References

1. Audemard, G., Simon, L.: Glucose SAT Solver, <http://www.lri.fr/~simon/?page=glucose>
2. Schneier, B., et al.: The Skein Hash Function Family, <http://www.skein-hash.info/sites/default/files/skein1.1.pdf>
3. Bernstein, D.J.: Second preimages for 6 (7? (8??)) rounds of Keccak? NIST mailing list (2010), http://ehash.iaik.tugraz.at/uploads/6/65/NIST-mailing-list_Bernstein-Daemen.txt
4. Biere, A.: Lingeling, <http://fmv.jku.at/lingeling>
5. Biryukov, A., Dunkelman, O., Keller, N., Khovratovich, D., Shamir, A.: Key Recovery Attacks of Practical Complexity on AES Variants With Up To 10 Rounds. Cryptology ePrint Archive, Report 2009/374 (2009), <http://eprint.iacr.org/2009/374>
6. Bouillaguet, C., Derbez, P., Dunkelman, O., Keller, N., Rijmen, V., Fouque, P.A.: Low Data Complexity Attacks on AES. Cryptology ePrint Archive, Report 2010/633 (2010), <http://eprint.iacr.org/2010/633>
7. Cook, S.A.: The complexity of theorem-proving procedures. In: Proceedings of the Third Annual ACM Symposium on Theory of Computing, STOC 1971, pp. 151–158. ACM, New York (1971)
8. Cook, S.A., Mitchell, D.G.: Finding hard instances of the satisfiability problem: A survey, pp. 1–17. American Mathematical Society (1997)
9. Courtois, N.T., Pieprzyk, J.: Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 267–287. Springer, Heidelberg (2002)
10. Davis, M., Logemann, G., Loveland, D.: A machine program for theorem-proving. Communications of the ACM 7(5), 394–397 (1962)
11. Davis, M., Putnam, H.: A computing procedure for quantification theory. Journal of the ACM 7, 201–215 (1960)
12. Fiorini, C., Martinelli, E., Massacci, F.: How to fake an RSA signature by encoding modular root finding as a SAT problem. Discrete Applied Mathematics 130, 101–127 (2003)
13. Bertoni, G., et al.: Keccak sponge function family main document, <http://keccak.noekeon.org/Keccak-main-2.1.pdf>
14. Gaj, K., Chodowicz, P.: FPGA and ASIC Implementations of AES. In: Koc, C.K. (ed.) Cryptographic Engineering, ch. 10, pp. 235–294. Springer (2009)
15. Aumasson, J.P., et al.: SHA-3 proposal BLAKE, <http://www.131002.net/blake/>
16. Jovanović, D., Janičić, P.: Logical Analysis of Hash Functions. In: Gramlich, B. (ed.) FroCos 2005. LNCS (LNAI), vol. 3717, pp. 200–215. Springer, Heidelberg (2005)
17. Lala, P.K.: Principles of modern digital design. Wiley-Interscience (2007)
18. Massacci, F.: Using Walk-SAT and Rel-SAT for cryptographic key search. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 290–295 (1999)
19. Mendel, F., Thomsen, S.: An Observation on JH-512 (2008), http://ehash.iaik.tugraz.at/uploads/d/da/Jh_preimage.pdf
20. Mironov, I., Zhang, L.: Applications of SAT Solvers to Cryptanalysis of Hash Functions. In: Biere, A., Gomes, C.P. (eds.) SAT 2006. LNCS, vol. 4121, pp. 102–115. Springer, Heidelberg (2006)
21. Gauravaram, P., et al.: Grøstl — a SHA-3 candidate, <http://www.groestl.info>

22. Rivest, R., et al.: The MD6 hash function, <http://groups.csail.mit.edu/cis/md6/>
23. Schlaffer, M.: Updated Differential Analysis of Grøestl. Grøestl website (January 2011), <http://groestl.info/groestl-analysis.pdf>
24. Soos, M.: CryptoMiniSat 2.5.0. In: SAT Race Competitive Event Booklet (July 2010), <http://www.msoos.org/cryptominisat2>
25. Soos, M.: Grain of Salt — An Automated Way to Test Stream Ciphers through SAT Solvers. In: Workshop on Tools for Cryptanalysis (2010)
26. Soos, M., Nohl, K., Castelluccia, C.: Extending SAT Solvers to Cryptographic Problems. In: Kullmann, O. (ed.) SAT 2009. LNCS, vol. 5584, pp. 244–257. Springer, Heidelberg (2009)
27. Wang, X., Yin, Y.L., Yu, H.: Finding Collisions in the Full SHA-1. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 17–36. Springer, Heidelberg (2005)
28. Wu, H.: Hash Function JH, <http://www3.ntu.edu.sg/home/wuhj/research/jh/>

Appendix

For the reader's convenience, we provide an example SystemVerilog code for SHA-1 used in the experiments with our toolkit. In many cases a code strongly resembles a pseudocode defining a given cryptographic algorithm. A reader familiar with C or Java should have no trouble adjusting the code to our toolkit's needs.

```

module sha1(IN, OUT);
    input [511:0] IN; // input here means 512-bit message block
    output [159:0] OUT; // output here means 160-bit hash
    reg [159:0] OUT;
    reg [31:0] W_words [95:0]; // registers for W words
    reg [31:0] h0 ,h1, h2, h3, h4;
    reg [31:0] a, b, c, d, e, f, k, temp, temp2;
    integer i;

    always @ (IN, OUT)
    begin

h0 = 32'h67452301; h1 = 32'hEFCDB89;
h2 = 32'h98BADCFE; h3 = 32'h10325476;
h4 = 32'hC3D2E1F0;

a = h0; b = h1; c = h2; d = h3; e = h4;

W_words[15] = IN[31:0]; W_words[14] = IN[63:32];
W_words[13] = IN[95:64]; W_words[12] = IN[127:96];
W_words[11] = IN[159:128]; W_words[10] = IN[191:160];
W_words[9] = IN[223:192]; W_words[8] = IN[255:224];
W_words[7] = IN[287:256]; W_words[6] = IN[319:288];
W_words[5] = IN[351:320]; W_words[4] = IN[383:352];
W_words[3] = IN[415:384]; W_words[2] = IN[447:416];
W_words[1] = IN[479:448]; W_words[0] = IN[511:480];

```

```

for (i=16; i<=79; i=i+1)
begin
W_words[i] = W_words[i-3] ^ W_words[i-8] ^ W_words[i-14] ^ W_words[i-16];
W_words[i] = {W_words[i][30:0], W_words[i][31]}; // leftrotate 1
end

for (i=0; i<=79; i=i+1) // main loop
begin
if ((i>=0) && (i<=19))
begin
f = (b & c) | ((~b) & d); k = 32'h5A827999;
end
if ((i>=20) && (i<=39))
begin
f = b ^ c ^ d; k = 32'h6ED9EBA1;
end
if ((i>=40) && (i<=59))
begin
f = (b & c) | (b & d) | (c & d); k = 32'h8F1BBCDC;
end
if ((i>=60) && (i<=79))
begin
f = b ^ c ^ d; k = 32'hCA62C1D6;
end

temp2 = {a[26:0], a[31:27]}; // a leftrotate 5
temp = temp2 + f + e + k + W_words[i];
e = d;
d = c;
c = {b[1:0], b[31:2]}; // b leftrotate 30
b = a;
a = temp;
end // end of main loop

h0 = h0 + a; h1 = h1 + b;
h2 = h2 + c; h3 = h3 + d; h4 = h4 + e;
OUT = {h0, h1, h2, h3, h4}; //HASH
end
endmodule

```

Usage Control Model Specification in XACML Policy Language

XACML Policy Engine of UCON

Um-e-Ghazia, Rahat Masood, Muhammad Awais Shibli, and Muhammad Bilal

National University of Science and Technology, Islamabad, Pakistan
{10msccssghazia, awais.shibli, 10msccsmmasood,
m.bilal}@seecs.edu.pk

Abstract. Usage control model (UCON) is one of the emerging and comprehensive attribute based access control model that has the ability of monitoring the continuous updates in a system making it better than the other models of access control. UCON is suitable for the distributed environment of grid and cloud computing platforms however the proper formulation of this model does not exist in literature in any policy specification standard. It is for this reason that UCON is not widely adopted as an access control model by industry, though research community is now paying attention to make standard policy specification for this model. In this paper we are suggesting the interpretation of UCON model in extensible access control markup language (XACML) which is an OASIS standard of access control policies. We also highlight UCON model features by explaining its core processes and characteristics with respect to the case study of financial application.

Keywords: Access Control, Authorization, Obligation, Condition, Policy, Attribute.

1 Introduction

Access control models play vital role in protecting digital resources in a way to control and mediate access from unauthorized users. These models assure the security requirements of different applications improving their protection from unauthorized access. They secure digital information and resources until the permission is granted to user and are suitable for closed domain comprising static entities. Continuous evolution of computing platforms demands advanced security procedures and mechanisms to handle the consequences of unauthorized access.

Usage control model (UCON) has been proposed by Park and Sandhu that caters heterogeneous and dynamic environment conditions [1]. It is an attribute based access control that judges three factors: authorization, obligations and conditions for access decision. Two distinctive characteristics are attribute mutability (updates) and access decision continuity which augments UCON model than other traditional access control models. In UCON model usage session is maintained that is categorized into

three parts: pre access, ongoing access and post access phases. When the subject S is accessing resource R , UCON decision factors i-e. attributes, obligations and conditions are evaluated before granting access which is called pre access phase. Evaluation is also performed during the time when R is in use by S called as ongoing access phase. In some cases, it is required to evaluate attributes and obligations after the completion of usage session which is known as post access phase. Decision is continuously evaluated during these three phases to monitor the changes in attribute values. In addition to it, UCON model is comprehensive enough to cover the traditional access control models. So it can be used to provide the better protection of system resources in a collaborative and dynamic environment. Despite of all the excellent features, UCON is not widely adopted as an access control model to restrict the unauthorized access. The major reason for this is that the model features are not been translated in any of the standard policy language to offer the proper formulation of model..

XACML [17] being a generic policy language of OASIS standard is suitable to represent the aforementioned features of UCON model. It describes the platform independent access control request/ response mechanism and policy specification. Its interoperability feature makes it widely used for various platforms and environments. XACML offers the extension points by introducing new data types, identifiers, elements and functions to offer a generic policy structure. Since UCON can facilitate the diverse range of applications like digital rights management (DRM), health care systems and social networking, it is highly encouraged to provide the formal specification of model in generic policy language like XACML. There is a need to define the separate profile of UCON in XACML that will enable organizations to adopt this flexible model. Also to guarantee the accurate access decision in different deployment scenarios, it is mandatory to propose the required alterations and additions in generic policy language of XACML which is not developed so far. We wish to propose the implementation of UCON model in XACML by incorporating additional elements and specify the information flow of different UCON processes in this paper.

Organization of paper is as follows: Section 2 presents the detailed UCON model and its core features, Section 3 includes the profile of UCON model in XACML, use of this profile is explained in Section 4 by taking the scenario of posting vouchers service in financial applications and Section 5 concludes the paper and present future directions.

2 Usage Control Model

UCON being an attribute based access control model accommodates the security requirements by the addition of more than one decision factors which makes it more reliable and flexible [3]. This model primarily restricts the usage of digital objects and provides the efficient mechanism to include the traditional access control models. Previous access control models only encompass authorization rules in making access decision; rather UCON model also consider the obligations and environmental conditions. Moreover collaborative environments demand the need of enhanced provisioning and controlled access to digital resources. In addition to the

immutable attributes that are explicitly modified by the administrator, system controlled mutable attributes are also managed by constant monitoring throughout the stages of usage session.

UCON model identifies three types of subjects; consumer, provider and identifyee. Consumers are the subjects who make request to perform certain action on object. Providers are the individuals who own services and issue the rights to the requesting party. Identifyee is the entity whose confidential information is incorporated within digital object. It is an optional group of subjects which may or may not be present depending on system requirements however it is always present in case of systems having users' confidential information. Depending on the job functions of subjects, three types of rights (actions) are specified namely consumer, provider and identifyee rights which indicates the set of actions or privileges on digital objects [1]. Apart from these, there are other actions as well that fall in the category to perform updates in attributes values during the phases of usage session which are termed as usage control actions [2].

UCON model also classify the objects as privacy sensitive and privacy non sensitive objects that determines whether the object contains critical information of identifyee subject or not. Improper management of privacy sensitive objects cause security breaches which results in data disclosure to unauthorized users and compromising data integrity. There is another phenomenon of UCON model called as reverse UCON in which the position of consumers and providers are inverted depending on the scenario. Complete classification of UCON subjects, objects and rights is shown in Figure 1.

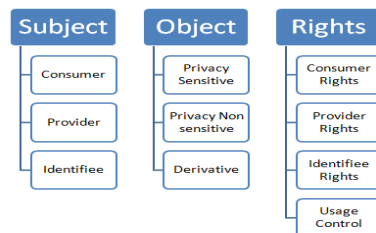


Fig. 1. UCON classification of subject, object and rights

All the traditional models include authorization rules that need to be satisfied before the using the resources called as rights related authorization rules. UCON identify additional obligations related authorization rules which are the set of actions related to access request to be completed before granting permissions of resources. Obligations are first time addressed by UCON model to improve the accuracy of access decision by enforcing users to perform certain actions before access. Obligations also act as functions that check whether the obligatory actions are fulfilled by the requesting entity or not. In order to further increase the accuracy of authorization decision, another factor to be considered important in UCON model are the environmental conditions such as ip addresses, current date or time. Conditions can be of two types; dynamic (stateful) and static (stateless). Dynamic conditions have constantly

changing information so they need to be evaluated for every update and static conditions do not have to be checked for each update during usage session [1].

2.1 Authorization Models of UCON

UCON Pre authorization models are carried out in the same way as for traditional authorization models. In this model, user credentials and resource attributes are checked before granting permission for the requested resource. Pre authorization models can be with immutable and mutable attributes (pre, ongoing and post updates) in which attributes either remain same during access phase or their values change before, during or after the access phase. Ongoing authorization models evaluate the attribute values during the resource usage by the user to perform the continuous verification. Ongoing authorization models can also be with immutable and mutable attributes as with pre authorization models. The effect of ongoing authorization evaluation in access control model is that the access rights can be revoked during the session as certain attribute value is changed.

2.2 Obligation Models of UCON

Obligation models comprise the pre and ongoing obligation monitoring of access request. They improve the accuracy of access decision in a way that it requests user to perform the access related mandatory actions first and then access would be granted.. Obligations that are used before and during the usage session are termed as pre and ongoing obligations respectively. Post obligations are introduced in order to execute actions after access session is finished i-e access fulfillment notifications to service provider. These post obligations can affect the decision of future usage sessions by generating request to policy repository for policy modification [14]. Pre, ongoing and post obligation models with immutable and mutable attributes perform the obligation checking before, during or after the access phase.

2.3 Condition Models of UCON

UCON has two condition models; *pre and ongoing condition with immutable attributes* to cater the environmental constraints and system related parameters. Conditions are evaluated before and during the session in the same way as authorization rules. Rather condition models do not consider the mutable attributes such as the changing location of a subject.

3 UCON Model Specification in XACML

In order to provide UCON model specification in XACML, we are going to introduce additional identifiers and attribute values to incorporate the features of UCON model. XACML has identifiers for subject categories like access-subject, recipient-subject, intermediary subject. They are used under the tag of

AttributeDesignator that is one of the methods of attributes retrieval in XACML. Subject-type identifier is introduced for UCON subject categories and their values might be consumer, provider and identifyee. UCON subject identifiers are used along with XACML subject category access-subject to further specify the type of accessing subject as follows.

```
<AttributeDesignator
Category=urn: oasis: names: tc: xacml: 1.0: subject-category:
access-subject
Type=urn: oasis: names: tc: xacml: 3.0: subject-type: consumer>
```

In the same way, action-type identifier is introduced with the XACML attribute category of action to reflect the UCON categories of action i-e consumer, provider, identifyee and usage control actions.

```
<AttributeDesignator
Category=urn: oasis: names: tc: xacml: 3.0: attribute-category:
action
Type=urn: oasis: names: tc: xacml: 3.0: action-type: usage-
control>
```

For demonstrating UCON objects, resource category identifier is created under attribute category of resource that has the value of privacy-sensitive, privacy-nonsensitive or derivative.

```
<AttributeDesignator
Category=urn: oasis: names: tc: xacml: 3.0: attribute-category:
resource
Type=urn: oasis: names: tc: xacml: 3.0: resource-category: de-
rivative>
```

Since the UCON model consider both mutable (updating values) as well as immutable (constant values) attributes, new identifier attribute-class is constructed to differentiate between them. Mutable attributes are then further narrow down into pre-mutable, ongoing mutable and post mutable. This general classification of attributes is used with all of the attribute categories of subject, resource, action and environment.

Rights related authorization rules can be mapped in XACML as general rules but the time of evaluating these rules needs to be managed. So the pre and ongoing element of authorization rules is explained by the rule-id attribute of the rule element.

```
RuleId="urn: oasis: names: tc: xacml: 3.0:pre-authorization"
```

Obligation element is specified in XACML for mandatory actions required to be performed by the subject that can also handle the obligation related authorization rules of

UCON. Obligation expression element includes arguments that are required to execute the obligation. We have proposed the new attribute namely `Fulfill-phase` for specification of pre and ongoing obligations. It indicates the access phase during which obligation must have to be satisfied by PEP, so it may have the values of `pre-access`, `ongoing-access` that reveals the pre and ongoing obligations.

```
<ObligationExpression
ObligationId="urn: oasis: names: tc: xacml: ucon-example: obli-
gation: license-agreement
Fulfill-phase="pre-access">
```

Condition element in XACML contain single expression element which includes functions to be evaluated. We have introduced additional attribute `condition-type` under the condition element to present the UCON dynamic and static conditions. Furthermore pre and ongoing element of condition models are expressed by introducing new attribute called as `evaluation-phase` under the condition element. It can have the value of `pre-access` or `ongoing access`.

```
<Condition
Condition-type = "urn: oasis: names: tc: xacml: 3.0: condition-
type: dynamic
Evaluation-phase= ongoing-access">
```

3.1 UCON Access Control Framework

We are proposing an access control framework which has the policy information flow modules of XACML like PDP, PEP, PIP and PAP. Policy repository is a unit in XACML that resides between the PAP and PDP containing the access control policies of corresponding model. Generally PAP creates the access control policies and pushed them into policy repository to be used for evaluation by PDP. We have proposed additional module integrated with PAP that has the capability to interpret the newly created UCON identifiers, attributes and their values. This interpretation will help to determine whether the processing is to be performed in before, ongoing or after phase of usage session. This module is called as *UCON policy builder* that incorporates UCON model features in policy specification, when PAP accepts the inputs from policy administrator to formulate the generic UCON policy (Fig. 2).

In addition to it, *UCON policy engine* module is incorporated with PDP that provides the main features of UCON model like attribute mutability and decision continuity. *UCON policy engine* has sub modules like *attribute mutability (AM)* and *decision continuity (DC)* as shown in Figure3. *AM* handles the updating of attribute values in three access phases; before, during and after access. As a result of attribute values modification, *DC* monitors the continuous policy evaluation of three access decision factors; authorization, obligation and condition. So the pre and ongoing models of authorization, obligation and conditions are deployed accordingly with mutable and immutable attributes.

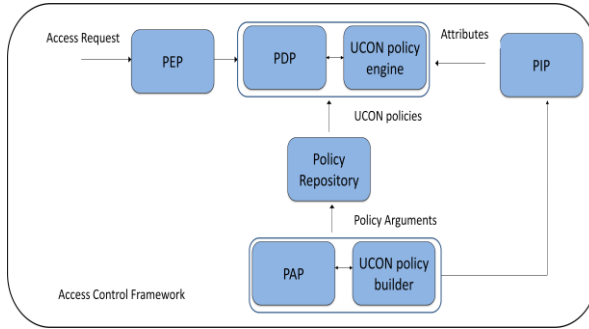


Fig. 2. UCON Access Control Framework

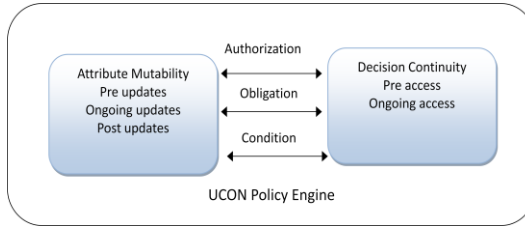


Fig. 3. UCON Policy Engine

UCON policy engine in conjunction with PDP evaluates the combined effect of target, condition, rules and obligations of policy or policy set and applies the specified combining algorithms to generate the final response for user. Functions for continuous policy evaluation are embedded in policy engine to carry out the constant monitoring of attributes, obligations and conditions in order to support the ongoing feature of UCON model. *UCON policy engine* in combination with *policy builder* will perform different functions such as obligation monitoring, attributes management, maintaining history of updated events of certain access request, notifies user about the current policy status for request.

4 Example Scenario

UCON model can be widely adopted in distributed environment to manage the controlled access for applications like health care systems, database management systems, resource sharing systems etc. We are considering the financial application to demonstrate the proposed specification of UCON model in XACML. Managing financial data is one of the key challenges in large enterprises. Financial data is comprised of financial entries regarding its employees, customers, vendors, assets, inventory, products, cost and profit centers etc hence play a vital role in enterprise progress and development. Often the highly secret and confidential in-formation related to enterprise and their employees also reside in these applications, thereby it is required

to provide security in terms of data privacy, data loss and data access. General Ledger (GL) is a key component of financial applications that acts as a central repository of company's accounting transactions. All the business transactions are posted to GL in the form of journal vouchers having debit and credit card entries. These entries are further classified into assets, liabilities, revenues, expenses, capital/owner equity. Core modules of GL includes chart of accounts, financial calendars, journal entries, ledgers, trial balance, balance sheet, profit and loss statement, cash flow statement and user defined report writer.

In this section, we will demonstrate our proposed methodology through the journal entries module. Journal entries are broadly categorized into five types like bank payment voucher, cash payment voucher, bank receipt voucher, cash receipt voucher and journal voucher. Every user of application is not authorized to create/post journal entries rather a user with specific attributes like *name*, *department* and *role* can create/post voucher for certain amount. For authorized users, they are not allowed to create/post entries for all accounts. Moreover an authorized user with permissions to create/post journal entries further requires quota/limit on journal entry amount for the permissible account. Different accounts act as object and the attributes allotted them are *accessing list* (specifies which user can access this account), *limit of total amount* (represents value that a user can post for a voucher). In this scenario policy is formulated as; user *John* having a role of *Director General* belonging to *Administration* department cannot post vouchers beyond the limit of 50,000\$ in a day time. Further he can only reference *City Bank Account # 345678B, 657463C* in the credit entries of all vouchers. This scenario workflow is based on pre authorization and ongoing authorization, pre obligation and ongoing obligation and pre condition that are described below according to aforementioned scenario and their corresponding XACML code snippets.

Before providing access to journal entry management service, above specified subject and object attributes are verified according to policy specification that corresponds to pre authorization mechanism. UCON policy engine will represent this pre authorization mechanism for the attribute *subject-id* "John" as follows.

```
<Rule RuleId="urn: oasis: names: tc: xacml: 2.0: ucon-example:
ongoing-authorization"...>
  <AttributeDesignator Category="urn: oasis: names: tc:
xacml: 1.0: subject-category: access-subject"
  Type="urn: oasis: names: tc: xacml: 3.0: subject-type:
consumer"
  AttributeId="urn: oasis: names: tc: xacml: 1.0: subject:
subject-id"
  DataType="http://www.w3.org/2001/XMLSchema#string"
  Class="urn: oasis: names: tc: xacml: 3.0: attribute-class:
immutable"/>
```

Since *voucher limit* of *John* is fifty thousand for a day, posting voucher service is revoked as limit reaches before a day time. Request is no more facilitated and a message is prompt to user that he is not eligible to post a voucher in present day. It is an example of *ongoing authorization model* of UCON which is shown below.

```

<Rule RuleId= "urn: oasis: names: tc: xacml: 3.0: ongoing-
authorization"...>
  <AttributeDesignator Category="urn: oasis: names: tc:
xacml: 1.0: subject-category: access-subject"
Type="urn: oasis: names: tc: xacml: 3.0: subject-type: con-
sumer"
AttributeId="urn: oasis: names: tc: xacml: 1.0: subject:
subject-voucher-limit"
DataType="http://www.w3.org/2001/XMLSchema#double"
Class="urn: oasis: names: tc: xacml: 3.0: attribute-class:
ongoing-mutable"/>

```

Accessing list attribute of account object is updated after the usage session which is the example of post update. This attribute keeps record of users that are accessing certain account, so it is classified as privacy-sensitive object. It is also useful for auditing and management purposes.

```

<Rule RuleId= "urn: oasis: names: tc: xacml: 3.0: ongoing-
authorization"...>
  <AttributeDesignator Category="urn: oasis: names: tc:
xacml: 3.0: attribute-category: resource"
Type="urn: oasis: names: tc: xacml: 3.0: resource-
category: privacy-sensitive"
AttributeId="urn: oasis: names: tc: xacml: 1.0: subject:
accessing-list"
DataType="http://www.w3.org/2001/XMLSchema#string"
Class="urn: oasis: names: tc: xacml: 3.0: attribute-class:
post-mutable"/>

```

Obligations are categorized as system related obligations and subject related obligations [14]. System related obligations are those actions that are executed by the service provider in order to ensure the verification of requesting party. On the other hand subject related obligations are performed by the requesting subject which is enforced by the service provider. In the present situation, subject related pre obligation is to accept the application terms and conditions before accessing journal management service. This subject related obligation is performed just once when the subject requests to access the service for the first time.

```

<ObligationExpression ObligationId="urn: oasis: names: tc:
xacml: ucon-example: obligation: license-agreement"
Fulfill-phase="pre-access">
  <AttributeAssignmentExpression
    <AttributeDesignator Category="urn: oasis: names: tc:
xacml: 1.0: attribute-category: resource
AttributeId="urn: oasis: names: tc: xacml: 1.0: re-
source: window-id"

```

```

</AttributeAssignmentExpression>
<AttributeAssignmentExpression
  <AttributeDesignator Category="urn: oasis: names: tc:
xacml: 1.0: subject-category: access-subject"
  AttributeId="urn: oasis: names: tc: xacml: 1.0: sub-
ject: subject-id">
  </AttributeAssignmentExpression>
</ObligationExpression>

```

As the specific limit of voucher is reached for particular user, insert option for voucher becomes invisible to that user. This is also an example of *ongoing obligation* performed by service provider. *Voucher limit* for *John* is fifty thousand in current scenario, after that the post voucher option is disable for him.

```

<ObligationExpression ObligationId="urn: oasis: names: tc:
xacml: ucon-example: obligation: disabling-post-voucher"
Fulfill-phase="ongoing-access">
  <AttributeAssignmentExpression
    <AttributeDesignator Category="urn: oasis: names: tc:
xacml: 1.0: attribute-category: resource
    AttributeId="urn: oasis: names: tc: xacml: 1.0: re-
source: disable-post-voucher-option">
    </AttributeAssignmentExpression>
  </ObligationExpression>

```

Further conditional parameters might be incorporated as pre condition in the form of user specific ip-address which helps to determine the user location before access. In this case, policy condition element states that specific ip-addresses of “x.x.x.x and “y.y.y.y” (which falls in the organization subnet) can access the service of posting journal voucher.

```

<Condition Condition-type = "urn: oasis: names: tc: xacml: 3.0:
condition-type: static"
Evaluation-phase= pre-access"
FunctionId="urn: oasis: names: tc: xacml: 1.0: function: not"
  <Apply
    FunctionId="urn: oasis: names: tc: xacml: 1.0: func-
tion: string-at-least-one-member-of">
    <AttributeDesignator
      AttributeId="urn: internetexplorer: names: internetex-
plorer: 2.1: environment: httpRequest: clientIpAddress"
      DataType="http://www.w3.org/2001/XMLSchema#string"/>
    <Apply FunctionId="urn: oasis: names: tc: xacml: 1.0:
function: string-bag"
    <AttributeValue

```

```

Data-
Type="http://www.w3.org/2001/XMLSchema#string">127.0.0.
1
</AttributeValue>
<AttributeValue
Data-
Type="http://www.w3.org/2001/XMLSchema#string">128.84.1
03.11
</AttributeValue>
</Apply>
</Apply>
</Condition>

```

5 Conclusion and Future Work

For the applicability of UCON model features in XACML, some of the corresponding modifications are proposed in paper to cater the authorization requirements of distributed environment. Our main objective was to develop the comprehensive UCON framework in XACML that should be reliable, flexible and scalable. XACML being a generic policy language can better translate the UCON model features. It will provide model specifications in a formal manner to be deployed as an access control model by practical application environment. This framework provides the policy administration interface that will help enterprises in implementing UCON model within their applications. It will be generic and consistent to accept the arguments according to scenario and improve the accuracy of access decision.

Future directions in this domain include the identification of issues and problems in UCON model with respect to distributed and collaborative platforms like grid and cloud computing. In addition to this, detailed performance analysis, usability and interoperability issues of UCON model need to be addressed. UCON models of authorization, obligations and conditions can further be investigated to integrate them with each other and to show interactions between parallel usage sessions. Moreover, UCON access control framework can be further extended to provide the main feature of extensibility. Different access control models can be incorporated within this framework to offer uniformity and consistency across applications. Depending on application security requirements, organizations can select any model with much more flexibility and ease. Distributed environments such as cloud computing can adopt this generic access control framework to provide better resource protection.

References

1. Park, J., Sandhu, R.: Towards Usage Control Models: Beyond Traditional Access Control. In: SACMAT 2002 Proceedings of 7th ACM Symposium on Access Control Models and Technologies (2002)

2. Zhang, X.: Formal model and analysis of usage control, PhD Thesis. George Mason University, Fairfax, USA (2006)
3. Lazouski, A., Martinelli, F., Moore, P.M.: Usage control in computer security: A survey. *Elsevier Journal of Computer Science Review* 4(2) (2010)
4. Park, J., Sandhu, R.: The UCON ABC Usage Control Model. *Journal of ACM Transactions on Information and System Security* 7(1) (2004)
5. Zhang, X., Nakae, M., Covington, M., Sandhu, R.: Toward a Usage-Based Security Framework for Collaborative Computing Systems. *Journal of ACM Transactions on Information and System Security* 11(1) (2008)
6. Kumaraguru, P., Cranor, L.F.: A Survey of privacy policy languages. In: *SOUPS 2007 Proceedings of Third Symposium on Usable Privacy and Security* (2007)
7. Gougliadis, A., Mavridis, I.: On the Definition of Access Control Requirements for Grid and Cloud Computing Systems. In: *GridNets 2009 Third International ICST Conference* (2009)
8. Zhang, X., Parisi-Presicce, F., Park, J., Sandhu, R.: A Logical Specification of Usage Control. In: *SACMAT 2004 ACM Transactions on Information and System Security* (2000)
9. Lu, J., Li, R., Varadharajan, V., Lu, Z., Ma, X.: Secure Interoperation in Multidomain Environments Employing UCON Policies. In: Samarati, P., Yung, M., Martinelli, F., Ardagna, C.A. (eds.) *ISC 2009. LNCS*, vol. 5735, pp. 395–402. Springer, Heidelberg (2009)
10. Subashini, S., Kavitha, V.: A survey on security issues in service delivery models of cloud computing. *Elsevier Journal of Network and Computer Applications* 34(1) (2010)
11. Haidar, D.A., CuppensBouahia, N., Cuppens, F., Debar, H.: An Extended RBAC profile in XACML. In: *SWS 2006 Proceedings of the 3rd ACM Workshop on Secure Web Services* (2006)
12. Chen, D., Huang, X., Ren, X.: Access Control of Cloud Services Based on UCON. In: *CloudCom 2009 Proceedings of the 1st International Conference on Cloud Computing* (2009)
13. Ali, T., Nauman, M., Fazl-e-Hadi, Muhaya, F.B.: On Usage Control of Multimedia Content in and through Cloud Computing Paradigm. In: *5th International Conference on Future Information Technology* (2010)
14. Katt, B., Zhang, X., Breu, R., Hafner, M., Seifert, J.-P.: A General Obligation Model and Continuity-Enhanced Policy Enforcement Engine for Usage Control. In: *SACMAT 2008 Proceedings of the 13th ACM Symposium on Access Control Models and Technologies* (2008)
15. xacml-3.0 core specifications, eXtensible Access Control Markup Language (XACML) Version 3.0 (April 2012), <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.pdf>
16. xacml-3.0 rbac specifications, XACML v3.0 Core and Hierarchical Role Based Access Control (RBAC) Profile Version 1.0 (August 2010), <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-rbac-v1-spec-cs-01-en.pdf>
17. A Brief Introduction to XACML, http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html

TIDS: Trust-Based Intrusion Detection System for Wireless Ad-hoc Networks

Novarun Deb and Nabendu Chaki

University of Calcutta

novarun.db@gmail.com, nabendu@ieee.org

Abstract. This paper aims to propose a new trust-based Intrusion Detection system (IDS) for wireless, ad-hoc networks with or without mobility of nodes. In fact, the proposed solution not only detects intrusions but also proactively responds towards route setup avoiding the compromised nodes. It could be extended for mesh or hybrid networking environment too. Trust is evaluated as the weighted sum of direct evaluation of the neighboring nodes as well as from the indirect references. A sliding window is defined on the time scale and the IDS is to be evoked after every time slice. Indirect reference is derived from the recommendations of those 1-hop neighbors of the target node that are also neighbors of the evaluating node. The performance of the proposed algorithm has been evaluated using the Qualnet network simulator. Simulation results also establish superiority of the proposed algorithm over HIDS, another recent trust-based IDS for wireless ad-hoc network.

Keywords: Greyhole attack, Denial of Service attack, RREQ packet, RREP packet, Trust Request packet.

1 Introduction

Wireless and cellular networks have tremendously grown over the last four to five years. New technologies have been deployed in this domain with MANETs and Wireless Mesh Networks among the most notable ones. Sensor networks are also finding major applications such as Border Area Surveillance or Disaster Recovery Management. Also, end-user requirements have resulted in cellular and mobile networks being exploited to their fullest. Millions of applications are being used by customers that inherently demand security. This is where Intrusion Detection System plays a very important role. Most wireless network technologies have energy constrained nodes. Consequently, computation intensive procedures are often avoided. Thus, unlike traditional hardwired networks, intrusion prevention is not at all an option for wireless ad-hoc networks as these are quite computation intensive.

Intrusion detection is one of those safety mechanisms that is energy-efficient as well. Also, more effective is an Intrusion Response System that takes some corrective measures once an intruder is detected. This paper aims to propose a new Intrusion Detection and Response System for Wireless ad-hoc networks, in general. The proposed solution not only detects intrusions during application traffic but can also be

proactively involved towards route setup. The most trusted route will be set up for better QoS. Trust is the basis of the proposed IDS. Several recently proposed trust models have been studied and reviewed in the state of the art section. All these models have certain shortcomings and are vulnerable to attacks under certain situations. Also, none of these trust models have been extensively tested on any network simulator for any type of comprehensive results. As part of the paper, a trust model has been proposed from which an intrusion detection algorithm has been designed.

The rest of the paper is structured as follows. Section 2 describes the State of the Art Review on Trust based IDS. Section 3 discusses in details the working of the Trust based algorithm (TIDS). Section 4 highlights the simulation results of our algorithm and compares its performance with another IDS algorithm HIDS[6]. Section 5 concludes the paper with Section 6 Acknowledgements and Section 7 listing the references.

2 State of the Art

Various models have been proposed for sharing resources in a P2P environment. Quite often, these models fail to consider the trust of peers prior to resource sharing. PET [1] is a one of the highly cited trust models where a peer always trusts itself. Trust on a peer increases slowly but decreases rapidly. In [1], trust is evaluated quantitatively as the combination of two components – reputation and risk. Reputation is a long term assessment of the behavior of the peer in the past. Risk on the other hand is a short term assessment of the peer's most recent behavior. Reputation component of trust comprises of two components – recommendation and direct interaction. Recommendations dominate trust evaluation when there has been no direct interaction in the past. A weighted evaluation of these components is used in evaluating Reputation. Direct interaction information is also used for evaluating the Risk component of Trust. PET classifies peers based on the QoS provided by them. Four major categories of QoS are Good, No Response, Low Grade, and Byzantine behavior. Nodes are rewarded positively for Good behavior only. Nodes are negatively rewarded for the other three categories. The magnitude of negativity decreases from Low Grade through No Response and Byzantine behavior. Risk is evaluated as the amount of negative score earned due to bad services by the peer in a specific time interval.

In [2], Cho et. al. have proposed trust management for MANETs using trust chain optimization. Trust is evaluated based on four components – residue energy level and co-operation (QoS Trust) and honesty and closeness (Social Trust). The trust value of a node i is evaluated by a node j as the weighted sum of these four components. Residue energy level and honesty trust component values are binary, co-operation trust component is a probabilistic value based on the node's behavior in the last update interval, and closeness component is an integer representing the number of 1-hop neighbors of a node. Every node evaluates trust of its 1-hop neighbors by observing its behavior to packet forwarding. Trust evaluation is broadcast throughout the network in the form of status exchange messages.

Li Xiong and Ling Liu proposed a new trust model in PeerTrust [3]. PeerTrust computes the trust of peers in a network as a function of 3 components. First, a node N becomes trustworthy when other peers who have interacted with N find it to behave normally. Second is the context of satisfaction. It defines the total number of interactions that a node has performed with its peers. Finally the Balance factor of trust is used to reduce the effects of incorrect satisfaction information coming from malicious nodes. A trust metric $T(u)$ for node u is computed as the total satisfaction earned by u and multiplied by the balance factor of each peer and averaged over the total number of interactions that u has participated in. However, PeerTrust fails to capture the most recent malicious behavior of highly reputed nodes. This is taken care of by specifying a sliding window on the time scale. PeerTrust uses the P-Grid algorithm for distribution and aggregation of trust data across a P2P network. A key value is assigned to each peer based on its ID. Each node stores and maintains trust data about one or more peers in the network. As peers can behave maliciously, any intentional false trust data about a peer gets replicated in the local databases of more than one peer. This redundancy has its overhead. Such malicious behavior could be avoided by following a voting by consensus algorithm.

Wang, Mokhtar and Macaulay proposed a trust model based on the concept of H-index. C-index [4] incorporates the past experience a peer node has had with a collaborator. The more the number of trustworthy recommendations from a peer node, the higher should be the credibility of its recommendations. Also, trust models should consider the diversity of trustworthy collaborations. The larger the number of peer nodes with which a node collaborates, the greater is the reliability of its recommendation. Trust Depth in a community of nodes is measured as the number of Pure Positive Feedbacks (PPF) a node receives from its peer. It is defined as the difference between the number of satisfactory and unsatisfactory feedbacks from that node. Trust Breadth is the number of peers from which a node receives at least one PPF. Based on TD and TB, the C-index of a node is evaluated. The C-index of a node is used in evaluating its trust. It is defined as the number of peers (Z) in a community of N nodes which have sent at least ' Z ' PPFs to the node. The C-index mechanism of trust measurement is much more robust as it is immune to attacks as any single node sending multiple PPFs to a node does not affect its C-index. However, the method remains vulnerable to synergistic attacks. The C-index mechanism fails when the number of attackers is larger than the current C-index of a node.

In [5], Luo, Liu, and Fan have proposed a trust model based on fuzzy recommendation for MANETs. Trust is defined by 3 components – past experience, current knowledge about the entity's behavior, and recommendations from trusted entities. The Fuzzy Trust Model centers around a parameter called the Local Satisfaction Degree (S_{ij}). S_{ij} is the difference between the number of successful and unsuccessful transactions between two nodes i and j . The Fuzzy Indirect Trust Model is the generic trust model that evaluates trust from two component values – Direct Trust and Recommendation Trust. Direct Trust is evaluated by a node on its neighbor as a result of the interactions between them. Recommendation Trust depends on the recommendations provided by a neighbor about a distant node. Recommendation Trust is evaluated by a node transitively or by consensus. It is evaluated as the combination of

the Recommendation from the neighbor and the Direct Trust that the node has on that neighbor. The neighboring node makes a recommendation about the distant node based on what it receives from its neighbors, transitively. The node has Direct Trust evaluated for all its neighbors and each neighbor makes a Recommendation about the distant node. Consensus Recommendation Trust is the union of all these trust recommendations. However, recommendations from a highly trusted node remain questionable (e.g. synergistic effect of selfish nodes). Thus, trust value of a node is computed globally by combining recommendations from all nodes. RFS-Trust uses an adjusted cosine similar function to find the similarity between nodes i and j . The higher the degree of similarity, more consistent is the evaluation of trust between the respective nodes as compared to other nodes in the network. Thus, it is not a high range of trust values that makes a node's recommendation credible. Rather, credibility of recommendations increases with similarity in rating opinions.

3 The Proposed Trust-Based IDS (TIDS)

Intrusions need to be detected under varying circumstances. This paper focuses on two such scenarios where intrusion detection becomes essential. Since the proposed algorithm is Trust based, intruders are identified on the basis of their trust values. Intrusion detection is essential during route setup. Good Quality of Service can be ensured only when the most trusted route is setup between the source and the destination. Thus, trust value of nodes has to be considered when Route Request and Route Reply packets are being exchanged. The dynamically changing topology of the mobile ad-hoc network causes the routes between them to change frequently. In such a scenario, intrusion detection is even more important as nodes may change their behavior over time. As long as packets are being sent along a particular route, some intermediate nodes may start behaving selfishly or maliciously. In order to detect such intruders, the IDS algorithms are to be evoked at regular intervals. The network should react differently for destination nodes and intermediate nodes. Whenever a destination node is found to be an intruder, the application is terminated and the destination node is blacklisted. If an intermediate node is found to be an intruder, it is bypassed and the route is re-established. The malicious node is also blacklisted.

Before getting into the details of the IDS, let us consider some of the common attacks. The most commonly simulated attack in networking journals is the blackhole attack where a node drops all the packets that are sent through it. However, considering the fact that attackers are intelligent enough, a more practical and realistic attack is the greyhole attack or selective forwarding. Here, a node behaves as a good node to increase its reputation within the network. Once it becomes highly reputed, it starts dropping packets. Later, it again increases its reputation and prevents itself from being detected. There is also the Denial of Service (DoS) attack that can be implemented in more ways than one. The motive behind DoS attack is to consume the resources of the network so that peers are denied service. Detecting spurious packet generation would become all the more difficult if the DoS agent is a member on the route from the source to destination of some application. A stand-alone node that generates

spurious packets can be easily detected. Thus, it is assumed that both greyhole attackers and DoS agents will be on the route from the source to the destination of some application. Rather, during route setup, these attackers will ensure that a route is set up through them by returning corrupt reachability information about the destination. Keeping these attack scenarios in mind, one can conclude that intrusion detection needs to be done only for the nodes that lie on the route between a source and a destination of some application. Nodes that are not part of active applications will not be a part of the intrusion detection as well. This makes the proposed intrusion detection algorithm a lightweight process. All nodes in the network need not execute the intrusion detection algorithm redundantly.

3.1 Working Principle

Intrusion detection is mandatory during the route setup process. The solution proposed in this paper is based on the following working principles.

- Every node maintains trust information about its 1 – hop neighbors.
- Trust is evaluated as the weighted sum of 2 components – *Direct Valuation* and *Indirect Reference*.
- Direct Valuation is again a function of 2 factors – *Reputation* and *Risk*. Reputation is the measure of the long term evaluation of the behavior of a node. Risk is the valuation of the most recent behavior of the node.
- A sliding window is defined on the time scale. The Intrusion detection algorithm is executed after every time slice.
- Indirect Reference refers to the recommendations from 1 – hop neighbors of the “target node” which are also neighbors of the “valuation node”.

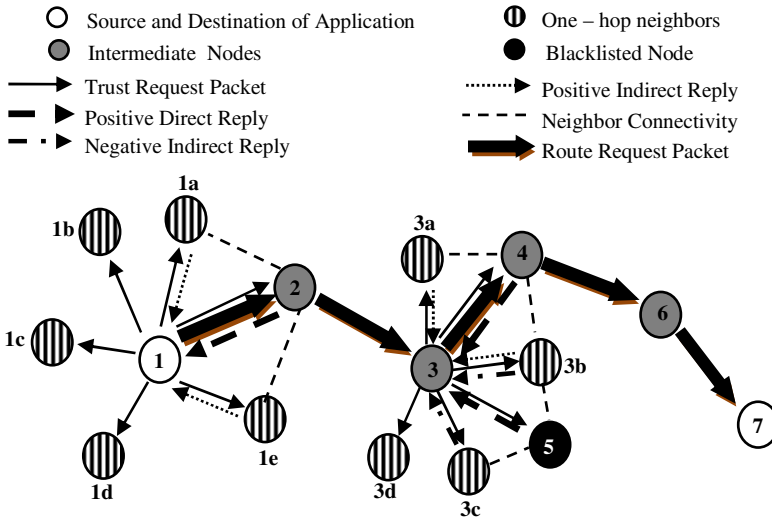


Fig. 1. The Route Request Mechanism

Fig.1 illustrates how the proposed intrusion detection algorithm works during the route setup process. Route Request packets are initiated from the source of an application. The *source node '1'* sends a Trust Request Packet to all its one-hop neighbors – *1a, 1b, 1c, 1d, 1e*, and *2*. Every node replies with Direct Valuation of itself and Indirect References about one-hop neighbors which are common to itself and the source of the Trust Request packet. Here the source node '*1*' receives replies from *1a, 1e*, and *2*. It is obvious that intruders will speak highly of themselves. Also, attackers can provide incorrect trust information about nodes in their efforts to establish routes through themselves. Thus, the source of the Trust Request packets does not believe the responses coming from its one-hop neighbors blindly. Since every node maintains trust information about its one-hop neighbors, the source associates a credibility factor with the replies coming from its neighbors. After trust evaluation, *node 2* is found to be the best node between the source and the destination.

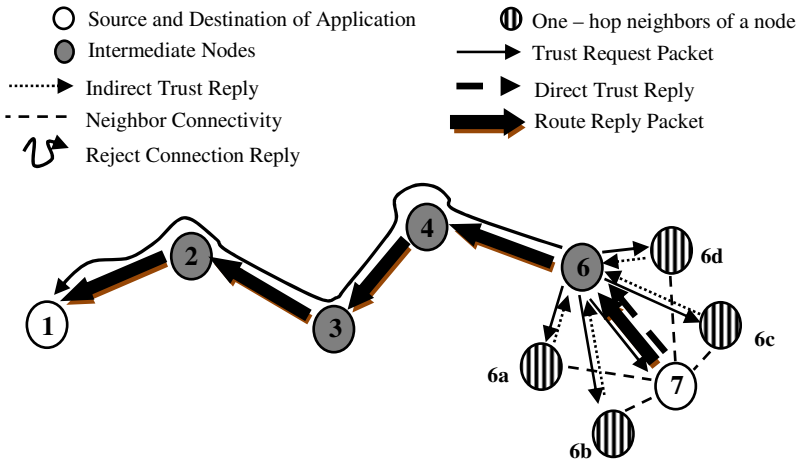


Fig. 2. The Route Reply Mechanism

This procedure is repeated at every node. The figure illustrates another scenario. When the Route Request packet comes to *node 3*, the same procedure is repeated as above. The one-hop neighbors of *node 3* – *3a, 3b, 3c, 3d, 4*, and *5* – reply to the Trust Request coming from *node 3*. *Node 3* gets positive replies about *node 4* but negative replies about *node 5*. *Node 3* associates the credibility of these replies coming from its one-hop neighbors. It evaluates *node 4* to be the most trustworthy and *node 5* as an intruder. Thus, the Route Request is forwarded in the direction of *node 4*. This procedure gets repeated until the Route Request reaches the destination node.

Fig.2 illustrates the procedure when the Route Request reaches the destination. When the Route Request reaches *node 6*, it sends Trust Request packets to all its neighbors – *6a, 6b, 6c, 6d*, and *7* – including the destination. The destination replies with a Route Reply and also mentions the number of its one-hop neighbors in the Route Reply message. All those one-hop neighbors of *node 6* that are also neighbors

of the destination return their trust information about the destination. *Node 6* evaluates the trust of the destination and decides whether to forward the Route Reply to the Source or to return a Connection Abort message.

3.2 Intrusion Detection and Rerouting

Intrusion detection also becomes essential as a part of maintenance. Once connection has been established between the source and the destination, application traffic starts flowing between the two. An intruder may start behaving maliciously or selfishly at some random time instant. Trust evaluation begins at the source. The source evaluates the trust of its one – hop neighbor which is on the route to the destination. Once trust is evaluated for the one – hop neighbor on the source – destination route, the same procedure is repeated for the next node on the route. This continues till the trust value of the destination is evaluated.

If during this process, a node detects its peer on the source – destination route to be behaving *selfishly* or *maliciously*, then the rerouting mechanism is initiated. Figure 3 best illustrates this mechanism. Suppose the existing route for application traffic is through $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 7$. During intrusion detection, *node 3* finds that *node 4* has been behaving in a malignant manner. *Node 3* discards the existing route and tries to reroute traffic to the destination bypassing *node 4*. It finds *node 5* as trusted and reestablishes connectivity with *node 6* via *node 5*. Thus, the newly established route for application traffic becomes $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7$.

If intrusion detection for maintenance finds that the destination has become malicious then the application is closed.

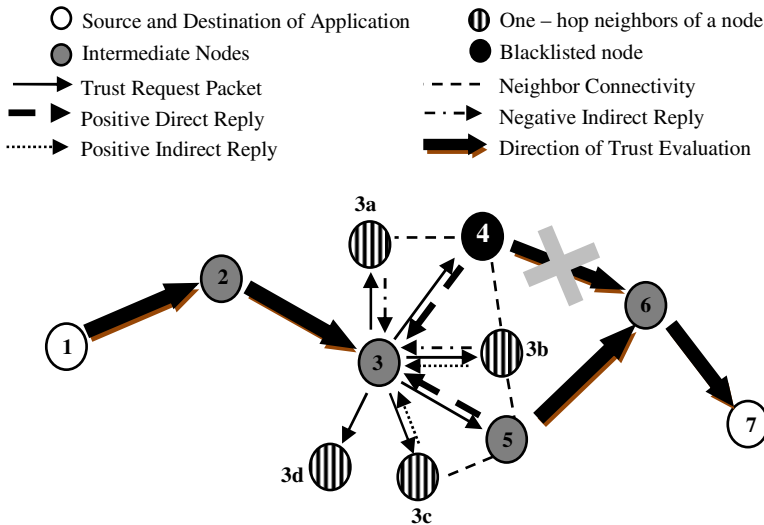


Fig. 3. The Rerouting Mechanism

3.3 The Trust Model

The entire process of routing and intrusion detection is based on the trust evaluated by a node for its one – hop neighbors. The trust model has two main underlying concepts – *Direct Valuation* and *Indirect Reference*.

Direct Valuation is a measure of how the node evaluates the Trust of its one-hop neighbors. Every node monitors the packet forwarding behavior of its one-hop neighbors. A benign node should forward all the packets that it receives from its previous hop neighbor. Thus, packet arrival rate (PAR) and packet delivery rate (PDR) play a decisive role in deciding the behavior of a node. For normal node behavior PAR and PDR tend to be equal. In other words, $PAR - PDR$ tend to zero. Keeping in mind wireless network constraints like mobility and link failure, the normal behavior of a node is classified when the difference ($PAR - PDR$) lies within a given threshold.

In the selective forwarding attack scenario, a node drops packet occasionally. At other times, it behaves like a normal node. Normal behavior of a node is positively rewarded by increasing that node's trust value. Thus, occasional malicious behavior becomes even more difficult to detect. The proposed IDRS addresses this issue using two separate measures for *Risk* and *Reputation*. Risk is a measure of the node's behavior in the last time slice since the last time the intrusion detection algorithm was run. Reputation is the measure of the long term behavior of a node. Classifying Direct Valuation into Risk and Reputation helps in identifying the most recent behavior of a node in contrast to its long term behavior on the time scale.

Since Direct Valuation depends on the PAR and PDR information coming from one-hop neighbors, attackers may easily tamper this information. Thus, trust of a node is not updated solely on the basis of Direct Valuation. One also needs to consider the reputation of the target node to all its one-hop neighbors. Thus, Indirect References are considered from all those one-hop neighbors that are common to both the evaluation node and the target node. Thus, Indirect Reference of the evaluation node consists of the Reputation information coming from all those one-hop neighbors which are also neighbors of the target node.

Every node maintains a Packet Receive (PR) and Packet Send (PS) counter. After every time slice, these counter values are sent to the node's one-hop neighbors. The neighbors keep a track of the Reputation of the node by summing the $PR - PS$ values coming at the end of each time slice. Also the value of the $PR - PS$ counters in the last time slice measures the Risk. When a node receives a IDS Request packet from an evaluation node, it sends its $PR - PS$ counter values, and Reputation and Trust information. The $PR - PS$ values of the target node is used to evaluate the Risk. These values are summed up with the existing Reputation data and Reputation information of other one – hop neighbors become the evaluation node's Indirect Reference information. These three measures are combined to evaluate the reward for the target node's behavior in the last time slice as follows:

$$\text{Reward} = (W_1 \times \text{Risk}) + (W_2 \times \text{Reputation}) + (W_3 \times \text{Indirect Reference}) \quad (1)$$

The above formula is used to generate negative rewards by assigning negative weights to W_1 , W_2 , and W_3 . Also, these weights are normalized so that $W_1 + W_2 + W_3$

= -1. This formula will be used only when the target node has behaved maliciously in the last time slice, i.e., $\text{abs}(\text{PAR} - \text{PDR}) > \text{Threshold}$. For normal behavior, the Reward generated is positive as follows:

$$\text{Reward} = (\text{PAR} + \text{PDR}) / 2 * W_4 \quad (2)$$

W_4 is chosen so that Positive reward is not very large. Nodes must not be able to increase their trust values rapidly by behaving normally in some time slices. Once the reward for a node is appropriately calculated, the trust value of the node is updated as follows:

$$\text{Trust}(t) = \text{Trust}(t-1) + \text{Reward} \quad (3)$$

Based on the above formula, the trust of a node may increase gradually or decrease rapidly. Once the trust value of a node is updated, it is checked whether the trust value falls below a certain threshold. If so, then the node is classified as an attacker.

3.4 Algorithm for Intrusion Detection during Route Setup

1. The source initiates route discovery by generating RREQ packets.
2. Whenever a node receives a RREQ packet it forwards the packet to the most trusted one-hop neighbor on the route to the destination.
3. The node broadcasts Trust Request packets to its one-hop neighbors.
4. All neighbors reply with packet forwarding information about itself and Trust information about their one-hop neighbors.
5. The source of the Trust Request packet evaluates the trust of all its one-hop neighbors.
6. The most trusted neighbor is forwarded the RREQ packet.
7. If a node finds the destination to be its neighbor, it forwards the RREQ packet to the destination.
8. Trust value of the destination is evaluated by the pre-destination node.
9. The destination responds with an RREP packet. Depending on the trust evaluated of the destination, the pre-destination node either forwards the RREP packet or returns a "Cancel Application" message towards the source.

3.5 Algorithm for Intrusion Detection as Part of Maintenance

1. After time slice expires the source initiates the Intrusion Detection Algorithm.
2. Source sends Trust Request Packet (TRP) to its 1- hop neighbors.
3. The 1 – hop neighbor on the route to the destination returns its Packet Forwarding information. This is the Direct Valuation data.
4. Those 1 – hop neighbors which are not on the route to the destination, check if the "target node" is their 1- hop neighbor.
5. If so, they return trust information about the target node to the source of the TRP. This is the Indirect Reference.
6. The sender of the TRP receives Direct and Indirect Information.

7. Reputation is the trust value of the target node currently available at the source of the TRP. Risk is the Packet Forwarding information returned by the target node for the last time slice.
8. Indirect recommendations coming from other 1 – hop neighbors are accumulated, averaged and combined with results from the previous step.
9. If the target node is found to be an intruder, then a WARNING message is sent to the source of the TRP that the route is no longer safe.
10. Whenever an intermediate node receives such a message it reestablishes a new route from itself to the destination.

4 Simulation Results

The proposed IDS has been successfully implemented using the standard Network Simulator - QualNet. In this simulation, some nodes have been arbitrarily initialized with higher trust values compared to other nodes. The proposed mechanism successfully sets up routes through the highly trusted nodes. Both Greyholes and DoS agents have been implemented as having high initial trust values. This is practical as attackers do try to attain high trust among their peers before launching an attack. The trust value of course changes dynamically during simulation. The data points collected reflect the sensitivity of TIDS compared to HIDS under similar conditions.

4.1 Simulator Parameter Settings

In order to compare the performance of the proposed solution in terms of Intrusion Detection, HIDS [6], another recent trust based IDS, has also been simulated under the same environment settings. The proposed TIDS solution is compared with HIDS to compare different attacks like *Greyhole*, and *Denial of Service (DoS)*. Table 1 describes the parameters with which we have simulated the proposed TIDS. Trust value nodes vary from 0 – 16. Trust value of 6 is the threshold value below which a node is detected as an intruder. All normal nodes are initialized with a threshold value of 6. Certain nodes can have higher trust. The .config file has been suitably modified to assign a trust value of 10 to these highly trusted nodes.

Table 1. Simulator parameter settings

Parameter	Value
Terrain area	1500X1500 m^2
Simulation time	200 sec
Mac Layer protocol	DCF of IEEE 802.11b standard
Network Layer protocol	AODV routing protocol
Traffic Model	CBR
Number of CBR applications	10% of total nodes
Highly Trusted Nodes	Randomly selected
IDS time slice	10 sec

4.2 Simulation Results

The following data were collected based on the above simulator settings. Four sets of data were collected. The average of this is taken for comparative analysis of performance against HIDS. Data is taken with respect to the number of iterations required for intrusion detection.

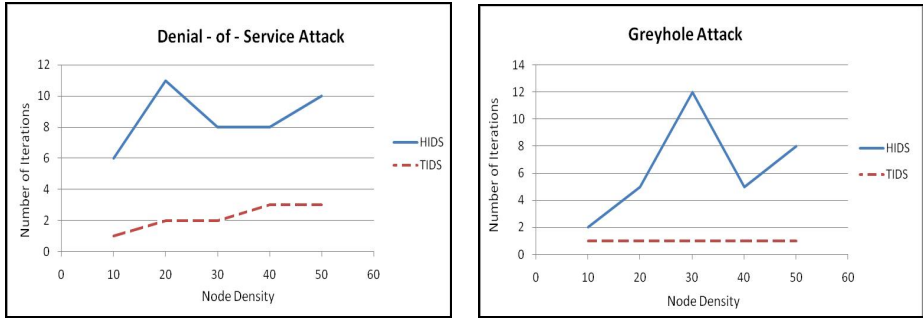


Fig. 3. Performance of TIDS and HIDS with variation in Node Density

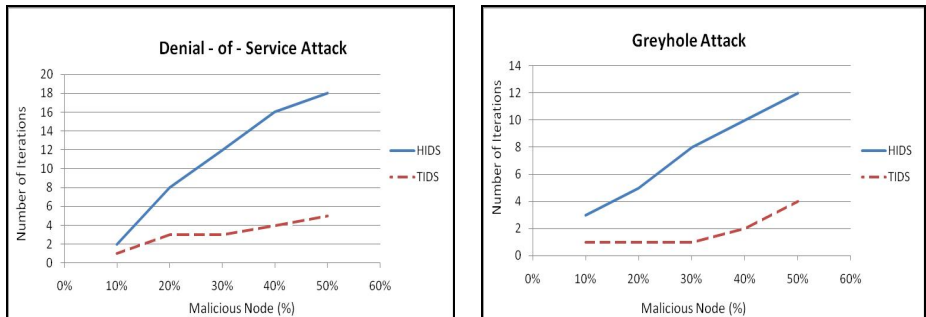


Fig. 4. Performance of TIDS and HIDS with variation in % of Malicious Nodes

The next set of data were taken with a fixed number of nodes ($=40$) and varying the percentage of malicious nodes. Both HIDS and TIDS performed reasonably well in terms of false negatives. None of the algorithms generated any false positives.

All results reflected the sensitivity of the newly proposed Trust model over the Honesty – based scheme proposed in HIDS. These results clearly indicate that TIDS is much more efficient in detecting malicious behavior.

5 Conclusion

In this paper a new trust based IDS has been proposed and evaluated against similar collaborative trust based IDS of recent past. In fact, the proposed TIDS not only detects intrusion, but also proactively responds in finding trusted routes based on this

detection. Thus, TIDS exceeds beyond just being an IDS and works more as an Intrusion Response System. The proposed methodology may be extended for Wireless Mesh Networks and Sensor Networks. QoS management may be done efficiently using the proposed Trust model. Also, Trust based routing can be deployed in wireless networks to reduce the chances of possible intrusions in the first place. There is scope for implementation in the domain of MANETs where the proposed algorithm can be simulated by varying mobility of the nodes.

Acknowledgement. We would like to thank the Advanced Technology Cell, DRDO Cell for approving our work as a Defense – related project. The contingency and Fellowship provided by the Advanced Technology Cell has played a vital role in the completion and publication of this work.

References

1. Liang, Z., Shi, W.: PET: A Personalized Trust Model with Reputation and Risk Evaluation for P2P Resource Sharing. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 2005) - Track 7, vol. 7, p. 201b (2005)
2. Cho, J.H., Swami, A., Chen, I.R.: Modeling and Analysis of Trust Management with Trust Chain Optimization in Mobile Ad Hoc Networks. In: Proc. of the 2009 International Conference on Computational Science and Engineering, vol. 02, pp. 641–650 (2009)
3. Xiong, L., Liu, L.: Building Trust in Decentralized Peer-to-Peer Electronic Communities. In: Proc. 5th Int'l Conf. Electronic Commerce Research, ICECR-5 (2002)
4. Wang, W.G., Mokhta, M., Linda, M.: C-index: trust depth, trust breadth, and a collective trust measurement. In: Proceedings of the Hypertext 2008 Workshop on Collaboration and Collective Intelligence, pp. 13–16 (2008)
5. Luo, J., Liu, H.X., Fan, M.Y.: A trust model based on fuzzy recommendation for mobile ad-hoc networks. *Computer Networks* 53(14), 2396–2407
6. Sil, P., Chaki, R., Chaki, N.: HIDS: Honesty-rate based collaborative Intrusion Detection System for Mobile Ad-Hoc Networks. In: Proc. of 7th IEEE International Conference on Computer Information Systems and Industrial Management Applications, CISIM (2008)

Intruder Data Classification Using GM-SOM

Petr Gajdoš and Pavel Moravec

Department of Computer Science, FEECS, VŠB – Technical University of Ostrava,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
{petr.gajdos,pavel.moravec}@vsb.cz

Abstract. This paper uses a simple modification of classic Kohonen network (SOM), which allows parallel processing of input data vectors or partitioning the problem in case of insufficient resources (memory, disc space, etc.) to process all input vectors at once. The algorithm has been implemented to meet a specification of modern multicore graphics processors to achieve massive parallelism. The algorithm pre-selects potential centroids of data clusters and uses them as weight vectors in the final SOM network. In this paper, the algorithm is used on a well-known KDD Cup 1999 intruders dataset.

Keywords: SOM, Kohonen Network, parallel computation, KDD Cup 1999 Data Set.

1 Introduction

With the massive boom of GPU-based calculations, massive parallelism, memory considerations, simplicity of algorithms and CPU-GPU interaction have yet again to play an important role. In this paper, we present a simple modification of classic Kohonen's self-organizing maps (SOM), which allows us to dynamically scale the computation to fully utilize the GPU-based approach.

There were some attempts to introduce parallelism in Kohonen networks [1][2][3][4][5], however we needed an approach which is simple and easy to implement. Moreover, it should work both with and without the bulk-loading algorithm [6].

In this paper, we present such approach, which divides the training set into several subsets and calculates the weights in multi-step approach. Calculated weights with nonzero number of hits serve as input vectors of SOM network in the following step. Presently, we use a two-step approach, however more steps could be used if necessary.

The paper is organized as follows: in second chapter we mention classic SOM networks and describe the basic variant we have used. In third chapter we describe our approach and provide the calculation algorithm, in fourth the GPU-based processing. The fifth chapter introduces experimental data we have used and the final chapter before conclusion presents the comparison of results provided by our method with classic SOM calculation.

2 Kohonen Self-organizing Neural Network

In following paragraphs, we will shortly describe the Kohonen self-organizing neural networks (self-organizing maps – SOM). The first self-organizing networks were proposed in the beginning of 70's by Malsburg and his successor Willshaw. SOM was

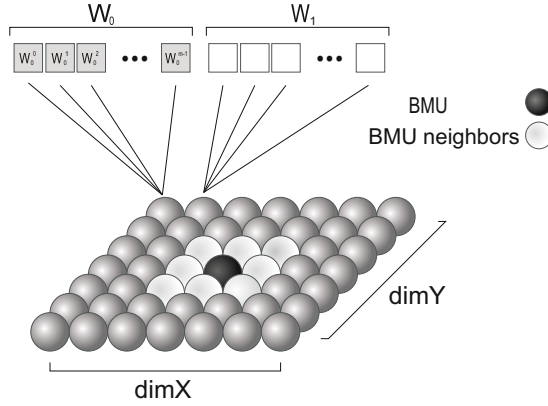


Fig. 1. Kohonen network structure

proposed by Teuvo Kohonen in the early 1980s and has been improved by his team since. The summary of this method can be found in [7].

The self-organizing map is one of the common approaches on how to represent and visualize data and how to map the original dimensionality and structure of the input space onto another – usually lower-dimensional – structure in the output space.

The basic idea of SOM is based on the human brain, which uses internal 2D or 3D representation of information. We can imagine the input data to be transformed to vectors, which are recorded in neural network. Most neurons in cortex are organized in 2D. Only the adjacent neurons are interconnected.

Besides of the input layer is in SOM only the output (competitive) layer. The number of inputs is equal to the dimension of input space. Every input is connected with each neuron in the grid, which is also an output (each neuron in grid is a component in output vector). With growing number of output neurons, the quality coverage of input space grows, but so does computation time.

SOM can be used as a classification or clustering tool that can find clusters of input data which are more closer to each other.

All experiments and examples in this paper respect following specification of the SOM (see also the Figure 1):

- The SOM is initialized as a network of fixed topology. The variables $dimX$ and $dimY$ are dimensions of such 2-dimensional topology.
- V^m represents an m -dimensional input vector.
- W^m represents an m -dimensional weight vector.
- The number of neurons is defined as $N = dimX * dimY$ and every neuron $n \in \{0, N - 1\}$ has its weight vector W_n^m .
- The neighborhood radius r is initialized to the value $min(dimX, dimY)/2$ and will be systematically reduced to a unit distance.
- All weights vectors are updated after particular input vector is processed.
- The number of epochs e is known at the beginning.

The Kohonen algorithm is defined as follows:

1. **Network initialization**

All weights are preset to a random or pre-calculated value. The learning factor η , $0 < \eta < 1$, which determines the speed of weight adaptation is set to a value slightly less than 1 and monotonically decreases to zero during learning process. So the weight adaptation is fastest in the beginning, being quite slow in the end.

2. **Learning of input vector**

Introduce k training input vectors V_1, V_2, \dots, V_k , which are introduced in random order.

3. **Distance calculation**

A neighborhood is defined around each neuron whose weights are going to change, if the neuron is selected in competition. Size, shape and the degree of influence of the neighborhood are parameters of the network and the last two decrease during the learning algorithm.

4. **Choice of closest neuron**

We select the closest neuron for introduced input.

5. **Weight adjustment**

The weights of closest neuron and its neighborhood will be adapted as follows:

$$W_{ij}(t+1) = W_{ij}(t) + \eta(t)h(v, t)(V_i - W_{ij}(t)),$$

where $i = 1, 2, \dots, \dim X$ a $j = 1, 2, \dots, \dim Y$ and the radius r of neuron's local neighborhood is determined by adaptation function $h(v)$.

6. **Go back to point 2 until the number of epochs e is reached.**

To obtain the best organization of neurons to clusters, a big neighborhood and a big influence of introduced input are chosen in the beginning. Then the primary clusters arise and the neighborhood and learning factor are reduced. Also the $\eta \rightarrow 0$, so the changes become less significant with each iteration.

3 GM-SOM Method

The main steps of SOM computation have already been described above. Following text is focused on description of proposed method, that in the end leads to results similar to the classic SOM (See also Figure 2 for illustration of our approach). We named the method Global-Merged SOM, which suggests, that the computation is divided into parts and then merged to obtain the expected result. Following steps describe the whole process of GM-SOM:

1. **Splitting of input set.** The set of input vectors is divided into a given number of parts. The precision of proposed method increases with the number of parts, however, it has own disadvantages related to larger set of vectors in the final phase of computation process. Thus the number of parts will be usually determined from the number of input vectors. Generally, $k \gg N * p$, where k is the number of input vector, N is the number of neurons and p is the number of parts. The mapping of input vectors into individual parts does not affect final result. This will be later demonstrated by the experiments, where all the input vectors were either split sequentially (images) or randomly.

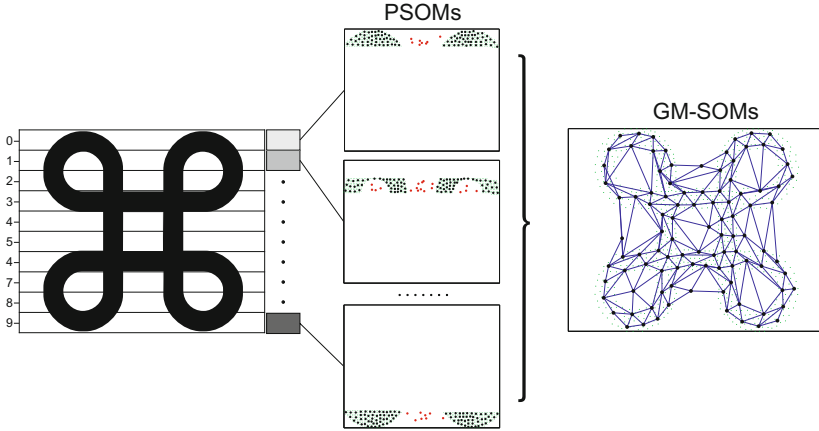


Fig. 2. GM-SOM: An Illustrative schema of the proposed method. All input vectors are divided into ten parts in this case.

2. **In parts computation.** Classic SOM method is applied on every part. For simplicity sake, an acronym *PSOM* will be used from now on to indicate SOM, which is computed on a given part. All PSOMs start with the same setting (the first distribution of weights vectors, number of neurons, etc.) Such division speeds up parallel computation of PSOMs on GPU. Moreover, the number of epochs can be lower than the the number of epochs used for processing of input set by one SOM. This is represented by a factor f , which is going to be set to $\frac{1}{3}$ in our experiments.
3. **Merging of parts.** Weight vectors, that where computed for each part and correspond to neurons with at least one hit, represent input vectors in the final phase of GM-SOM. The unused neurons and their weight vectors have light gray color in Figure 2. A merged SOM with the same setting is computed and output weights vectors make the final result of proposed method.

The main difference between the proposed algorithm and well known batch SOM algorithms is, that individual parts are fully independent on each other and they update different PSOMs. Moreover, different SOM algorithms can be applied on PSOM of a given part, which makes the proposed algorithm more variable. Next advantage can be seen in different settings of PSOMs. Thus more dense neuron network can be used in case of larger input set. The last advantage consists in a possibility of incremental updating of GM-SOM. Any additional set of input vectors will be processed by a new PSOM in a separate part and the final SOM will be re-learned. For interpretation see Figure 3.

4 GPU Computing

Modern graphics hardware plays an important role in the area of parallel computing. Graphics cards have been used to accelerate gaming and 3D graphics applications, but

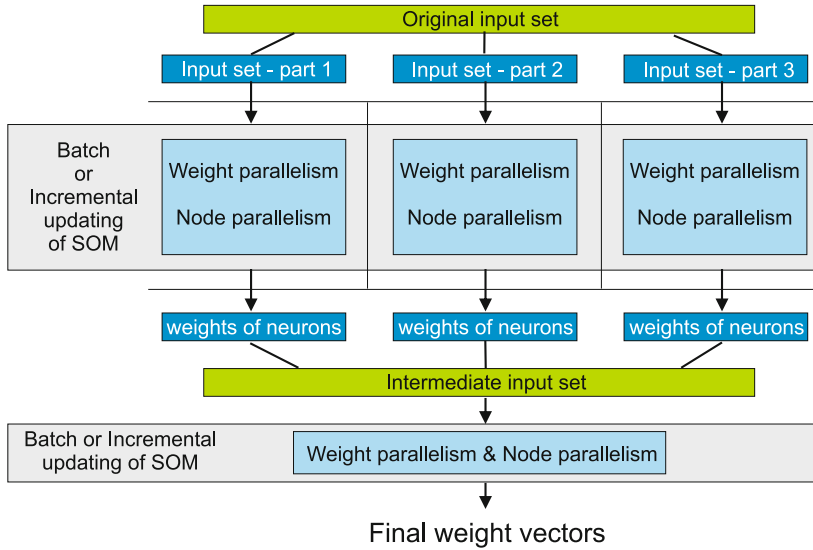


Fig. 3. GM-SOM: Parallelization of the SOM computation by proposed method

recently, they have been used to accelerate computations for relatively remote topics, e.g. remote sensing, environmental monitoring, business forecasting, medical applications or physical simulations etc. Architecture of GPUs (Graphics Processing Unit) is suitable for vector and matrix algebra operations, which leads to a wide use of GPUs in the area of information retrieval, data mining, image processing, data compression, etc. Nowadays, the programmer does not need to be an expert in graphics hardware because of existence of various APIs (Application Programming Interface), which help programmers to implement their software faster. Nevertheless, it will be always necessary to follow basic rules of GPU programming to write a more efficient code.

Four main APIs exist today. The first two are vendor specific, i.e. they were developed by two main GPU producers - AMD/ATI and nVidia. The API developed by AMD/ATI is called ATI Stream and the API developed by nVidia is called nVidia CUDA (Compute Unified Device Architecture). Both APIs are able to provide similar results. The remaining two APIs are universal. The first one was designed by Khronos Group and it is called OpenCL (Open Computing Language) and the second was designed by Microsoft as a part of DirectX and it is called Direct Compute. All APIs provide a general purpose parallel computing architectures that leverages the parallel computation engine in graphics processing units.

The main advantage of GPU is its structure. Standard CPUs (central processing units) contain usually 1-4 complex computational cores, registers and large cache memory. GPUs contain up to several hundreds of simplified execution cores grouped into so-called multiprocessors. Every SIMD (Single Instruction Multiple Data) multiprocessor drives eight arithmetic logic units (ALU) which process the data, thus each ALU of a

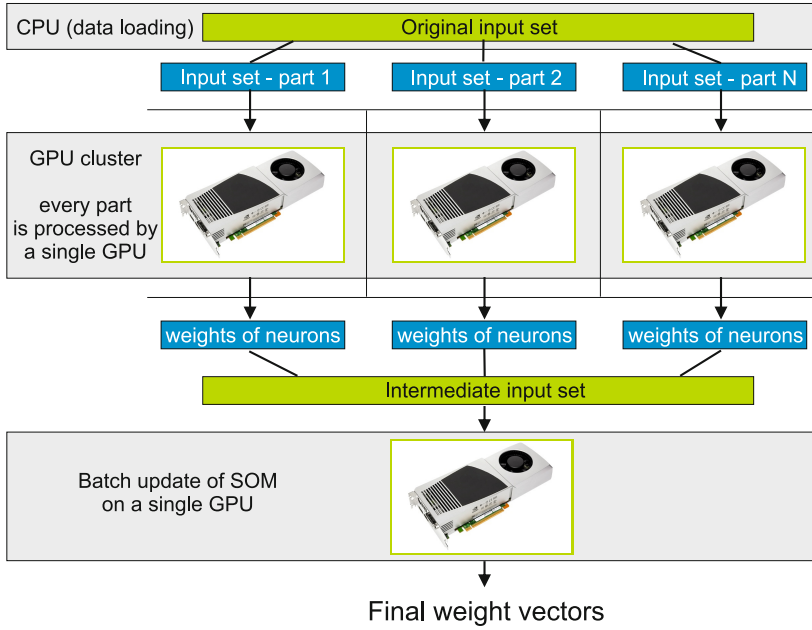


Fig. 4. GM-SOM: Parallelization of the SOM computation by proposed method

multiprocessor executes the same operations on different data, lying in the registers. In contrast to standard CPUs which can reschedule operations (out-of-order execution), the selected GPU is an in-order architecture. This drawback is overcome by using multiple threads as described by Wellein et al. [8]. Current general-purpose CPUs with clock rates of 3 GHz outperform a single ALU of the multiprocessors with its rather slow 1.3 GHz. The huge number of parallel processors on a single chip compensates this drawback.

The GPU computing has been used in many areas. Andreut [9] described CUDA-based computing for two variants of Principal Component Analysis (PCA). The usage of parallel computing improved efficiency of the algorithm more than 12 times in comparison with CPU. Preis et al. [10] applied GPU on methods of fluctuation analysis, which includes determination of scaling behavior of a particular stochastic process and equilibrium autocorrelation function in financial markets. The computation was more than 80 times faster than the previous version running on CPU. Patnaik et al. [11] used GPU in the area of temporal data mining in neuroscience. They analyzed spike train data with the aid of a novel frequent episode discovery algorithm, achieving a more than $430\times$ speedup. The GPU computation has also already been used in intrusion detection systems [12] and for human iris identification (for our experiments on this topic see [13]).

The use of our SOM modification on GPU is illustrated in Figure 4.

5 Data Collection

The KDD Cup 1999 collection [14] represents data indicating several basic types of attack on computer networks. Significant features have been recorded and the type of attack has been evaluated. The data is based on the 1998 DARPA Intrusion Detection Evaluation Program, which has been prepared and managed by MIT Lincoln Labs. The main task is to build a predictive model (i.e. a classifier) capable of distinguishing between intrusions (attacks), and normal connections.

The 1999 KDD intrusion detection contest dataset used a specific format of features, extracted from original raw data, which corresponded to 7 weeks of traffic and had approximately 4 GiB of compressed data streams, which were processed into about five million connection records.

Each of the records consists of approximately 100 bytes and the result is labeled based on the actual attack type (or normal, in case it is part of normal traffic) which falls into one of the four basic categories:

- *DOS* – denial-of-service – the machine is flooded with requests and the exhaustion of resources is attempted. Nowadays, it is mostly used in the distributed form.
- *R2L* – remote to local – unauthorized access from a remote machine, e.g. trying to guess passwords or SSH keys.
- *U2R* – user to root – unauthorized user rights elevation, typically through some exploit (e.g. using buffer overflow techniques) to gain superuser (root) privileges.
- *probing* – port scans, active network surveillance and other probing.

There are 22 defined attack types, which were typical in 1998 and the normal category in the training data and some additional attack types present only in the testing data, which the classification framework should detect. The records contain both the categorical, Boolean and numerical attributes, which makes the classification harder.

6 Experiments

The Table 1 is a summary table for seven performed experiments. The whole training data collection was divided into several parts for individual experiments. In each experiment, every part represented an input data set for SOM. Then all partial results (weights of neurons) were merged together and became a new input set for final computation of SOM. Thus the total time consists of particular computation times of SOMs plus required merging time. Sufficient number of graphics processor units (GPU) are required for such computation. Usually, more graphics contexts can be created on a single GPU, however, they share all resources (memory, multiprocessors, etc.). If the number of GPUs is less than the number of parts, and just a single graphics context per GPU can be created (because of memory requirements, etc.), more parts must be computed in series. Then the computation times of all such parts must be summed. Therefore, the column “Total time” of the Table 1 shows ideal computation times in case of sufficient computation power.

The number of parts affects the precision of SOM classification. The higher number of parts the smaller precision. It depends on process of division of input data set as

Table 1. Computation times with respect to subdividing of input data set

exp. id	parts [#]	size of a part [#]	Comp. time per part [s]	Merging time [s]	Total time [s]	Precision [%]
1	1	1469529	1249	0	1249	94,29
2	5	293906	249	4	453	94,13
3	10	146953	124	9	133	93,57
4	20	73477	62	18	80	90,72
5	30	48985	43	27	70	88,47
6	40	36739	32	36	68	87,91
7	50	29391	26	45	71	87,34

well. Every part consists of random set of input vectors in our experiments. Every two parts represent disjoint sets of input vectors. Finally, every vector of the original data set (non-divided) belongs to some part. The ratio between the number of parts and final precision of classification method will be the subject of further research.

7 Conclusion

The need of parallel computation of SOM drove us to a new method, that has been also utilized in this paper. Although it has some common features with well known SOM batch or hierarchical algorithms, it is not one of them, as it has its unique properties. The results presented in previous section show that whilst the classic SOM is much faster because of the GPU-based computation, the use of PSOMs and their computation by separate nodes further improves the computation time.

Firstly, the proposed algorithm can utilize the power of batch processing in all inner parts (PSOMs). Secondly, all PSOMs can have different number of neurons in their networks, which could be found in hierarchical algorithms. Lastly, our method excludes neurons, which do not cover any input vectors in the intermediate phase of GM-SOM.

All experiments suggest, that the results are very close to results provided by classic SOM algorithm. Also, since we have used the KDD Cup 1999 collection, the data is comparable with other experiments done on this collection.

Acknowledgment. This work was supported by the Bio-Inspired Methods: research, development and knowledge transfer project, reg. no. CZ.1.07/2.3.00/20.0073 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic.

References

1. Mann, R., Haykin, S.: A parallel implementation of Kohonen's feature maps on the warp systolic computer. In: Proc. IJCNN-90-WASH-DC, Int. Joint Conf. on Neural Networks, vol. II, pp. 84–87. Lawrence Erlbaum, Hillsdale (1990)

2. Openshaw, S., Turton, I.: A parallel Kohonen algorithm for the classification of large spatial datasets. *Computers & Geosciences* 22(9), 1019–1026 (1996)
3. Nordström, T.: Designing parallel computers for self organizing maps. In: *Forth Swedish Workshop on Computer System Architecture* (1992)
4. Valova, I., Szer, D., Gueorguieva, N., Buer, A.: A parallel growing architecture for self-organizing maps with unsupervised learning. *Neurocomputing* 68, 177–195 (2005)
5. Wei-gang, L.: A study of parallel self-organizing map. In: *Proceedings of the International Joint Conference on Neural Networks* (1999)
6. Fort, J., Letremy, P., Cottrel, M.: Advantages and drawbacks of the batch Kohonen algorithm. In: *Proceedings of the 10th European-Symposium on Artificial Neural Networks, ESANN 2002*, pp. 223–230 (2002)
7. Kohonen, T.: *Self-Organizing Maps*, 2nd (extended) edn. Springer, Berlin (1997)
8. Hager, G., Zeiser, T., Wellein, G.: Data access optimizations for highly threaded multi-core CPUs with multiple memory controllers. In: *IPDPS*, pp. 1–7. IEEE (2008)
9. Andrecut, M.: Parallel GPU implementation of iterative PCA algorithms. *Journal of Computational Biology* 16(11), 1593–1599 (2009)
10. Preis, T., Virnau, P., Paul, W., Schneider, J.J.: Accelerated fluctuation analysis by graphic cards and complex pattern formation in financial markets. *New Journal of Physics* 11(9), 093024 (21p.) (2009)
11. Patnaik, D., Ponce, S.P., Cao, Y., Ramakrishnan, N.: Accelerator-oriented algorithm transformation for temporal data mining. *CoRR abs/0905.2203* (2009)
12. Platos, J., Kromer, P., Snasel, V., Abraham, A.: Scaling IDS construction based on non-negative matrix factorization using GPU computing. In: *2010 Sixth International Conference on Information Assurance and Security (IAS)*, pp. 86–91 (August 2010)
13. Gajdos, P., Platos, J., Moravec, P.: Iris recognition on GPU with the usage of non-negative matrix factorization. In: *2010 10th International Conference on Intelligent Systems Design and Applications (ISDA)*, November 29–December 1, pp. 894–899 (2010)
14. Stolfo, S., Fan, W., Lee, W., Prodromidis, A., Chan, P.: Cost-based modeling for fraud and intrusion detection: results from the jam project. In: *Proceedings of the DARPA Information Survivability Conference and Exposition, DISCEX 2000*, vol. 2, pp. 130–144 (2000)

Method for Identification of Suitable Persons in Collaborators' Networks

Pavla Dráždilová, Alisa Babskova, Jan Martinovič,
Kateřina Slaninová, and Štěpán Minks

VŠB - Technical University of Ostrava,
Faculty of Electrical Engineering and Computer Science,
17. Listopadu 15/2172, 708 33 Ostrava, Czech Republic
{pavla.drazdilova,jan.martinovic,alisa.babskova.st,
katerina.slaninova,min111}@vsb.cz

Abstract. Finding and recommendation of suitable persons based on their characteristics in social or collaboration networks is still a big challenge. The purpose of this paper is to discover and recommend suitable persons or whole community within a developers' network. The experiments were realized on the data collection of specialized web portal used for collaboration of developers - Codeplex.com. Users registered on this portal can participate in multiple projects, discussions, adding and sharing source codes or documentations, issue a release, etc. In the paper we deal with strength extraction between the developers based on their association with selected terms. We have used the approach for extraction of initial metadata, and we have used modified Jaccard coefficient for description of the strength of relations between developers. Proposed method is usable for creation of derived collaborators' subnetwork, where as input is used the set of words, which will describe the area or sphere, wherein we want to find or recommend suitable community and the words specify relation between the developers in the network. Obtained subnetwork describe a structure of developers' collaboration on projects, described by selected term.

1 Introduction

Recently the concept of social networks and online communities is becoming still more and more popular. As a result, the number of their users significantly increasing. Reasons for communication between people and creation of social networks in our time are various: study, hobby, work, games and programming is not the exception.

OSS (Open Source Software) is a example of a dynamic network, as well as a prototype of complex networks emerging on the Internet. By working through the Internet, interactions between developers can be considered as relations in the synthetic network of collaborators. These relations arise when the developers join the project and begin to communicate with others. OSS network consists of two entities - developers and projects. An examples of such OSS social network established on the basis of interaction between the participants is CodePlex.

Many programmers on the Internet are looking for interesting ideas, or assistance when implementing their own solutions. Online collaboration is no longer a novelty in

our times and it is run by people all over the world. However, searching for suitable and capable people who could implement a particular idea at reasonable deadlines and high quality is an eternal problem.

In this paper we try to determine the strength of relationship or similarity between CodePlex developers in the context of projects they work on. To determine the context, we used project key words, which in the case of the CodePlex are extracted from project descriptions. We would find some developers or some community, which is specified by key words, for a recommendation.

Some related work dealing with the recommendation in the social network. In the article [1] authors studies people recommendations designed to help users find known, offline contacts and discover new friends on social networking sites. Other approach is in the article [4], where authors examine the dynamics of social network structures in Open Source Software teams but data were extracted monthly from the bug tracking system in order to achieve a longitudinal view of the interaction pattern of each project.

2 CODEPLEX

CodePlex is Microsoft's open source project hosting web site. You can use CodePlex to find open source software or create new projects to share with the world. Codeplex.com has 11 years old, it is ranked 2,107 in the world, a low rank means that this website gets lots of visitors. Its primary traffic from United States and is ranked 3,175 in United States. It has 104 subdomains with traffic. It has 136,500 visitors per day, and has 436,800 pageviews per day. CodePlex is mainly used by developers for collaboration on projects, sharing source codes, communication and software development. Generally, registered users can participate in multiple projects, discussions, adding the source code and documentation, issue a release, etc. Some of the users have defined a specific role within the project for which they work. Each user has his own page, where he can share information about himself, his projects on which he currently works, and the most recent activities. The CodePlex projects themselves can be considered as a very interesting source of information. In addition to the list of users and roles, CodePlex enables register keywords, add description of the project, the number of visits, status, date of creation, url and other information about the project. All activities are carried out on CodePlex by a particular user within a specific project.

Database which was created as a result of data obtained from CodePlex.com, consists of 6 main tables: User, Project, Discussions, RecentActivity, Membership and SourceCode (see Table I).

In CodePlex, we can see two types of entities: users and projects. Both are represented by tables that contain specific characteristics. The table User contains informations about users such as login, personalStatement, createdOn, lastVisit and url of user page. The table Project contains some characteristics of project in Codeplex: tags, date od created on, status, license, pageViews, count of visits, description and url of project page.

The undirect connection between the user and the project is implemented through activities within the scope of the project. These activities are in the database CodePlex

Table 1. The CodePlex database tables

Table	Number of lines
User	96251
Project	21184
Discussions	397329
RecentActivity	72285
Membership	126759
SourceCode	610917

Table 2. The CodePlex activities

Activity	Meaning
SourceCode	records about added projects
Discussion	discussions about the project and the responses of individual users
RecentActivity	check-ins, task records, add Wiki information, notes about Release version etc
Membership	able to trace the users' participation in the projects and their assigned role

divided into different types: SourceCode, Discussion, RecentActivity and Membership (see Table 2).

We can represent CodePlex as a bipartite graph of users and projects, where the edge between the user and the project is a user's activity in a project.

If we look at the data that we have in the Table User, we are not able to define the user's profile. It consists of the field of interest, what he deals with, the programming language he uses and at what level. PersonalStatement attribute is used to describe the user, but from the total set of our users downloaded, there was not a single one, who would fill it up. On the other hand, the project has enough information defined – which fields are concerned, how long it lasted, whether it is completed, which technology it is used, etc.

The main attribute, carrying the largest set of information, is the project Description – the description of the project itself.

Using activities such as user links to the projects, we are able to determine with some probability an area of specialization and a work of each user. For example, if a user is working on three projects written in .NET and one in Java, we could include him in .NET programmers with high probability, and less likely recommend him as a Java programmer.

In other words, terms or description of the project may not only help us to provide more information about projects, but also to determine the user's area of interests or abilities. As a result, the way we are able to compare user attributes determines the similarity to other network participants.

3 Collaborators Network and Projects Network

Whenever we think about collaboration between two persons, we not only look at the relationship itself, but also at the context. It is clear that depending on context, the

strength of relationship changes. Therefore, we divide collaboration into two main parts *Developers' Relationship* and *Developers' Context*. We consider the relation between developers and the term describes the context between developers.

Developers have additional attributes. Usually it could be publications, teams, organizations, projects, etc. We called it attribute domain, in our case DCP_D be a set of projects in Codeplex, then CP_D are attributes for all D_i developers, where objects is one developer's attributes described as $CP_{D_i} \subseteq DCP_D$.

3.1 Developers' and Context Relationship

We describe a developers' relationship as commutative operation on cartesian product of developer's attribute $X \times X$, where output is mapped to the set of real numbers \mathbb{R} .

We use Jaccard coefficient ([2]) for evaluation of developers relations using their attributes.

$$AttributeScore(CP_{D_i}, CP_{D_j}) = \frac{|CP_{D_i} \cap CP_{D_j}|}{|CP_{D_i} \cup CP_{D_j}|} \quad (1)$$

As we discussed above, every developer has it's attributes. Moreover, each project has a description text. If we use lexical analysis on this text, we can define a term set for every developer as T_{D_i} and this term set contains all terms of projects, which developer D_i participated. The extracted text is proceed to methods, which remove words that do not carry any important information. The main issue of this paper is not to describe this kind of methods. More could be found in [6,3].

Term set T consists of all developers term sets $\{T_{D_0}, T_{D_1}, \dots, T_{D_n}\} = T$, when the domain for terms T could be obtained as union of all terms extracted for each person $D_T = T_{D_0} \cup T_{D_1} \cup \dots \cup T_{D_n}$.

The whole process of obtaining term sets is described in [5], so we just reminding $(t_k \text{ in } T_{D_i})$ stands for the number of terms t_k by T_{D_i} and $(t_k \text{ in } T)$ stands for the number of terms t_k in descriptions of all projects by T .

We can evaluate association between the selected term $t_k \in D_T$ and a developer $D_i \in D$:

$$R(T_{D_i}, t_k) = \frac{(t_k \text{ in } T_{D_i})}{(t_k \text{ in } T) + |T_{D_i}| - (t_k \text{ in } T_{D_i})} \quad (2)$$

We normalize $R(T_{D_i}, t_k)$ such that $R_{Norm}(T_{D_i}, t_k) \in [0, 1]$:

$$R_{Norm}(T_{D_i}, t_k) = \frac{R(T_{D_i}, t_k)}{\max(R(T_{D_i}, t_1), \dots, R(T_{D_i}, t_{|T_{D_i}|}))} \quad (3)$$

Evaluation of the whole relationship context of two persons D_i and D_j has two steps. First, we compute association between D_i and select term t_k , and between the second developer D_j and t_k separately. Afterwards, because each part is already evaluated by real number, we combine both results in the same way; we can combine the whole result in equation one. In CodePlex we see the description text for the developer as the

all description of all projects he is working on, joined together. We obtain equation for the $ContextScore$:

$$ContextScore(T_{D_i}, T_{D_j}, t_k) = R_{Norm}(T_{D_i}, t_k) R_{Norm}(T_{D_j}, t_k) \quad (4)$$

3.2 Collaboration – Whole Score

The last step is to define Score, which consists of $AttributeScore$ and $ContextScore$:

$$Score(CP_{D_i}, CP_{D_j}, T_{D_i}, T_{D_j}, t_k) = AttributeScore(CP_{D_i}, CP_{D_j}) ContextScore(T_{D_i}, T_{D_j}, t_k) \quad (5)$$

This equation evaluates the relation between developers depending on the selected words, which represent the context. So we get a evaluation for the new subnet, which is specified by selected terms.

3.3 Construction of the Collaborators Graph

To describe the network of collaboration, we use standard weighted graph $G_D(V_D, E_D)$, where weighted function is defined as $w_D : E_D(G) \mapsto \mathbb{R}$, when $w_D(e) \geq 0$.

The determination of set D is simple, because objects of vertices set V_D match with objects of set D , so $V_D = D$. However, we can do the same with all the possible pairs from set D to assign a set of edges E_D ; it is better to design the algorithm to each implementation at first, and to reduce the number of useless computations. In addition, we must choose term t_k for function w_D , which reflects the context. Because only the commutative operations are used, we do not need to take into consideration the order of attribute objects in function parameters. Moreover E_D is two-object set, where the order of objects does not matter, so the evaluating is done just once.

When we construct graph based on developers' projects relationship, we use $AttributeScore(CP_{D_i}, CP_{D_j})$ as w_D , where no term is needed, then simply $V_D = D$, which means that every developer is a vertex in the graph. Then, for each developer $D_i \in D$ we find collaborators D_{i_C} and for each collaborator $D_j \in D_{i_C}$ we create two-object set $\{D_i, D_j\}$, which corresponds with an edge in the graph. Equation (5) is then used to evaluate the edge.

The function $Score(CP_{D_i}, CP_{D_j}, T_{D_i}, T_{D_j}, t_k)$ is used for evaluating the edges in the context of the term. The only difference is, that majority of developers has not chosen term in their description text, so the result will be 0 and no edge would exists. Hence, we first determine subset of developers $D_{t_k} \subseteq D$ for those that have a term in their description text, followed by the same steps described in the last paragraph to compute developers' projects relationship. Then, the term t_k is used for computation of the second part in $ContextScore(T_{D_i}, T_{D_j}, t_k)$. Finally, we calculate the whole $Score$ by multiplication of both parts.

3.4 Construction of the Projects Graph

We consider as well as developers, as well as projects. We define *Projects' Relationship* and *Projects' Context*.

We use Jaccard coefficient for evaluation of projects relations using their attributes - $AttributeScore$.

$$AttributeScore(CP_{P_i}, CP_{P_j}) = \frac{|CP_{P_i} \cap CP_{P_j}|}{|CP_{P_i} \cup CP_{P_j}|} \quad (6)$$

We evaluate association between the selected term $t_k \in D_T$ and a project $P_i \in P$:

$$R(T_{P_i}, t_k) = \frac{(t_k \text{ in } T_{P_i})}{(t_k \text{ in } T) + |T_{P_i}| - (t_k \text{ in } T_{P_i})} \quad (7)$$

We compute equation for the $ContextScore$:

$$ContextScore(T_{P_i}, T_{P_j}, t_k) = R_{Norm}(T_{P_i}, t_k) R_{Norm}(T_{P_j}, t_k) \quad (8)$$

The last step is to calculate $Score$, which consists of $AttributeScore$ and $ContextScore$:

$$Score(CP_{P_i}, CP_{P_j}, T_{P_i}, T_{P_j}, t_k) = AttributeScore(CP_{P_i}, CP_{P_j}) ContextScore(T_{P_i}, T_{P_j}, t_k) \quad (9)$$

To describe the network of projects, we use standard weighted graph $G(V_P, E_P)$, where weighted function is defined as $w_P : E_P(G) \mapsto \mathbb{R}$, when $w_P(e) \geq 0$. We consider $V_P = P$ and we use $Score$ for edges evaluation in the new graph $G(V_P, E_P)$.

4 Experiments

For the basic computation of the collaboration, we chose the terms "iphone", "wp7", "android" and apply it to the formula 3.

The results were limited to the collaborators with whose the person has collaborated together on the project at least once. We show centrality value of selected nodes in the Table 3. These centralities characterize the position of vertices in the network.

Table 3. Centralities of developers

User	Degree	Weighted degree	Closeness	Betweenness
raja4567	34	7,69353	1,92	4948,5
modder	4	0,0033	2,879	4143
raouf	7	0,0366	2,879	0

We show in the Table 4 values of $AttributeScore$ for person with nickname modder, in the Table 5 for person with nickname raja4567 and in the Table 6 for person with nickname raouf.

We can immediately notice that even though "modder" and "raouf" do not participate on many projects with "raja4567" (they have one common project), the AttributeScore is 0.01176471 and 0.01204819. For example "shankar00" participate on 2 projects with "raja4567" and the AttributeScore is 0.02469136. User "modder" has only one common project with "shankar00", but AttributeScore is strong, probably because "shankar00" not cooperate with many other persons.

Table 4. Collaborators of "modder"

Number	Collaborators	Projects	Common projects	Attribute _{Score}
1	modder	5	5	1
2	doln	1	1	0,2
3	draculus	1	1	0,2
4	FreQi	1	1	0,2
...				
13	shankar00	3	1	0,1666667
...				
25	raja4567	81	1	0,01176471

Table 5. Collaborators of "raja4567"

Number	Collaborators	Projects	Common projects	Attribute _{Score}
1	raja4567	81	81	1
2	senux	7	5	0,0602
3	atechnikality	8	5	0,0595
4	sagarjena	9	5	0,059
...				
15	shankar00	2	2	0,025
...				
408	raouf	3	1	0,012
...				
429	modder	5	1	0,01176471

Table 6. Collaborators of "raouf"

Number	Collaborators	Projects	Common projects	Attribute _{Score}
1	raouf	3	3	1
2	bvencel	1	1	0,33
3	KathyWu	1	1	0,33
4	srikanth602	1	1	0,33
5	Lickie	1	1	0,33
...				
15	raja4567	81	1	0,01204819

4.1 Key Terms Computation for Developers

At first, we have calculated the keywords for the "modder", "raja4567" and "raouf". We have selected only the some terms for illustration (see Table 7). For comparison we marked some terms (bold text), which was used as a context between developers.

In the Figure 1 is whole network of collaborators for the selected terms "iphone", "wp7", "android". The edge weights are evaluated by *Score*. This subnetwork has 199 connected components (communities) with collaborating developers.

Table 7. Key Terms for the persons "modder", "raja4567" and "raouf"

number	t_k	t_k in $T_{p_{modder}}$	t_k	t_k in $T_{p_{raja4567}}$	t_k	t_k in $T_{p_{raouf}}$
1	mediascout	1	licens	1	torchlight	1
2	ne	0,7800623	distribut	0,7	resx	0,7017544
3	sal	0,5980892	contributor	0,6934211	crunch	0,6028985
4	movi	0,4556041	work	0,6413794	decenc	0,333333
5	rapid	0,4191964	term	0,5994475	svt	0,3301587
6	rockethub	0,401282	notic	0,5570145	blackberri	0,2729659
7	myne	0,401282	modif	0,5359043	empti	0,2396313
...						
14					android	0,1438451
...						
143	wp7	0,1152854				
...						
1305			android	0,01322994		
...						
2572			iphon	0,00643989		
...						
2587			wp7	0,006402954		

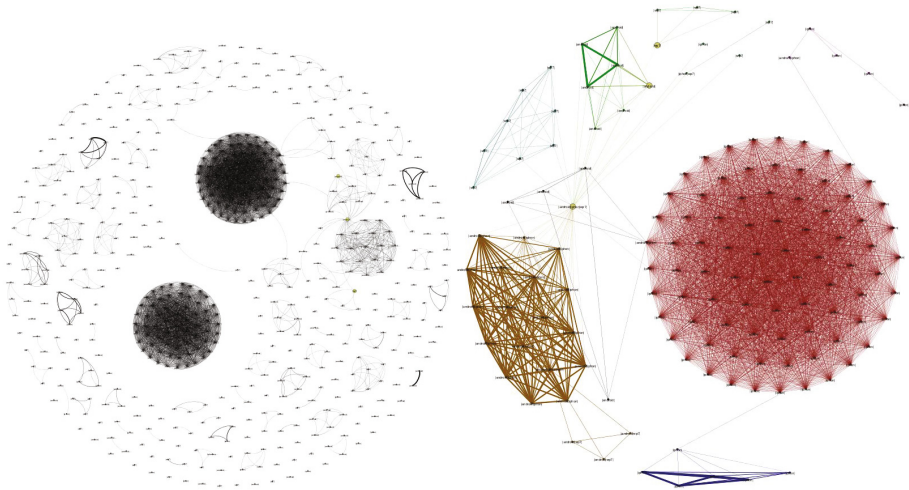


Fig. 1. Synthetic collaborators network for the terms "iphone", "wp7", "android" and selected subnetwork with developers "modder", "raja4567" and "raouf"

Second part of Figure 1 shows graph of the connected component which contain selected and highlighted developers. We can see that selected developers are not in the one community of collaborators. They are connected, but the relation is too weak. They are not suitable for recommendation.

We used our algorithm for spectral clustering [7] and we detect communities of more collaborated developers. Then we can recommend the "green" community, which contain developers with the stronger relation in the context of selected words.

4.2 Subnetwork of Projects

We chose the terms "iphone", "wp7", "android" and create subnetwork of projects. The graph of this subnetwork (see Figure 2) is not connected (contain 243 connected components) and most components are isolated vertices. When we extend the selected terms and create the new subnetwork in the context with terms "iphone", "wp7", "android" + "silverlight", than is obvious a importance of the term "silverlight" which connect more projects.

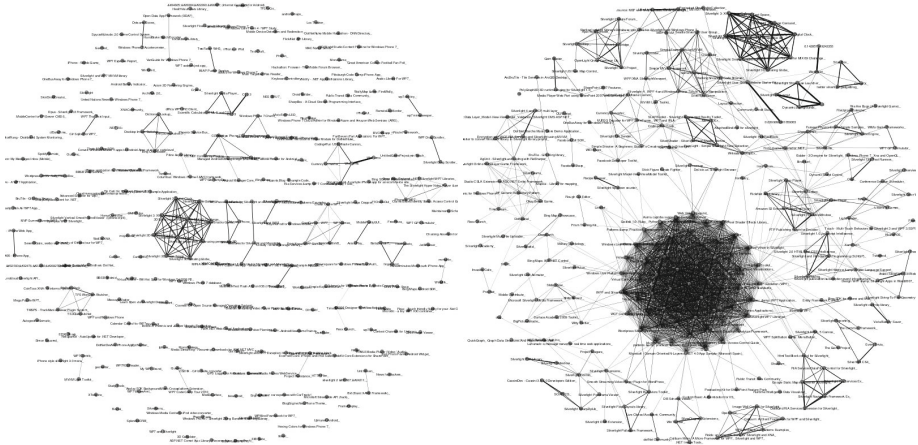


Fig. 2. Extracted subnetwork of projects for selected terms "iphone", "wp7", "android" and the other subnetwork have terms "iphone", "wp7", "android" + "silverlight"

The Figure 1 and the Figure 2 were visualized using the program Gephi¹.

5 Conclusion

Research presented in this article is oriented to the strength extraction between persons based on their context in the CodePlex. The method was presented using the data collection from the CodePlex database, which contains information of the activities of developers in the project. The proposed method is usable for the development of collaboration network. The description of this network is based on the set of terms (as the input), which are used in the description of projects by the given developer. Using

¹<http://gephi.org/>

this method, we have obtained the new weight in the synthetic collaborators network. By means of the set of selected term, belonging to one (or more) persons, we can construct the subnetwork with only the context-related collaborators. This subnetwork can be very helpful in searching of the persons who are interested in the same area, defined by the selected term. It is usable for members of the project management, who need to find suitable developers specialized to certain area.

Acknowledgment. This work was supported by SGS, VSB – Technical University of Ostrava, Czech Republic, under the grant No. SP2012/151 Large graph analysis and processing and by the Bio-Inspired Methods: research, development and knowledge transfer project, reg. no. CZ.1.07/2.3.00/20.0073 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic.

References

1. Chen, J., Geyer, W., Dugan, C., Muller, M., Guy, I.: Make new friends, but keep the old. In: Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, vol. (1), p. 201 (2009)
2. Deza, M.-M., Deza, E.: Dictionary of Distances. Elsevier Science, Amsterdam (2006)
3. Konchady, M.: Text Mining Application Programming, 1st edn. Charles River Media (May 2006)
4. Long, Y., Siau, K.: Social network structures in open source software development teams. *Journal of Database Management* 18(2), 25–40 (2007)
5. Minks, S., Martinovic, J., Drazdilova, P., Slaninova, K.: Author cooperation based on terms of article titles from dblp. In: IHCI 2011 (2011)
6. Porter, M.F.: An algorithm for suffix stripping. *Program* 14(3), 130–137 (1980)
7. Vojacek, L., Martinovic, J., Slaninova, K., Drazdilova, P., Dvorsky, J.: Combined method for effective clustering based on parallel som and spectral clustering. In: DATESO, pp. 120–131 (2011)

A Graph-Based Formalism for Controlling Access to a Digital Library Ontology

Subhasis Dasgupta¹ and Aditya Bagchi²

¹ Indian Statistical Institute, 203 B T Road, Kolkata 700108, India
dasgupta.subhasis@gmail.com

² Indian Statistical Institute, 203 B T Road, Kolkata 700108, India
aditya@isical.ac.in

Abstract. This paper presents a graph-based formalism for an Ontology Based Access Control (OBAC) system applied to Digital Library (DL) ontology. It uses graph transformations, a graphical specification technique based on a generalization of classical string grammars to nonlinear structures. The proposed formalism provides an executable specification that exploits existing tools of graph grammar to verify the properties of a graph-based access control mechanism applicable to a digital library ontology description. It also provides a uniform specification for controlling access not only at the concept level but also at the level of the documents covered by the concepts including node obfuscation, if required.

Keywords: Ontology, Digital Library, Multiple Inheritance, OBAC.

1 Introduction

Recent study on the modeling of digital library (DL) suggests an ontological structure for its representation [1], where documents may be classified and stored against appropriate concepts present in the ontology. Such DL ontology usually considers an underlying tree structure for its implementation, i.e. one concept can have only one parent concept [2][3]. However in real life situation, a particular concept may have more than one parent concepts. For example, a concept named Database may be reached from Computer Science & Engineering (CS), Geographic Information System (GIS) or Biology/Bio-informatics (BIO). This consideration changes the underlying structure of the ontology from a tree to a Directed Acyclic Graph (DAG).

Now, the three parent concepts of Database may have distinct or even overlapping user communities. As a result, any document under Database may be of interest to more than one of the above three user communities. Research work to control access to a digital library, done so far, ensures that a user must possess appropriate authorization to get access to a concept. However, if access to a concept is granted, all documents under it are available to the concerned user. Some work has already been done to control access even at the document level. In other words, a user getting access to a concept may not get access to all the documents covered by that concept, particularly in a situation when a concept

has multiple parent concepts. This gives rise to a flexible access control system for a DL environment hitherto unexplored [4]. This earlier work considered some implementation issues related to such access control environment. Present paper, however, offers a graph-based formalism for an Ontology Based Access Control (OBAC) system applicable to Digital Library (DL) ontology. It uses graph transformations, a graphical specification technique based on a generalization of classical string grammars to nonlinear structures [5][6]. The proposed formalism is useful for the following reasons:

- To specify the properties of a proposed Access Control specification for Digital Library Ontology
- To provide an executable specification that exploits existing tools to verify the properties of a graph-based access control mechanism applicable to a digital library ontology description.
- To provide a uniform specification for controlling access not only at the concept level but also at the level of the documents covered by the concepts including node obfuscation.

Similar attempt has already been taken to model Discretionary, Lattice based and Role Based Access Control system using similar graph-based formalism [7][8]. However, formalism proposed in this paper has considered a single user environment. Role/User Group and any possible conflict arising out of them will be studied later.

While Section 1 provides the introduction, Section 2 covers the technological preliminaries. Section 3 provides the graph model of concept hierarchy and Section 4 gives the fundamentals of graph transformation and security model. Section 5 covers the policy algebra. Section 6 draws the conclusion.

2 Technological Preliminaries

This paper proposes a flexible access control system for retrieving documents using a digital library ontology supporting an underlying DAG structure to handle multiple parent concept problem. So here a concept may have more than one parent concept in the hierarchy. As a result, the documents under a concept can be categorized against the concepts above it. A user can access a document only if he/she has appropriate authorization to access the category to which the document is placed. Figure 1 shows an environment where documents covered under the concept Database may be contributed by or of interest to any users of the parent concepts. So a document under a child concept can be a member of one or more than one of the parent concepts. Consequently, documents under a child concept having n parents, can be classified into $(2^n - 1)$ categories. So, the Database concept in Figure 1 can be classified into $(2^3 - 1)$ or 7 categories. Figure 2 shows the Venn diagram corresponding to the concept Database having three parent concepts Computer Science (CS), Geographic Information System (GIS) and Bio-Informatics (BIO) as explained earlier. So, a document under the concept Database may be of interest to the users of CS/GIS/BIO or any combinations of them. Situation depicted in Figure 1 and Figure 2 is very common

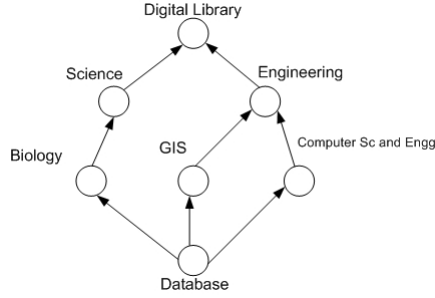


Fig. 1. An ontological structure with the concept Database having three parent concepts Biology, GIS and Computer Sc and Engg

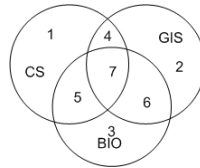


Fig. 2. Possible document categories under the common concept "DATABASE"

in case of a digital library. However, the present implementations avoid such document classification possibility and keep the child concept under the parent concept that contributes maximum number of documents. So, according to the above example, the concept Database would possibly be kept in the CS path with all documents under it. Any user of GIS or BIO community will directly access the Database concept and would be able to get all the documents under it. However, the proposed system provides $(2^3 - 1) = 7$ document classes as shown in Figure. 3. Depending on the parent concept where a user is authorized, corresponding document classes under Database concept and hence the documents covered by them can be accessed. An ontology based system has been discussed in many documents [9][10][11]. This section describes the related technologies.

- **Ontology:** The fundamental objective of Ontology is to form a semantic network based system for organizing concepts in a directed graph structure and to provide a mechanism to search a concept from such a structure by which a given schema element is referred. It also finds other related elements/concepts in the ontology. Ontology can be defined by a directed graph, where each node is a concept. If O is an ontology represented as $O = (C, L)$ then C is a concept, and L is the link between two concepts representing their semantic relationship.
- **Properties:** Each ontology has a set of properties, classified into either object property or data property. Data property describes about the data and Object property deals with the concepts. All domain property of a

concept c can be represented as $P(c) = DP(c) \cup OP(c)$ [10], where $DP(c)$ is the data property and $OP(c)$ is the concept property. The proposed system has used two types of links between concepts: *isSubClassOf* and *hasContributedTo* to represent the relations among concepts. These are object properties. In Figure. 1, *Biology (isSubClassOf) Science*, so the $OP(C_{Biology}.(isSubClassOf)) \in \{Science\}$.

- **Concept:** Each ontology has a set of semantically related concepts, $C(o) = \{c_1, c_2, \dots, c_n\}$. Fig. 1 is showing a Digital Library(DL) ontology, hence $C(DL) = \{C_{database}, C_{biology}, C_{computerScandEngg}, \dots, C_{DigitalLibrary}\}$.
- **Concept Hierarchy:** Ontology structure is a DAG, where a concept may have more than one parent concepts. Concepts in an ontology o can be represented as a partial order set. Given two concept $(Science, Biology) \in (DigitalLibrary)$ where $isSubClassOf(Biology) = Science$, i.e. Biology is more specialized than Science. It can also be denoted as $C_{Biology} \prec C_{Science}$.
- **Document Class:** Documents in a concept are classified into several classes depending upon the number of first degree parents. If a concept has n number of parents then the concept should have $(2^n - 1)$ number of document class. In Figure. 1 database has three parent concepts. So the documents covered by the concept database has 7 possible document classes.
- **Document Anotation:** Gonçalves et. al. has published some work on ontological representation of digital library. Concept of an ontology can be identified by it's URI. In the present system, a document is identified by its concept URI with a unique document-id suffixed.

3 Graph Model of Concept Hierarchy

A Graph model has been adopted in this paper to represent the DL Ontology. In an ontology, each node represents a concept and each link running from one concept to another represents the relationship between the concerned concepts. Present research effort has considered two types of relationship:

1. **isSubclassOf** : *isSubClassOf* relationship represents a partial ordered set. In the graph model, *isSubClassOf* represents the parent-child relationship. In Figure 1, *Biology (isSubClassOf) Science* i.e. in the graph model Biology will be a child concept of Science. *isSubClassOf* is a non-commutative property.
2. **hasContributedTo** : Multiple parent relationship has been represented through *hasContributedTo* relationship. In Figure 1 *Database* has been contributed by three parent concepts *Biology*, *GIS* and *Computer Sc and Engg*. This relationship has been represented as *hasContributedTo* in the Graph Model. *Biology hasContributedTo Database*. This relationship is also non-commutative.

3.1 Multiple Parent Representation

As explained earlier, if n number of parent concepts contribute to a child concept then there would be $(2^n - 1)$ number of document classes, i.e. if a node has

three parent concepts then there would be 7 possible classification of documents. This relationship is also represented by *hasContributedTo* relation. Figure.3 illustrates the relationship for Database Concept.

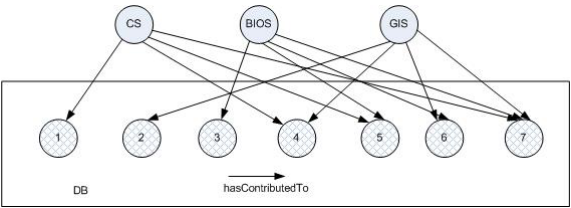


Fig. 3. *hasContributedTo* Relationship for all the Classification

Table 1. Document Class

Document Class	Access
1	CS
2	GIS
3	BIO
4	CS , GIS
5	CS , BIO
6	BIO, GIS
7	CS, BIO, GIS

3.2 Entities of OBAC Model

OBAC can be represented by a set of Subject (*S*), Object (*O*), Access Rights (*A*) and Sign (*V*). Depending upon the context and policy, subject can be users, groups, roles or applications. Object, on the other hand, can be a group of ontologies, an ontology, a concept within an ontology or a document inside a concept. Present effort has considered only read and browse operations for a digital library. Sign can either be positive or negative, signifying access permission or explicit denial respectively. OBAC model contains following entities :

- 1. User : User may be an individual user or a group of users. $U = \{u_i\}, i \geq 1$
- 2. Object : Object can be a document under a concept or a concept itself or an entire ontology or even a group of ontologies. $O = \{o_i\}, i \geq 1$
- 3. Subject : A Subject may refer to an individual or a user group or a role or may even be a web service.
- 4. Access rights : At present, the OBAC model has considered *read* and *browse* access rights only. $A \in \{read, browse\}$. A valid user can browse through all the nodes by default but would need explicit positive authorization to read any document under a concept.

5. Sign : Sign are of two types, $ve+$ or $ve-$, positive for access permission and negative for explicit denial. $\vartheta = \{+, -\}$
6. Policy : Access control policy is a tuple (S, O, a, ϑ) , where S is the subject, O is the object, a is the access rights and ϑ is the sign.

3.3 Relationship in OBAC

As mentioned earlier, ontology contains a set of related concepts and the relationship among the concepts with their data properties and object properties. Present paper considers *isSubClassOf* and *hasContributedTo* relations among the concepts.

From the inter-concept relations, following relationships can be derived in the OBAC model:

1. Inclusion(\odot) : $(C_i \odot C_j)$ signifies that concept C_i is included in concept C_j . Inclusion relationship is non-commutative i.e. $C_i \odot C_j \neq C_j \odot C_i$. However, Inclusion relationship is transitive.
2. Inferable (\implies) : If a concept C_i infers the existence of another concept C_j then C_j is inferable from C_i i.e. $C_i \implies C_j$. Inferable relationship is non-commutative, i.e. $C_i \implies C_j \neq C_j \implies C_i$, and transitive i.e. if $C_i \implies C_j$ and $C_j \implies C_k$ then $C_i \implies C_k$.
Since the present proposal considers only *isSubClassOf* relation between concepts and *hasContributedTo* relation between concepts and document classes, Inclusion and Inferable would virtually be similar.
3. Partially Inferable: (\rightharpoonup) : If C_i and C_j are two concepts, then $C_i \rightharpoonup C_j$ signifies that the concept C_i can partially infer the concept C_j . This relationship is also non-commutative and transitive. This relationship will be particularly important for concepts with multiple parents.
4. Non Inferable (\nRightarrow) : If a concept C_i cannot infer the existence of another concept C_j , the relationship is non-inferable.

3.4 Authorization

An authorization is defined by a four tuple $(\rho, \delta, v, \vartheta)$ where ρ is a member of the subject set S , δ is a member of the object set O , v is a member of the set of access rights A available in the DL system and ϑ is either $+ve$ or $-ve$ signifying positive or negative authorization respectively. In Concept level authorization, an object O is identified by its ontology URI (with the path expression). Each concept C maintains an authorization list represented by S, A, V where a particular authorization of the form (ρ, v, ϑ) signifies that $\rho \in S, v \in A, \vartheta \in \{+, -\}$. Here, the subject ρ can access concept C with access right v if $\vartheta = +ve$ or cannot access the same if $\vartheta = -ve$. Once again in the present paper, A is limited to read and browse only. $\vartheta = +ve$ signifies read is permitted and $\vartheta = -ve$ if read is not permitted. In the present proposal browse to any concept is permitted by default.

4 Fundamentals of Graph Morphism and Security Model

This section discusses the formal method of graph transformation, i.e. transformation steps and rules. A formal introduction to Graph based formalism can be found at [6] and a RBAC implementation of the model has been developed by Koch et. al. [7]. In very brief, a graph represents a state of a system and a rule is a transformation $\{r : L \rightarrow R\}$, where both L and R are graphs i.e. the left-hand side transforms to the right-hand side by the graph transformation rule. L is the original graph and R is the transformed graph after applying relevant access control policies. The rule, $\{r : L \rightarrow R\}$ consist an injective partial mapping from left-hand side to right-side, among the set of nodes r_n and the set of links/relations r_e . Each mapping, should be compatible with graph structure and the type of the node. If the mapping is total the graph morphism is total. The L describe which objects a Graph G must contain for the rule $\{r : L \rightarrow R\}$ to be applicable to G . Nodes and edges of L , whose partial mapping are un-defined or restricted by the rules, will be either deleted or modified. However, nodes and edges of L , those are defined at rule will be stored at R after transformation. The basic idea of a graph transformation [6] considers a production rule $p : (L \rightsquigarrow R)$, where L and R are called the left-hand and right-hand side, respectively, as a finite schematic description of potentially infinite set of direct derivations. If the *match* m fix the occurrence of L in a given graph G , then $(G \xrightarrow{p,m} H)$ denotes a direct derivation where p is applied to G leading to directive graph H . The H is obtained by replacing the occurrence of L in G by R . From algebraic approaches, the graph has been considered as a collection of edges (E) and vertices (V). However, source $s : E \rightarrow V$ and target $t : E \rightarrow V$ are two unary operations. Figure 4, shows the intermittent state of transformation, where each graph production $p : (L \rightsquigarrow R)$ defines the partial correspondence between the elements of left-hand side and the right-hand side on the basis of a rule, determining which edge should be present and which edge should be deleted. A match $m : L \rightarrow R$ for a production p is a graph homomorphism, mapping nodes and edges of L to R in such a way that the graphical structure and the levels are preserved. Considering the scenario in Figure 4 consider the production $p_1 : (L_1 \rightsquigarrow G_1)$ which

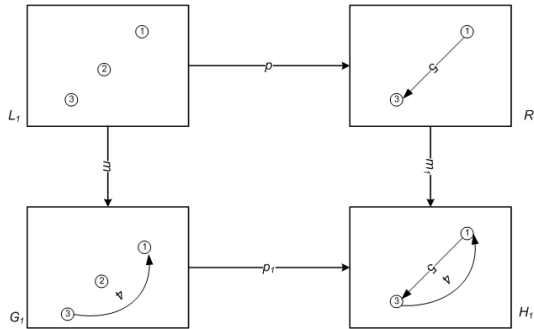


Fig. 4. Graph Morphism Basics

applied on the graph G_1 . The number written in the vertices and edges consider the partial correspondence between $L_1 \rightarrow G_1$. Same number represent the same object before and after the transformation. The production $p_1 : (L_1 \rightsquigarrow G_1)$ gets three vertices at the left hand side L_1 and leaves behind two vertices, connected by an edge depicting a rule to permit information flow for security. The match $m_1 : (L_1 \rightarrow G_1)$ maps each element of L_1 to the element of G_1 carrying the same set of numbers. Following the other production/transformation rules in the same way, derived rule for H_1 will be $G_1 - (L_1 - R_1) \cup (R_1 - L_1)$.

4.1 Example of Graph Morphism

Notion of rule and its application to a graph have been described by using three examples. Three example cases are **Node Traversal**, **Node Obfuscation** and **Partial Inferences**. Figure 5 describe the Node traversal. Let, a query wants to access a document under the concept *Bio*. However, the access of concept *Bio* can be done through *Science* node. Hence, the query will try to access the node through the concept *Science*. In the L rule hence a dashed line has been added. The dashed edge in the left side represents a *negative application condition* [7]. A *negative application condition* for the rule $\{r : L \rightarrow R\}$ is a set of (L, N) where L is the subgraph of N , and N is the structure as a whole and must occur in the original structure, i.e. the whole ontology G . Hence, in Figure 5 query has inferred its access from the rule and right hand side has shown the access path.

Node obfuscation is a very common issue for this kind of structure both in XML and Ontological environments [12] [13] [14]. In the previous example, query can trace the exitance of the node *Science*. If the name of the node is sensitive for some user then system will block the identity of the node and it will be obfuscated for the concerned user. Now the query will pass through the node without accessing the node name or its structural details. Documents covered by an obfuscated node will not be available to the user as well. Figure 6 creates this scenario. In the left side the query intended its access for node *Science*, which has a *negative application condition*, and by the rule set the system has found that node *Science* should be obfuscated for this query. Hence, the rule $\{r : L \rightarrow R\}$ creates the new graph on the right hand side. **Partial Inferable** is another prolific problem discussed in many research papers. Mutiple parent condition

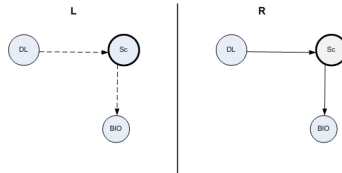


Fig. 5. Graph Transformation for Node Inference

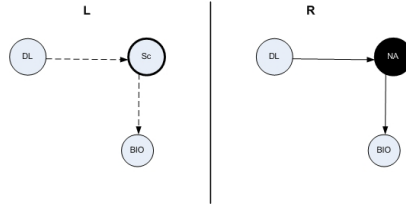


Fig. 6. Graph Transformation for Node Obfuscation

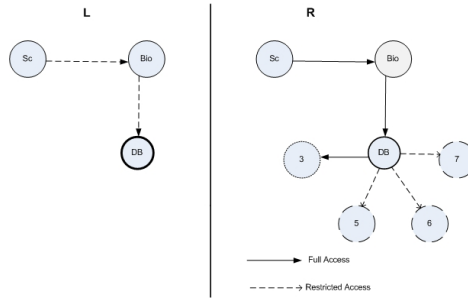


Fig. 7. Graph Transformation for Partial Inference

Table 2. Example Policy for Partial Inference

Name	Source	Joint	No Contribution
Policy A	Full	abstract	No access
Policy B	Full	No access	No access
Policy C	Full	Full	abstract

has also been addressed here through graph morphism. As mentioned earlier, that documents under *database* concept have been contributed by three parent concepts and thus the documents under *database* concept can be classified into $(2^3 - 1)$ classes. Graph transformation ensures partial access to documents or access to relevant document classes only.

5 Policy Algebra

The production rules of the graph are generated by a rule engine depending upon the access control algebra. This section will define the formal algebra for our model. Our model considers each node as a concept and the concept level authorization can be represented by S, O, A and V , where S is the concept or ontology or a group of concept from a set of ontology, concepts are related through

semantic relationships, we are considering *isSubClassOf* and *hasContributedTo* relations here. Thus authorization for concept i can be represented as :

$$ca_i = (\rho_i, \delta_i, v_i, \vartheta) \quad (1)$$

Where , ρ_i, δ_i, v_i are the instances from S, O, A . We denote set of authorization explicitly specified for a concept. In addition to explicit authorizations, other authorizations can be generated through propagation. Hence, a propagation from concept i to j can generate the authorization according to the policy. Policy defines authorization of an object for a subject. Hence a simple policy can be defined as

$$E_i = (C_i, u_i, \{R\}, \{+\})$$

where, C_i is a constant of S or a variable over S , u_i is a constant of O or a variable over O , R is the access right and $+$ is the sign. Two or more policies can be joined to create combined policy. Hence, the combination of policy can be expressed as BNF grammar, i.e. a combination of policies can be written as

$$E ::= id|E + E|E \& E|E - E|E.C|o(E, E, E)|E * r|T(E)|(E) \quad (2)$$

$$T ::= \tau id.T|\tau.E \quad (3)$$

Here , id is the token type of policy identifiers, E is the nonterminal describing policy *expression* , T is the policy template, C and r are the authorization content and authorization rules respectively. Above symbols have been disambiguated by proper precedence rules of the operators [15]. The policy can be stated explicitly or generated by some rule engine. Combination of more than one policy may be enforced over one *environment*, we will refer them as the composite policy. Using the rule , some policies can be obtained by the grammar through algebraic operators. We refer them as derived policies E_{der} which should satisfy the following rules :

1. Reflexivity : For all tuples of the E are inherited by E_{der}
2. Inheritance Rule : $(C_i, u_j, \{R\}, \{+\}) \in E \wedge (\forall u_i \in U | u_j \xrightarrow{a} u_i) \longrightarrow (C_i, u_j, \{R\}, \{+\}) \in E_{der}$
3. Override Rule : $(C_i, u_j, \{R\}, \{+\}) \in E \wedge (C_i, u_j, \{R\}, \{-\}) \in E_{der} \longrightarrow E_{der} \wedge (C_i, u_j, \{R\}, \{+\}) \notin E_{der}$
4. Commutative : $E_a + E_b = E_b + E_a$

5.1 Policy Construction

OBAC system contains two type of policies, **Simple Policy** and **Composite Policy**, Simple policy is the granularity of policy i.e. the mapping of rules with ground algebra, where as combinations of more than one simple policies can create a composite policy. Consider the previous example of *CS*, *GIS*, *BIO* and Database(*DB*). We are considering that all the documents in *DB* has been contributed by three parent concepts. The policy of each concept has been represented by P_{CS} , P_{GIS} and P_{BIO} respectively, and those are the **Simple Policy**.

On the contrary, the documents at **DB** has been annotated by either *cs.db*, *gis.db* or *bio.db*. Now the policy for accessing document (d) for **DB** will be a **Composite Policy** that can be represented as:

$$\Pi_{DB} = \left\{ \begin{array}{ll} P_{gis.db} + P_{cs.db} + P_{bio.db} + \delta_{db} & \text{if } (d \leq cs.db \wedge d \leq gis.db \wedge d \leq bio.db) \\ P_{cs.db} + P_{gis.db} + \delta_{db} & \text{if } (d \leq cs.db \wedge gis.db) \\ P_{bio.db} + P_{gis.db} + \delta_{db} & \text{if } (d \leq bio.db \wedge gis.db) \\ P_{cs.db} + P_{bio.db} + \delta_{db} & \text{if } (d \leq cs.db \wedge bio.db) \\ P_{cs.db} + \delta_{db} & \text{if } (d \leq cs.db) \\ P_{gis.db} + \delta_{db} & \text{if } (d \leq gis.db) \end{array} \right. \quad (4)$$

Where, Π_{DB} is the composite policy of **DB**. Here δ_{db} represents the concept specific constraints common for all document classes. Policy of a interrelated concepts/ontology can be represented by *Composite Policies*. If we have n number of related concept in an ontology, we can represent the policy for the total ontology as:

$$\Pi_C = \Pi_1 + \Pi_2 + + \Pi_k + \Pi_{k+1} + + \Pi_n \quad (5)$$

Where, Π_C is the policy of the ontology and Π_1, Π_2 are the policies of the respective concepts.

6 Conclusion

This paper provides a graph based formalism for controlling access to a digital library ontology. Using a graph specification technique based on classical string grammars to nonlinear structures, this paper proposes graph transformation rules to represent different access control situations. Corresponding Policy algebra and policy derivations have also been provided for clarification of the formalism. The entire study has not considered user groups/roles and any possible conflicts arising out of them. It would be part of future work.

References

1. Gonçalves, M.A., Fox, E.A., Watson, L.T.: Towards a digital library theory: a formal digital library ontology. *Int. J. Digit. Libr.* 8, 91–114 (2008)
2. Adam, N.R., Atluri, V., Bertino, E., Ferrari, E.: A content-based authorization model for digital libraries. *IEEE Transactions on Knowledge and Data Engineering* 14, 296–315 (2002)

3. Ray, I., Chakraborty, S.: A Framework for Flexible Access Control in Digital Library Systems. In: Damiani, E., Liu, P. (eds.) *Data and Applications Security 2006*. LNCS, vol. 4127, pp. 252–266. Springer, Heidelberg (2006)
4. Dasgupta, S., Bagchi, A.: Controlled Access over Documents for Concepts Having Multiple Parents in a Digital Library Ontology. In: Chaki, N., Cortesi, A. (eds.) *CISIM 2011*. CCIS, vol. 245, pp. 277–285. Springer, Heidelberg (2011)
5. Ehrig, H., Engels, G., Kreowski, H.J., Rozenberg, G. (eds.): *Handbook of graph grammars and computing by graph transformation: applications, languages, and tools*, vol. 2. World Scientific Publishing Co., Inc., River Edge (1999)
6. Corradini, A., Montanari, U., Rossi, F., Ehrig, H., Heckel, R., Löwe, M.: In: *Algebraic approaches to graph transformation. Part I: basic concepts and double pushout approach*, pp. 163–245. World Scientific Publishing Co., Inc., River Edge (1997)
7. Koch, M., Mancini, L.V., Parisi-Presicce, F.: Graph-based specification of access control policies. *Journal of Computer and System Sciences* 71(1), 1–33 (2005)
8. Koch, M., Mancini, L.V., Parisi-Presicce, F.: On the specification and evolution of access control policies. In: *SACMAT*, pp. 121–130 (2001)
9. Kashyap, V., Sheth, A.: Semantic and schematic similarities between database objects: a context-based approach. *The VLDB Journal* 5, 276–304 (1996)
10. Qin, L., Atluri, V.: Semantics aware security policy specification for the semantic web data. *Int. J. Inf. Comput. Secur.* 4, 52–75 (2010)
11. Ouksel, A.M., Ahmed, I.: Ontologies are not the panacea in data integration: A flexible coordinator to mediate context construction. *Distributed and Parallel Databases* 7, 7–35 (1999), doi:10.1023/A:1008626109650
12. Damiani, E., di Vimercati, S.D.C., Fugazza, C., Samarati, P.: Modality conflicts in semantics aware access control. In: Wolber, D., Calder, N., Brooks, C.H., Ginige, A. (eds.) *ICWE*, pp. 249–256. ACM (2006)
13. Gabillon, A.: A Formal Access Control Model for XML Databases. In: Jonker, W., Petković, M. (eds.) *SDM 2005*. LNCS, vol. 3674, pp. 86–103. Springer, Heidelberg (2005)
14. Kaushik, S., Wijesekera, D., Ammann, P.: Policy-based dissemination of partial web-ontologies. In: Damiani, E., Maruyama, H. (eds.) *Proceedings of the 2nd ACM Workshop On Secure Web Services, SWS 2005*, Fairfax, VA, USA, November 11, pp. 43–52. ACM (2005)
15. Bonatti, P.A., di Vimercati, S.D.C., Samarati, P.: An algebra for composing access control policies. *ACM Trans. Inf. Syst. Secur.* 5(1), 1–35 (2002)

Role Approach in Access Control Development with the Usage Control Concept

Aneta Poniszewska-Maranda

Institute of Information Technology, Technical University of Lodz, Poland
anetap@ics.p.lodz.pl

Abstract. Development of information technology, progress and increase of information flow have the impact on the development of enterprises and require rapid changes in their information systems. Very important stage of data protection in information system is the creation of high level model, satisfying the needs of system protection and security. This paper presents the role engineering aspects of access control for dynamic information systems. The objective is to find the approach of access control to ensure in perspective the global coherence of security rules for all components of an system.

1 Introduction

Development of information technology, progress and increase of information flow have the impact on the development of enterprises and require rapid changes in their information systems. The growth and complexity of functionalities that they currently should face causes that their design and realization become the difficult tasks and strategic for the enterprises at the same time. Data protection against improper disclosure or modification in the information system is an important issue of each security policy realized in the institution.

Role engineering for role-based access control is a process consisting of determination and definition of roles, permissions, security constraints and their relations. The company's roles can be regarded in two aspects - functional (reflects the main business functions of the company) and organizational (reflects the organization hierarchy of the company). Before a concrete access control model can be implemented, the role engineering process has to take place. The aspect of role engineering is connected close to the aspects of analysis and design of information systems. During our previous studies, we examined the design methods of the information systems and the design methods of information system security [7]. However, it is difficult to find the global method that takes into account both the design of system and its associated security scheme.

The process of roles identification and setting up in an organization is a complex task because very often the responsibilities of actors in an organization and their functions are few or badly formalized. Moreover, the role concept is an abstract approach. It does not correspond to particular physical being and therefore it is very difficult to give definitions that comprise the whole world.

To conclude, the research of roles in security schema needs a real engineering approach that provides a guide identify, specify and maintain them.

Many different works concerning the problem of role engineering were presented in the literature [13–19] and others. First of all, most of the paper tread the static aspects of access control domains. They do not take into consideration the problems of dynamic changes affecting the access control rules in modern information systems, particularly distributed information systems. Moreover, whereas everyone agrees that safety must be taken into account as quickly as possible, the proposed role engineering approaches are often disconnected from the design and the evolution of information system for which they are provided.

Coyne in [16] proposes to collect different user activities to define the candidate roles. Fernandez et al. in [14] propose to use the use cases to determines the role rights of system users but do not describe the constraints aspect and role-hierarchies. Epstein et al. [17] propose to express RBAC elements with UML diagrams but do not define the role engineering process. Roeckle et al. [19] propose a process-oriented approach for role engineering but only on meta level without details about role hierarchy, derivation of permissions and relation between some elements. Epstein and Sandhu [18] describe role engineering of role-permission assignment but not go into detail about constrains and role hierarchies in the process. Neumann and Strembeck propose in [9, 10] very interesting scenario-driven role engineering process for RBAC model. However this proposition does not take into consideration the dynamic aspects of access control.

This paper presents the engineering aspects in access control of dynamic information systems. The paper is composed as follows: section 2 presents the role concepts and usage concept in aspect of access control, section 3 gives the outline of approach based on these concepts (Usage Role Based Access Control - URBAC). Section 4 deals with the cooperation of two actors in role creation process of information system security, while section 5 shows the representation of URBAC approach using the UML concepts. Section 6 describes the process of roles production based on URBAC approach presenting the stage of security profile creation for the system's users.

2 Access Control Based on Role and Usage Concepts

Development in area of information systems, especially modern, distributed information systems causes that traditional access control models are not sufficient and adequate for such systems in many cases. Some imperfections were found in domain of these security models [7]:

- traditional access control models give only the possibility to define the authorizations but do not provide the mechanisms to specify the obligations or conditions of access control,
- developers or security administrators can only pre-define the access rights that will be next granted to the subjects,

- decision about granting or not an access can be only made before the required access but not during the access,
- mutable attributes can be defined for subjects and for objects of a system.

These disadvantages and the needs of present information systems caused the creation of unified model that can encompass the use of traditional access control models and allow to define the dynamic rules of access control. Two access control concepts are chosen in our studies to develop the new approach for dynamic information systems: role concept and usage concept. The first one allows to represent the whole system organization in the complete, precise way while the second one allows to describe the usage control with authorizations, obligations, conditions, continuity (ongoing control) and mutability attributes.

Role-based access control approach [1] or its extensions [7, 8] requires the identification of roles in a system. The role is properly viewed as a semantic structure around which the access control policy is formulated. Each role realizes a specific task in the enterprise process and it contains many functions that the user can take. For each role it is possible to choose the necessary system functions. Thus, a role can be presented as a set of functions that this role can take and realize. Each function can have one or more permissions, and a function can be defined as a set or sequence of permissions. If an access to an object is required, then the necessary permissions can be assigned to the function to complete the desired job. Specific access rights are necessary to realize a role or a particular function of this role.

The usage control approach [4–6] is based on three sets of decision factors: authorizations, obligations and conditions that have to be evaluated for the usage decision. The obligations and conditions are added in the approach to resolve certain shortcomings characteristic for the traditional access control strategies. The most important properties of usage control that distinguish it from traditional access control concepts and trust management systems are the continuity of usage decisions and the mutability of attributes. Continuity means that usage control decision can be determined and enforced before an access (as in traditional access control models) and also during the ongoing period of the access. Mutability represents the mutability of subject and object attributes that can be either mutable or immutable. The mutable attributes can be modified by the actions of subjects and the immutable attributes can be modified only by the administrative actions.

3 Approach of Role Based Access Control with Usage Control

The complexity of actual organizations (i.e. matrix based organizational structure) has guided our proposition to extend the standard RBAC model by role management facilities that allow a finer distribution of responsibilities in an enterprise. On the other hand, the dynamism of actual information systems was

the reason to use the concepts of Usage Control to develop the new implementation of access control models to support the management of information system security.

The proposed access control approach was based on two access control concepts: role concepts and usage concepts. It was named Usage Role-based Access Control (URBAC) [20, 21]. The term usage means usage of rights on information system objects. The "rights" include the rights to use particular objects and also to delegate the rights to other subjects.

The detailed view of usage role-based access control approach is presented on figure 1.

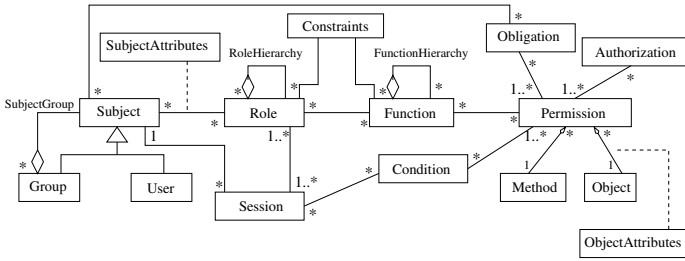


Fig. 1. Meta-model of URBAC model

Subjects can be regarded as individual human beings. They hold and execute indirectly certain rights on the objects. Subject permits to formalize the assignment of *users* or *groups of users* to the roles. Subject can be viewed as the base type of all users and groups of users in a system.

A **Role** is a job function or a job title within the organization with some associated semantics regarding the authority and responsibility conferred on a member of the role. The role can represent a competency to do a specific task, and it can embody the authority and responsibility. The roles are created for various job functions in an organization. The direct relation is established between roles and subjects that represent the users or groups of users. It is also possible to define the hierarchy of roles, represented by aggregation relation *RoleHierarchy*, which represents the inheritance relations between the roles.

The association relation between the roles and subjects is described by the association class **SubjectAttributes** that represents the additional subject attributes (i.e. subject properties) as in Usage Control. *Subject attributes* provide additional properties, describing the subjects, that can be used for the usage decision process, for example an identity, enterprise role, credit, membership, security level. Each role allows the realization of a specific task associated with an enterprise process. A role can contain many functions **Function** that a user can apply. Consequently, a role can be viewed as a set of functions that this role can take to realize a specific job. It is also possible to define the hierarchy of functions, represented by the aggregation relation named *FunctionHierarchy*, which provides the hierarchical order of system functions.

Each function can perform one or more operations, a function needs to be associated with a set of related permissions **Permission**. To perform an operation one has the access to required object, so necessary permissions should be assigned to corresponding function. The permission determines the execution right for a particular method on the particular object. In order to access the data, stored in an object, a message has to be sent to this object. This message causes an execution of particular method **Method** on this object **Object**. Very often the constraints have to be defined in assignment process of permissions to the objects. Such constraints are represented by the authorizations and also by the obligations and/or conditions.

Authorization (A) is a logical predicate attached to a permission that determines the permission validity depending on the access rules, object attributes and subject attributes. **Obligation (B)** is a functional predicate that verifies the mandatory requirements, i.e. a function that a user has to perform before or during an access. **Conditions (C)** evaluate the current environmental or system status for the usage decision concerning the permission constraint.

A constraint determines that some permission is valid only for a part of the object instances. Therefore, the *permission* can be presented as a function $p(o, m, Cst)$ where o is an object, m is a method which can be executed on this object and Cst is a set of constraints which determine this permission. Taking into consideration a concept of authorization, obligation and condition, the set of constraints can take the following form $Cst = \{A, B, C\}$ and the permission can be presented as a function $p(o, m, \{A, B, C\})$. According to this, the permission is given to all instances of the object class except the contrary specification.

The **objects** are the entities that can be accessed or used by the users. The objects can be either privacy sensitive or privacy non-sensitive. The relation between objects and their permissions are additionally described by association class **ObjectAttributes** that represents the additional object attributes (i.e. object properties) that can not be specified in the object's class and they can be used for usage decision process. The examples of object attributes are security labels, ownerships or security classes. They can be also mutable or immutable as subject attributes do. The **constraints** can be defined for each main element of the model presented above (i.e. user, group, subject, session, role, function, permission, object and method), and also for the relationships among the elements. The concept of constraints was described widely in the literature [3, 8, 11, 13]. It is possible to distinguish different types of constraints, static and dynamic, that can be attached to different model elements.

4 Two Actors in Role Creation Process of Information System Security

Two types of actors cooperate in the design and realization of security schema of an information system [8]: on the one hand it is application/system developer who knows its specification that should be realized and on the other hand it is security administrator who knows the general security rules and constraints that

should be taken into consideration on the whole company level. We propose to partition the responsibilities between these two actors in the process of definition and implementation of security schema on access control level and to determine their cooperation in order to establish the global access control schema that fulfill the concepts of URBAC. This partition of responsibilities is presented in figure 2 and the responsibilities were divided into two stages: conception stage and exploitation stage.

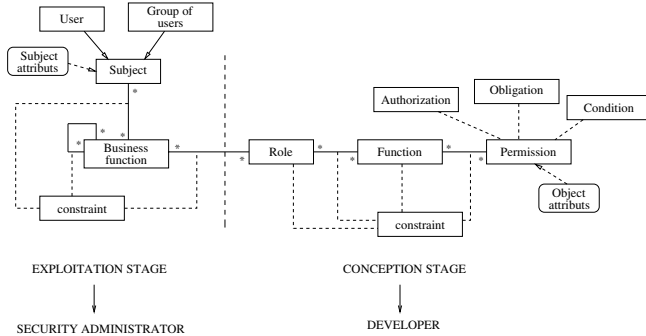


Fig. 2. Two actors in creation of information system security schema

Conception Stage. The realization process of information system or simple application is provoked by a client's request or in general by the client's needs to create a new information system or new application (i.e. to add a new component to the existing information system). Basing on the client's needs and requirements the application/system developer creates the logical model of application/system and next designs the project of this system that will be the base for its implementation. This model and next the project contain all the elements expressing the client's needs (i.e. needs of future users).

The application developer defines the elements of this application and its constraints corresponding to the client's specifications. These elements can be presented in a form adequate to the access control concepts - it will be the URBAC approach in our case. Therefore, the developer generates the sets of following elements: roles, functions, permissions and security constraints. These sets of elements should be presented to the security administrator in a useful and legible form. The duties of application developer basing on URBAC are:

- definition of permissions - identification of methods and objects on which these methods can be executed,
- definition of object attributes associated to certain objects according with access control rules,
- assignment of elements: permissions to functions and functions to roles,
- definition of security constraints associated to the elements of the application, i.e. authorizations, obligations and conditions on the permissions and standard constraints on roles, functions and their relationships.

Exploitation Stage. The exploitation stage is realized by the security administrator on the global level of information system. The security administrator defines the administration rules and company constraints according to the global security policy and application/system rules received from the developer. He should also check if these new security constraints remain in agreement with the security constraints defined for the elements of existing information system in order to guarantee the global coherence of the new information system.

The security administrator received from the developer the sets of elements in the form adequate to URBAC: set of roles, set of functions, set of permissions and set of security constraints of the application. He uses these sets to manage the global security of the information system. First of all he defines the users' rights to use the particular applications. Two sets on company level are important to define the users' rights: persons working for the enterprise (i.e. users) and functions realized in the enterprise (i.e. business functions).

Security administrator is also responsible for the definition of security constraints for these assignments on global level with respect of defined security rules. These constraints concern first of all the following relations: user-businessFunction, businessFunction-businessFunction and businessFunction-role.

Therefore, the duties of security administrator are as follows:

- definition of users' rights basing on their responsibilities and their business functions in an organization - assignment of users to the roles of information system,
- organize the users in groups and definition the access control rights for the groups of users that realize for example the same business functions - assignment of groups to the roles of information system,
- definition of subject (i.e. user or group of users) attributes associated to certain users or groups of users that allows to determine the dynamic aspects of security constraints,
- definition of security constraints for the relationships between users and roles or group of users and roles.

The second important task of security administrator is the management of set of applications assuring the global coherence of the whole system on access control level. This assurance is exceptionally important in case of addition of new application in an information system when new access control elements appear both on local level and on global level.

5 Representation of URBAC Using the UML Concepts

Unified Modeling Language (UML) is now a standard language for analysis and design of information systems [2]. It is used almost all over the world in software engineering field for object-oriented analysis and design. It has a set of different models and diagrams that allow to present the system project from different points of view showing the whole system together with its components. Some elements and concepts of UML can be used to implement the URBAC approach,

especially during the design stage of information system and its associated security schema based on URBAC.

From this reason, UML was chosen to be used in role engineering process to implement and realize the URBAC approach. To accomplish this, the concepts of UML and URBAC should firstly be joined. Two types of UML diagrams have been chosen to provide the URBAC: use case diagram and interaction diagram. The use case diagram presents the system's functions from the user point of view. It define the system's behavior without functioning details. The interaction diagram describes the behavior of one use case [2]. It represents the objects and messages exchanged during the use case processing.

The relationships between UML concepts and concepts of usage role-based access control are as follows (Fig. 3):

- role (R) from access control model can be presented as an UML actor,
- function (F) from URBAC can be represented by an UML use case,
- each actor from use case diagram can be in interaction with a set of use cases and these relations specify the relations of R-F type (between roles and functions),
- methods executed in sequence diagrams and also in other UML diagrams can represent the methods of URBAC,
- objects that occur in UML diagrams, e.g. sequence diagram, communication diagram, can be attached to the object concept of access control model,
- permissions (P) of URBAC can be found examining the sequence diagram(s) describing the particular use case,
- use case diagram offers four types of relations between its elements:
 - communication relation between an actor and a use case that represents the relation between a role and a function, i.e. R-F relation,
 - generalization relation between actors, representing the inheritance relation between roles (R-R relation),
 - two types of relations between use cases represent the inheritance relations between functions of URBAC, i.e. F-F relations
 - subject attributes (e.g. user attributes) from URBAC can be represented by the set of attributes defined for an instance of actor class of UML,
 - concept of object attributes from URBAC can be attached to set of attributes defined for the objects in its class specification.

The concept of constraints of URBAC approach corresponds directly to the constraint concept existing in UML. The security constraints of URBAC can be defined for different elements and for relations between these elements. These constraints can be presented on UML diagrams corresponding to types and locations of elements for which these constraints will be defined.

The authorization is a constraint attached to a permission that determines the permission validity depending on defined access rules. It can be represented by the UML constraint defined for the method's execution in sequence diagram.

The obligation is a constraint defined on a permission but it concerns also the subject (e.g. a user) - subject should fulfill the obligation executing a function before or during an access. This type of constraints can be presented as UML

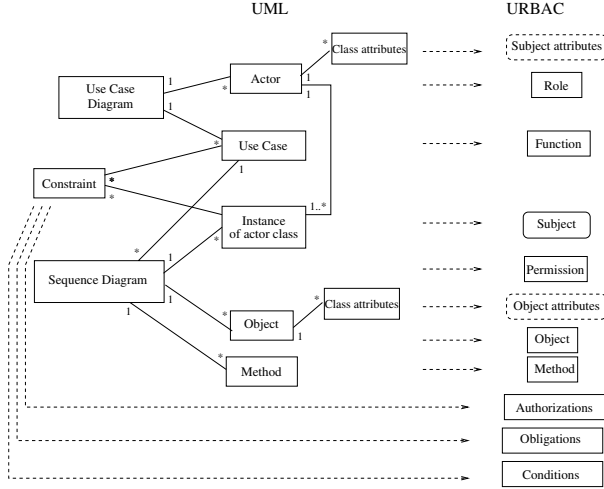


Fig. 3. Relationships between concepts of URBAC and UML concepts

constraint in sequence diagram (as pre-condition or an invariant), especially from version 2.0 of UML that provide the combinator fragments in sequence diagrams (e.g. "alt" or "opt") allowing the definition of constraint conditions.

The condition is a constraint also defined on a permission but it concerns the session element. It defines current environmental or system status and states during the user session that can be used for the usage decision. The conditions can be also represented by UML constraints defined in sequence diagrams (mainly as an invariants).

Remaining types of constraints represent the constraints defined for the roles, functions and their relations. Such constraints are represented by the UML constraints defined for actors, use cases and their relations on use case diagrams or sometimes on sequence diagrams.

6 Production of Roles Based on URBAC Approach

The process of role production is based on the connections between UML and URBAC, described in section 5. It can be automatic or partially automatic. Two types of UML diagrams were used to realize this process: use case diagrams, where roles and functions of a system are defined and sequence diagrams, where permissions are assigned to the rights of method executions realized in framework of each use case. These two types of diagrams should be examined to identify the roles of URBAC, the functions that are used by these roles to interact with the information system, the permissions needed to realize these functions and the constraints that determine the possible rights.

To obtain these elements of URBAC, first the rules for creation of set of roles has to be defined.

Each subject (i.e. user or group of users) in an information system is assigned to a security profile (i.e. user profile) which is defined by the set of roles that can be played by him. A security profile is defined by a pair $(s, \text{listRoles}(s))$: s is a subject, $\text{listRoles}(s)$ is a set of roles assigned to this subject. Taking into consideration the concept of user, such profile can be defined as follows: $(u, \text{listRoles}(u))$, where u is a user, $\text{listRoles}(u)$.

The process of creation of user profiles, i.e. production of set of roles, in information system with the use of UML diagrams contains two stages [7]:

Determination of a Function with Assigned Permissions

As it was shown in section 5, a use case of UML meta-model corresponds to a function of URBAC model. Use cases define the system functionality or in other words the interactions and needs of system's users that cooperate with the system. Each use case should be defined by its scenario that defines the specification of the use case interaction in form of sequence of actions performed on the system's objects. It allows the definition of set of privileges for execution of different actions on the objects. Therefore, in order to identify the permissions assigned to a function it is necessary to start from the sequence diagram corresponding to the function.

Each message $\text{msg}(o_1, o_2, m)$ in the interaction sent by object o_1 to object o_2 to execute method m on object o_2 should have a suitable permission assigned. This permission corresponds to the execution of method m on object o_2 . On the addition of security constraints, the definition of the message is as follows: $\text{msg}(o_1, o_2, m, \text{cst})$ where cst represents a set of constraints - authorizations, obligations and conditions: $\text{cst}(p) = A(p) \cup B(p) \cup C(p)$

Consequently, the set of permissions for interaction i is defined as follows:

$$P(i) = \{p \mid \varphi(\text{msg}(o_1, o_2, m, \text{cst})) = p(m, o_2) \wedge \text{cst}(p) = \text{true}\}$$

where φ is a function that assigns a permission to message msg , o_1 is an actor or class instance that can execute method m on object o_2 and cst is a set of constraints

$$\text{cst}(p) = A(p_{m,o_2}) \cup B(p_{m,o_2}) \cup C(p_{m,o_2}).$$

Sequence diagram in UML meta-model is defined by the set of interactions. Therefore, the set of permissions determined for sequence diagram d_S and described by the set of interactions D_S is as follows:

$$P(d_S) = \bigcup_{i \in D_S} P(i)$$

The use case μ described by set M of interaction diagrams d_i has the following set of permissions assigned to it:

$$P(\mu) = \bigcup_{d_i \in M} P(d_i)$$

The set $P(\mu)$ represents the set of permissions assigned to the function specified by the use case μ .

Determination of a Role with Assigned Functions

The use case diagram presents the system's functionality from the point of view of the actors. It is possible to find the set of use cases (i.e. URBAC functions) for each actor (i.e. URBAC role) examining this type of UML diagrams. Therefore, the determination of a role with the set of functions assigned to it will be realized by examining the relationships between the actors and use cases on the use case diagram. The use case diagram ucd_i contains the use cases (i.e. functions) attached to chosen actors (i.e. roles). The set of functions assigned to role r_j , described by one use case diagram, is defined by the functions that are in direct or indirect relations with this role (i.e. by the inheritance relations between the functions) on this diagram:

$$F(r_{ucd_i}) = \{f \mid f = uc, uc \in ucd_i \wedge (r_{ucd_i}, f) \in R - F\}$$

$$\cup \{f' \mid f' = uc, uc \in ucd_i \wedge ((f, f') \in F - F \wedge (r_{ucd_i}, f) \in R - F)\}$$

The set of functions of role r_j is defined by the union of use cases assigned to this role in all use case diagrams describing the whole system application D_{uc} :

$$F(r_j) = \bigcup_{ucd_i \in D_{uc}} F(r_{ucd_i})$$

In order to define the security profiles for system users or groups of users, the set of roles should be assigned to subject profiles (i.e. user profiles). This task is realized by security administrator during the exploitation stage who has to take into consideration the security constraints defined on the global level and the subject attributes defined for the subjects that determine the access control rights of particular system users.

7 Conclusion

The concepts of access control approach presented in the paper were used to define the process of role engineering for creation of security profiles for users of information system. The paper presents the representation of URBAC using the UML concepts, the process of roles production based on URBAC, the stages of creation of user profiles. The process of role production is a very important stage in definition of logical security policy of an information system. It can be realized by two actors: application developer and security administrator who cooperate with each other to guarantee the global coherence on access control level.

The aspects of presented approach are implemented on software platform that provides with the software tool to manage the logical security of company information system from the point of view of application developer and from the point of view of security administrator. Our next research is concentrated on aspects of security constraints for URBAC approach and on the algorithm to maintain the coherence of URABC scheme during the addition of new application.

References

1. Ferraiolo, D., Sandhu, R.S., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed NIST Role-Based Access control. ACM TISSEC (2001)
2. OMG Unified Modeling Language (OMG UML): Superstructure. Version 2.2, The Object Management Group (February 2009)
3. Park, J., Zhang, X., Sandhu, R.: Attribute Mutability in Usage Control. In: Farkas, C., Samarati, P. (eds.) *Data and Applications Security XVIII*. IFIP, vol. 144, pp. 15–29. Springer, Boston (2004)
4. Lazouski, A., Martinelli, F., Mori, P.: Usage control in computer security: A survey. *Computer Science Review* 4(2), 81–99 (2010)
5. Pretschner, A., Hilty, M., Basin, D.: Distributed usage control. *Communications of the ACM* 49(9) (September 2006)
6. Zhang, X., Parisi-Presicce, F., Sandhu, R., Park, J.: Formal Model and Policy Specification of Usage Control. *ACM TISSEC* 8(4), 351–387 (2005)
7. Goncalves, G., Poniszewska-Maranda, A.: Role engineering: from design to evaluation of security schemas. *Journal of Systems and Software* 81(8)
8. Poniszewska-Maranda, A.: Conception Approach of Access Control in Heterogeneous Information Systems using UML. *Journal of Telecommunication Systems* 45(2-3), 177–190 (2010)
9. Neumann, G., Strembeck, M.: A Scenario-driven Role Engineering Process for Functional RBAC Roles. In: *Proc. of 7th ACM SACMAT, USA* (June 2002)
10. Strembeck, M.: Scenario-Driven Role Engineering. *IEEE Security & Privacy* 8(1) (January/February 2010)
11. Strembeck, M., Neumann, G.: An Integrated Approach to Engineer and Enforce Context Constraints in RBAC Environments. *ACM TISSEC* 7(3) (2004)
12. Coyne, E.J., Davis, J.M.: *Role Engineering for Enterprise Security Management*. Artech House (2008)
13. Bertino, E., Ferrari, E., Atluri, V.: The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. *ACM TISSEC* 2(1)
14. Fernandez, E.B., Hawkins, J.C.: Determining Role Rights from Use Cases. In: *Proc. of 2nd ACM Workshop on Role-Based Access Control (RBAC), USA* (1997)
15. Basin, D., Doser, J., Lodderstedt, T.: Model driven security: From UML models to access control infrastructures. *ACM Transactions on Software Engineering Methodology* 15, 39–91 (2006)
16. Coyne, E.J.: Role engineering. In: *Proc. of the ACM Workshop on Role-Based Access Control* (1996)
17. Epstein, P., Sandhu, R.: Towards a UML Based Approach to Role Engineering. In: *Proc. of the ACM Workshop on Role-Based Access Control* (1999)
18. Epstein, P., Sandhu, R.: Engineering of Role-Permission Assignment to Role Engineering. In: *Proc. of 17th ACSAC* (2001)
19. Roeckle, H., Schimpf, G., Weidinger, R.: Process-oriented approach for role-finding to implement Role-based security administration in a large industrial organization. In: *Proc. of ACM Workshop on role-Based Access Control* (2000)
20. Poniszewska-Maranda, A.: Implementation of Access Control Model for Distributed Information Systems Using Usage Control. In: Bouvry, P., Kłopotek, M.A., Leprévost, F., Marciniak, M., Mykowiecka, A., Rybiński, H. (eds.) *SIIS 2011*. LNCS, vol. 7053, pp. 54–67. Springer, Heidelberg (2012)
21. Poniszewska-Maranda, A.: Administration of access control in information systems using URBAC model. *Journal of Applied Computer Science* 19(2) (2011)

A New Algorithm for Rotation Detection in Iris Pattern Recognition

Krzysztof Misztal^{1,2}, Jacek Tabor², and Khalid Saeed¹

¹ AGH University of Science and Technology
Faculty of Physics and Applied Computer Science
al. A. Mickiewicza 30, 30-059 Kraków, Poland
Krzysztof.Misztal@fis.agh.edu.pl, saeed@agh.edu.pl

² Jagiellonian University
Faculty of Mathematics and Computer Science
Łojasiewicza 6, 30-348 Kraków, Poland
tabor@ii.uj.edu.pl

Abstract. A new method for finding the rotation angle in iris images for biometric identification is presented in this paper. The proposed approach is based on Fourier descriptors analysis and algebraic properties of vector rotation in complex space.

Keywords: Iris pattern recognition, rotation estimation, rotation recovery, Fourier descriptors.

1 Introduction

The iris is an important candidate for the source of the unique human population characteristics. It is very stable over time – the iris is determined during the first years of our lives and does not change until the death. Moreover, the study shows the iris structure is minimally dependent on our genes and allows to identify even identical twins. In addition, the modern iris biometrics systems are fast and have a high accuracy verification.

In light of the above, it seems that the iris is ideal for biometrics. However, the iris recognition process is complex and the construction of a complete system requires addressing a number of problems. The current approaches to iris pattern recognition include: the Gabor wavelet approach by Daugman [2], the Laplacian parameter approach by Wildes et al. [9], zero-crossings of the wavelet transform at various resolution levels by Boles et al. [1], the Independent Component Analysis approach by Huang et al. [6], the texture analysis using multi-channel Gabor filtering and wavelet transform by Zhu et al. [10], and many others ideas. Each method has its own advantages and disadvantages.

Let us focus our attention on the classical John Daugman algorithm [3] – the most popular approach to individual identification by iris pattern recognition. and discuss how the algorithm copes with rotations. Let us start with the explanation and role of iris rotation angle. Fig. 1 present rotated and non-rotated iris images. When capturing iris images, most systems seek to achieve the same

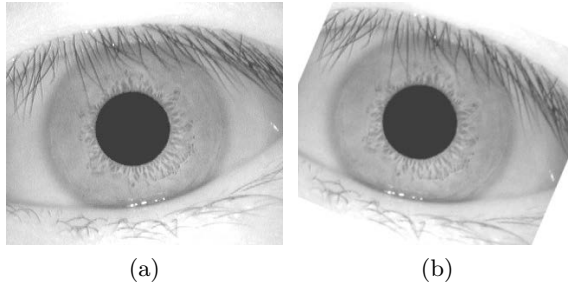


Fig. 1. Iris images: original (non-rotated) and rotated (23 degree)

environmental conditions and the position of the face. Under some assumed restrictions the obtained images provide the highest reliability of identification methods. However, these methods are not comfortable for the users. The system proposed by Daugman detects the rotation angle by "brute force", i.e., by testing all accessible angles and selects the best of them. However, such a way, as we will show later is not optimal in practice. Our aim in this paper is to present alternative approach to this method. The method determines the rotation angle directly. Based on this knowledge we hope to develop a more efficient algorithm to identify individuals.

In the first part of our work we briefly describe the classical Daugman algorithm. In the next chapter we will focus on the problem of detecting the rotation angle of the iris image. Then we describe the rotation angle estimation algorithm using Fourier transform. Further considerations will include construction of our algorithm. The last part of the work contains numerical experiments that demonstrate how our rotation angle estimation algorithm is relatively insensitive to both the eye and the camera:

- rotation,
- contrast modification,
- illumination level modification,
- blurring,
- sharpening.

This makes it suitable to be used even in the first stage of preprocessing images for the algorithm of individual verification.

2 Iris Code Matching Algorithms Survey

In this section we describe the main stages in the iris pattern recognition algorithm proposed by John Daugman [3].

After positive iris detection on an image and automatic segmentation we can match the iris region. To simplify, we assume we can describe the iris using only two circles – the arcs of the upper and lower eyelids are invisible on the surface

of the iris. The next step in the iris pattern recognition algorithm performs the normalization process, which includes conversion of the iris annulus into the rectangular sized fixed image.

In general, images standardization is the initial processing procedure which allows their mutual comparison and further analysis. Comparing methods use predetermined patterns of the iris therefore the iris images could be compared with patterns from the database. In the develop of authors' method, we relied on the method of the normalization, called Daugman's Rubber Sheet Model. This

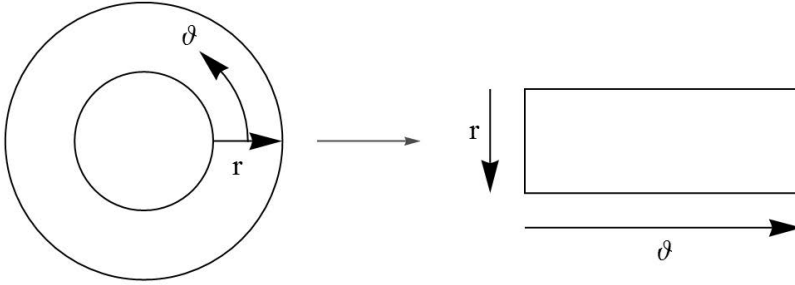


Fig. 2. Daugman's Rubber Sheet Model

model maps each point inside the area of the iris with a pair of polar coordinates (r, θ) , where $r \in [0, 1]$ and $\theta \in [0, 2\pi]$ (see Fig. 2).

Afterwards, we are able to introduce the feature encoding methods. In order to describe and recognize large individuals population only the significant features must be encoded. In Daugman's algorithm Gabor filters are used to obtain an optimal coding. As an outcome a phase sequence is obtained, 2048 bytes of data to describe the phase characteristics of the iris in a polar coordinate system – IrisCode. It is not affected by contrast, camera gain or illumination levels, so it is common in iris pattern recognition. Our method also has such properties.

For matching, the Hamming distance is chosen as a metric for recognition. The Hamming distance is calculated using only the bits generated from the true iris region, and this modified Hamming distance formula is given as

$$HD = \frac{\|(\text{code A} \oplus \text{code B}) \cap \text{mask A} \cap \text{mask B}\|}{\|\text{mask A} \cap \text{mask B}\|}$$

where codes A and B are the two bit-wise templates to compare, mask A and mask B are the corresponding noise masks.

In theory, Hamming distance of two iris templates generated from the same iris equals 0.0. However, since the normalization is not perfect this will not occur in practice. We obtain some noise that goes undetected, and hence a difference will always appear.

2.1 Daugman's Approach – Brute Force Matching

The following method, suggested by Daugman, corrects the misalignment in the normalized iris pattern. In order to cope with rotational inconsistencies, when the Hamming distance of two templates is calculated, one template is shifted to one bit left and right. Then the Hamming distance values are calculated. This bit offset in the horizontal direction corresponds to the primary market area of the iris angle indicated by the Rubber Sheet Model. As for the iris distance we choose only the lowest, which corresponds to the best match between two templates.

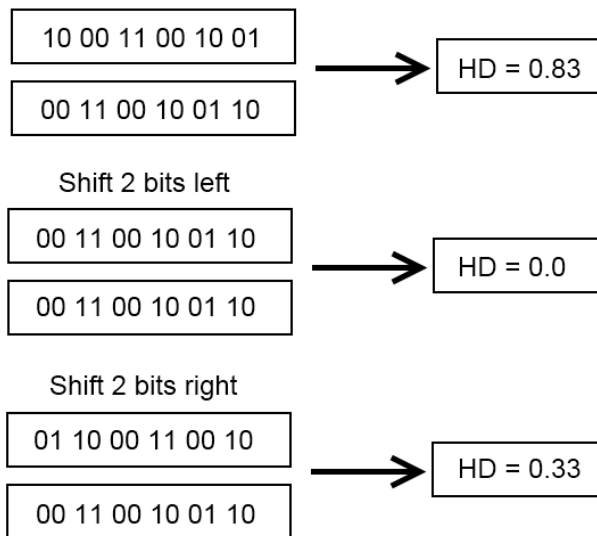


Fig. 3. The shifting process for one shift left and right

The number of bits transferred during each shift by two times the number of filters used, since each filter will generate two bits of information from one pixel normalized region. The real number of changes necessary to normalize rotational inconsistencies will be determined by the maximum angle difference between two pictures of the same eye. One change is defined as one shift to the left, then one shift to the right. The example of shifting process is presented in Fig. 3.

In practice, when we configure the algorithm, we fix the maximum number of shifts to the left and to the right. This brute force rotation angle estimation by comparisons is made with each element in the database – this is the main disadvantage of this method.

2.2 Authors' Approach: Fourier Descriptor Extended Method – Angle Calculation

Discrete Fourier Transform (DFT). For convenience of the reader and to established notation we recall the basic definitions of DFT. DFT transforms $(c_0, \dots, c_{N-1}) \in \mathbb{C}^n$ into a sequence (C_0, \dots, C_{N-1}) of complex numbers using the frequency domain representation:

$$C_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} c_n \exp \left(-i \frac{2\pi n k}{N} \right), \quad 0 \leq k \leq N-1,$$

where $c_n = a_n + ib_n$, for $k = 0, \dots, N-1$. This representation is widely used in data compression [7], partial differential equations, data mining, etc.

It is easy to see that

$$C_0 = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} c_n$$

and

$$C_1 = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} c_n \exp \left(-i \frac{2\pi n}{N} \right). \quad (1)$$

Remark 1. Fourier descriptors can be used to capture the main details of a boundary. This property is valuable because the coefficients usually keep the shape information. Thus they can be used as the basis for differentiating between distinct boundary shapes [5].

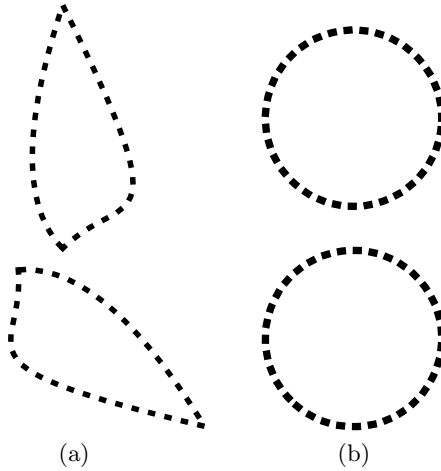


Fig. 4. Example of two shapes with digital boundary. The rotation of the second one does not allow to estimate angle of rotation.

This allows recovering the rotation for images with ordinary boundaries (see Fig. 4). However, it is useless in the case of rotation of the circles. Moreover, it can be affected by image modification like contrast correction, gamma correction, blurring, etc.

The Fourier Descriptor Approach to Rotation Detection in Iris Pattern Recognition. In this section we present how to recover the rotation angle using the modified idea of Fourier transform with vector angle calculation. The method from the above remark is not suitable in this situation, because the iris is represented as concentric circles and their rotations cannot be detected by boundary positions testing.

Assume that after using Daugman's Rubber Sheet Model we obtain two normalized iris I and I_R for original and rotated images respectively. Images are assumed to be in gray-scale and each pixel in the image is coded by a number from the set $\{0, \dots, 255\}$. Next, we resize the images to obtain the same size, for example we set the new size to 8×256 (image height \times width).

As an outcome we obtain two matrices

$$\begin{aligned}\tilde{I} &= (c_{kn})_{k,n} \in M_{8 \times 256}(\mathbb{R}), \\ \tilde{I}_R &= (c_{kn}^R)_{k,n} \in M_{8 \times 256}(\mathbb{R}).\end{aligned}$$

Next, we construct the vectors of features describing \tilde{I} and \tilde{I}_R by Fourier descriptors. For each row of these matrices we calculate the first Fourier descriptor according to equation (II)

$$\begin{aligned}C_k &= \frac{1}{16} \sum_{n=0}^{255} c_{kn} \exp\left(-i \frac{\pi n}{128}\right), \text{ for } k = 1, \dots, 8, \\ C_k^R &= \frac{1}{16} \sum_{n=0}^{255} c_{kn}^R \exp\left(-i \frac{\pi n}{128}\right), \text{ for } k = 1, \dots, 8.\end{aligned}$$

We put

$$x = (C_1, \dots, C_8), \quad x^R = (C_1^R, \dots, C_8^R).$$

Under the considerations of subsection about Fourier Descriptor Method, we get that those vectors are in the form $x = e^{i\varphi}v$, $x^R = w$, for some $v, w \in \mathbb{C}^8$.

Remark 2. The problem of finding the angle of rotation in the iris images can be formulated mathematically as follows.

Let non-trivial vectors $v, w \in \mathbb{C}^8$ be fixed. We are looking for the minimum of the function f :

$$f_{v,w}: [0, 2\pi] \ni \varphi \rightarrow \|e^{i\varphi}v - w\|^2 \in \mathbb{R}_+.$$

These functions describe the rotation one of the vectors relatively to the second one. We want to reduce the vectors to the same one-dimensional linear subspace. This will reduce the difference between them.

After simple transformation we get

$$\begin{aligned} f_{v,w}(\varphi) &= \|v\|^2 + \|w\|^2 - (e^{i\varphi}\langle v, w \rangle + \overline{e^{i\varphi}\langle v, w \rangle}) \\ &= \|v\|^2 + \|w\|^2 - 2\Re(e^{i\varphi}\langle v, w \rangle). \end{aligned}$$

It is easy to see that the minimum is reached when the value of

$$\Re(e^{i\varphi}\langle v, w \rangle)$$

is the largest. A trivial observation shows that

$$\Re(e^{i\varphi}\langle v, w \rangle) \leq |e^{i\varphi}\langle v, w \rangle| = |\langle v, w \rangle|, \text{ for } \varphi \in [0, 2\pi].$$

Consequently, we get $e^{i\varphi} = \overline{\langle v, w \rangle} / |\langle v, w \rangle|$, since $z \cdot \bar{z} = |z|^2$ for $z \in \mathbb{C}$ and $|e^{i\varphi}| = 1$. Thus the minimum of the function f is reached for

$$\varphi = \text{Arg} \frac{\overline{\langle v, w \rangle}}{|\langle v, w \rangle|}. \quad (2)$$

Thus, with a description by a vector of features, we can find the angle by which the iris has been rotated.

Therefore, we can apply observation from previous section and by equation (2) we obtain

$$\varphi = \text{Arg} \frac{\overline{\langle x, x^R \rangle}}{|\langle x, x^R \rangle|}, \quad (3)$$

where φ is the desired angle.

Summarizing, our approach to the problem of rotation angle estimation is stated as follows:

1. perform Daugman's Rubber Sheet Model standardization for I and I_R (we use rubber sheets of 8×256);
2. calculate Fourier Transform for each row of rubber sheets;
3. select the first Fourier descriptor from each rubber's row and build the vectors x and x^R ;
4. estimate angle of rotation using equation (3);
5. compare irises by Hamming distance using estimated angle.

The above steps link properties of the Fourier transform and the algebraic calculations. The computation is based on simple mathematical operations which make it easy for implementation.

3 Application in Iris Matching

In the last part of this work we present the comparison of classical Daugman's shifting method with our algorithm. Moreover, we make some experiments with noising and other "destruction" of information on iris images (ex. blur, gamma correction, sharpen) and examine the algorithm under such circumstances [8].

We prepare implementation of Daugman's iris recognition algorithm based on [3] and [2]. Besides we extend this application by adding the implementation of our method.

3.1 Comparison between Classical and Authors' Method in Rotation Angle Detection

Firstly, we compare the efficiency of shifting method with our method. Table 1 contains comparison between the classical method and ours.

We evaluate IrisCode for rotated (for angle $0, \dots, 9$ degrees) and non-rotated images. Then we run shifting method to see what angle (column Rec.) was chosen and what the value of Hamming distance (column HD) is. In this case IrisCode evaluated for rotated iris image was compared with shifted original IrisCode in maximum 10 shifts. Next, we use our method which estimates rotation angle

Table 1. Comparison of angle detection in classical methods with shifts and authors' method

Angle	Classical		Authors'	
	Rec.	HD	Rec.	HD
0	0	0	0.0	0
1	0	0	0.0	0
2	1	0	1.4	0
3	3	0	2.8	0
4	3	0	2.8	0
5	4	0	4.2	0
6	6	0	5.6	0
7	6	0	5.6	0
8	7	0	7.0	0
9	8	0	8.4	0

(column Rec.), and then the estimated angle was rounded to the nearest integer and used to generate shifted IrisCode. The obtained distance is presented in column HD.

The experiment results show that our method gives the same angle as the brute force method, with the advantage that calculations were performed faster.

3.2 Sensitivity to Image Modifications

During the second experiment we want to check if our method is insensitive to image destruction. Fig. 5 presents images used in this experiment. But before we present experiment details let us put some notations.

Let I denote image of size $w \times h$ pixels (width \times height) – it is a matrix, so $I \in \{0, \dots, 255\}^{w \times h}$ for gray-scale image. Then $I(x, y)$ for $x \in \{1, \dots, w\}$, $y \in \{1, \dots, h\}$ denotes colors value at position (x, y) on image. By I_N we denote the new image (output image) created by modifying the input image I .

We can now proceed in explaining the original image (Fig. 5a) modification descriptions. We use the following methods:

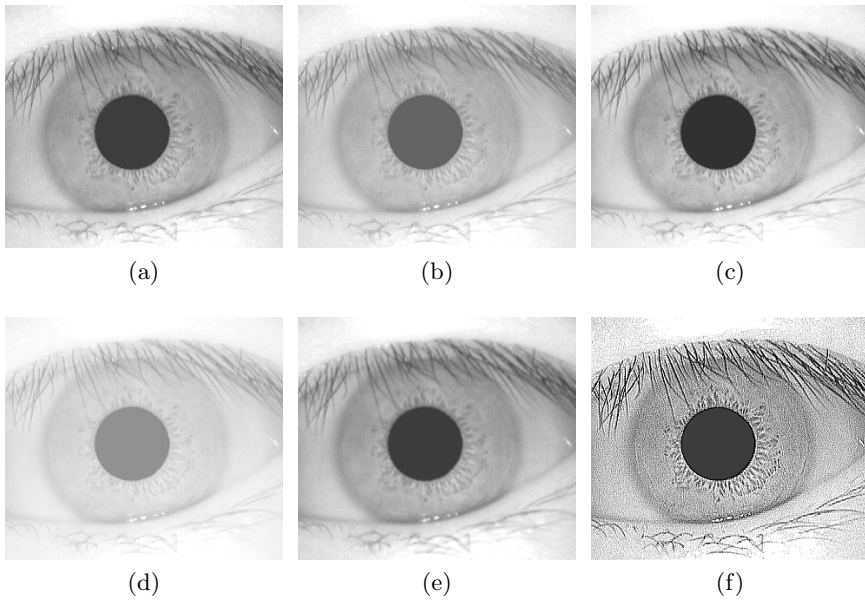


Fig. 5. Iris image modified by different methods: **5a** – original image; **5b** – brightness correction (+50); **5c** – contrast correction (+30); **5d** – gamma correction (+2.5); **5e** – blur; **5f** – sharpen

- brightness correction (level $\delta = +50$) (Fig. **5b**) – this method removes shadows without affecting the rest of the image, by

$$I_N(x, y) = \min(255, I(x, y) + \delta) \text{ for } x \in \{1, \dots, w\}, y \in \{1, \dots, h\},$$

- contrast correction (level $\alpha = +30$) (Fig. **5c**) – this method changes luminance on image and makes objects distinguishable

$$I_N(x, y) = \min \left(255, \max \left(0, \left\lceil \frac{255}{255 - \alpha} (I(x, y) - 127) + 127 \right\rceil \right) \right),$$

$$\text{for } x \in \{1, \dots, w\}, y \in \{1, \dots, h\}$$

- gamma correction (level $\gamma = 2.5$) (Fig. **5d**) – it is nonlinear modification of brightness

$$I_N(x, y) = \left\lceil 255 \left(\frac{I(x, y)}{255} \right)^\gamma \right\rceil \text{ for } x \in \{1, \dots, w\}, y \in \{1, \dots, h\},$$

- blur (Fig. **5e**) – it is a convolution filter with mask

$$B = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix},$$

used to reduce noise and details information on image. In this case linear filtering with a mask of size 3×3 is given by the expression

$$I_N(x, y) = \sum_{s=-1}^1 \sum_{t=-1}^1 B(s, t) I(x + s, y + t).$$

To generate full filtered image this equation should be applied for $x \in \{1, \dots, w\}$ and $y \in \{1, \dots, h\}$ (see [5]).

- sharpen (Fig. 5d) – it is a convolution filter with a mask

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}.$$

Sharpens the image making it look crisper and making the edges in the image more distinct.

Table 2. Angle detection in different image modifications (angle is given in degrees)

Method	Angle									
	5		14		23		47		76	
	Rec.	Error	Rec.	Error	Rec.	Error	Rec.	Error	Rec.	Error
None	4.22	0.78	12.66	1.34	22.50	0.50	46.41	0.59	75.94	0.06
Brightness	4.22	0.78	12.66	1.34	22.51	0.49	46.41	0.59	75.94	0.06
Contrast	3.99	1.01	12.43	1.57	22.27	0.73	46.18	0.82	75.71	0.29
Gamma	3.98	1.02	12.41	1.59	22.26	0.74	46.16	0.84	75.70	0.30
Blur	3.69	1.31	11.99	2.01	21.87	1.73	45.69	1.31	75.13	0.87
Sharpen	3.66	1.34	13.72	0.28	23.10	0.10	47.92	0.92	77.83	1.83

Table 3. Angle detection in different image modifications (angle is given in degrees) obtained by classical shifting

Method	Angle									
	5		14		23		47		76	
	Rec.	Error	Rec.	Error	Rec.	Error	Rec.	Error	Rec.	Error
None	5	0	14	0	23	0	47	0	76	0
Brightness	0	5	10	4	0	23	38	9	8	68
Contrast	0	5	10	4	0	23	38	9	8	68
Gamma	0	5	11	3	1	22	38	9	8	68
Blur	0	5	13	2	3	20	60	13	30	46
Sharpen	0	5	10	4	0	23	38	9	8	68

Table 2 summarizes the values of estimated angles with error level obtained by our method. Table 3 contains angles recovered by classical Daugman method. The error level in both examples suggests that our method is not affected by the proposed image factors.

4 Conclusions

The iris identification and verification process is complex and the construction of a complete system requires addressing a number of problems. One of the most important and most difficult ones is that of detecting the rotation angle of the iris image. The authors have introduced a new method to solve this problem based on simple mathematical operations which have made it easy for implementation. Fourier transform and some algebraic calculations are utilized to work out an algorithm of high success rate whose performance is presented in this work. It has described and shown the rotation angle estimation using Fourier transform. The numerical experiments have demonstrated how the authors algorithm is relatively insensitive to both the eye and the camera relative to rotation, contrast changes, illumination level and other factors that usually affect the image under processing.

Acknowledgment. The work is supported by AGH University of Science and Technology, Krakow (grant no. 11.11-220-01).

References

1. Boles, W.W., Boashash, B.: A human identification technique using images of the iris and wavelet transform. *IEEE Transactions on Signal Processing* 46(4), 1185–1188 (1998)
2. Daugman, J.: High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(11), 1148–1161 (1993)
3. Daugman, J.: How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology* 14(1), 21–30 (2004)
4. Elshoura, S.M., Megherbi, D.B.: A new rotation estimation and recovery algorithm. In: 2010 International Symposium on Computer Communication Control and Automation (3CA), vol. 1, pp. 411–414 (2010)
5. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*. Prentice Hall, Upper Saddle River (2002)
6. Huang, Y.P., Luo, S.W., Chen, E.Y.: An efficient iris recognition system. In: *Proceedings of the 2002 International Conference on Machine Learning and Cybernetics*, vol. 1, pp. 450–454 (2002)
7. Salomon, D.: *Data compression: the complete reference*. Springer-Verlag New York Inc. (2007)
8. Tomeo-Reyes, I., Liu-Jimenez, J., Rubio-Polo, I., Redondo-Justo, J., Sanchez-Reillo, R.: Input images in iris recognition systems: A case study. In: 2011 IEEE International Systems Conference (SysCon), pp. 501–505 (2011)
9. Wildes, R.P.: Iris recognition: an emerging biometric technology. *Proceedings of the IEEE* 85(9), 1348–1363 (1997)
10. Zhu, Y., Tan, T., Wang, Y.: Biometric personal identification based on iris patterns. In: *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 2, pp. 801–804 (2000)

Outlier Removal in 2D Leap Frog Algorithm

Ryszard Kozera^{1,2} and Jacek Tchórzewski

¹ Faculty of Mathematics and Information Science
Warsaw University of Technology
00-661 Pl. Politechniki 1, Warsaw, Poland

² Faculty of Applied Informatics and Mathematics
Warsaw University of Life Sciences - SGGW
02-776 Nowoursynowska 159, Warsaw, Poland

r.kozera@mini.pw.edu.pl, tchorzewski.jacek@gmail.com

Abstract. In this paper a 2D Leap Frog Algorithm is applied to solve the so-called *noisy Photometric Stereo* problem. In 3-source Photometric Stereo (noiseless or noisy) an ideal unknown Lambertian surface is illuminated from distant light-source directions (their directions are assumed to be linearly independent). The subsequent goal, given three images is to reconstruct the illuminated object's shape. Ultimately, in the presence of noise, this problem leads to a highly non-linear optimization task with the corresponding cost function having a large number of independent variables. One method to solve it is *2D Leap Frog Algorithm*. During reconstruction, problem that commonly arises, renders the *outliers* generated in the retrieved shape. In this paper we implement 2D Leap Frog. In particular we focus on choosing snapshot size and on invoking two algorithms that can remove outliers from reconstructed shape. Performance of extended 2D Leap Frog is illustrated by examples chosen especially to demonstrate how this solution is applicable in computer vision. Remarkably, this optimization scheme can also be used for an arbitrary optimization problem depending on large number of variables.

Keywords: Shape from Shading, noise removal, optimization, outliers.

1 Introduction

One of the most important problems in the field of computer vision is an issue of shape reconstruction from its image(s) data. There are two main approaches to tackle this problem. One method (discussed in this article) is based on surface S reconstruction from the data obtained from its image(s) created by single camera (the so-called *Shape from Shading* [4]). The second class of techniques is the shape reconstruction based on multiple camera input data and the usage of *triangulation like methods* [5]. In the first method single illumination yields the so-called single image Shape-from-Shading problem. On the other hand a multiple illumination in Shape from Shading is called Photometric Stereo (see e.g. Horn [3], Kozera [8] and Noakes and Kozera [9]). As opposed to the ideal

continuous setting in Shape from Shading (see e.g. [4]) an additional problem arises when noise is added to the image(s). This forms noisy Photometric Stereo problem (or noisy single image Shape from Shading problem, respectively). As it turns out it is modeled by the corresponding highly non-linear optimization task in many multiple variables. One of the computational techniques striving to deal with such optimization is a 2D Leap Frog Algorithm [6] and [9], that is recalled and modified in this paper.

While dealing with classical Shape from Shading, one seeks a function $u : \Omega \subseteq \mathbb{R}^2 \mapsto \mathbb{R}$, that represents distance of surface point $(x, y, u(x, y)) \in \text{graph}(u) = S$ from a camera (where Ω is a picture within the camera). For a Lambertian surface (which is a perfect diffuser) with a constant albedo, illuminated from distant light-source direction $p = (p_1, p_2, p_3)$ the image irradiance equation in the *continuous setting* is given over Ω as (see Horn [3]):

$$E_p(x, y) = \frac{p_1 u_x(x, y) + p_2 u_y(x, y) - p_3}{\sqrt{p_1^2 + p_2^2 + p_3^2} \sqrt{u_x^2(x, y) + u_y^2(x, y) + 1}}. \quad (1)$$

For the remaining two light-source directions $q = (q_1, q_2, q_3)$ and $r = (r_1, r_2, r_3)$, similar image irradiance equations can be derived. Note that we urge $\det(p, q, r) \neq 0$ (i.e. light-source directions are to be linearly independent).

In the case of *discrete model* (e.g. with noise added) a pertinent function is to be minimized, later called a cost function \mathcal{E} . It is derived from the physical concepts of Photometric Stereo and from the properties of a Lambertian surface. In real case our Ω is not continuous but discrete as it addresses the image(s) represented by a collection of pixels. Assume that the number of all image pixels is n^2 . The general formula for such cost function (by [1]) reads (see Noakes and Kozera [9]):

$$\mathcal{E}(u) = \sum_{i,j=2}^{n-1} \mathcal{E}_{i,j}(u), \quad (2)$$

where $u \in \mathbb{R}^{n^2-4}$ and (i, j) -pixel energy value $\mathcal{E}_{i,j} = \mathcal{E}_{i,j}^p + \mathcal{E}_{i,j}^r + \mathcal{E}_{i,j}^q$ with:

$$\mathcal{E}_{i,j}^p(u) = \left(\frac{p_1 \left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} \right) + p_2 \left(\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} \right) - p_3}{\|p\| \sqrt{\left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} \right)^2 + \left(\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} \right)^2 + 1}} - E_p(x_i, y_j) \right)^2, \quad (3)$$

where $\|p\| = \sqrt{p_1^2 + p_2^2 + p_3^2}$, and $\mathcal{E}_{i,j}^r$ and $\mathcal{E}_{i,j}^q$ are defined similarly to [3]. Note that [3] is a discretized version of image irradiance equation [1] at internal image pixel point (i, j) upon using central difference derivative approximations:

$$u_x(i, j) \approx \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}$$

and $u_y(x, y)$ is estimated similarly.

We look for discrete u which minimizes (2). At this moment it is clearly visible that there exists strong *nonlinearity* in formula (2). The cost function \mathcal{E} depends on $n^2 - 4$ variables (image corners do not participate in reconstruction) which represents almost entire image resolution (in practice a very big number). Thus the problem (2) usually forms an optimization task depending on a very large number of variables. Using here Newton's method based on inversion $D^2\mathcal{E}$ over entire Ω is a huge computational task.

2 2D Leap Frog Algorithm

The main idea standing behind the 2D Leap Frog Algorithm is to find the sub-optimal minima of a function \mathcal{E} (see (2)) by accumulating the results of the so-called local area optimizers. It is worth mentioning the difference between the terms *local area minimum* versus *local minimum* of \mathcal{E} . The difference is that the local area minimum is a suboptimal minimum for a cost function \mathcal{E} considered with the values u of over pixels taken from some local area $\Omega_{loc} \subseteq \Omega$. On the other hand the local minimum of \mathcal{E} is any sub-optimal solution to (2) over entire image Ω . Note that, according to formula (1) a visible part of the Lambertian surface corresponds to the case, where $0 \leq \cos \Theta \leq 1$, whereas the invisible one results when $-1 \leq \cos \Theta < 0$ (where Θ denotes the angle between the light-source direction and the surface normal at point $(x, y, u(x, y))$). In both cases, however the image irradiance equations can still be mathematically posed and solved (this also refers to (2)). Hence the corresponding optimization problem (2) can be tackled at least theoretically over entire Ω , which is rectangular (camera sensor). Thus for simplicity we assume and solve the optimization problem (2) under the constraint that the data over entire rectangular image are accessible. Let us denote an image space $\Omega = [0, 1] \times [0, 1]$ with given $(k; l) \in \mathbb{N}$ as length and height in pixels. The space Ω is now divided into atomic subspaces, described as follows:

$$S_{i_q j_q}^l = \left| \frac{i_q - 1}{2^l}, \frac{i_q}{2^l} \right| \times \left| \frac{j_q - 1}{2^l}, \frac{j_q}{2^l} \right|, \quad \text{for } 1 \leq i_q, j_q \leq 2^l.$$

The main part of the 2D Leap Frog Algorithm works only on these rectangular subspaces, that are each denoted by SQ_{ij}^{lm} , where i and j are the row and column indices, and l and m are the width and height of the subspace.

Assume now that $u_1^{k_1}, \dots, u_s^{k_1}$ represents some free variables of \mathcal{E} and the remaining $u_1^{k_2}, \dots, u_t^{k_2}$ are temporarily frozen (where $s, t \in \mathbb{N}$). Then our global energy \mathcal{E} can be split into two components:

$$\mathcal{E}(u) = \mathcal{E}_1(u_1^{k_1}, \dots, u_s^{k_1}) + \mathcal{E}_2(u_1^{k_2}, \dots, u_t^{k_2}). \quad (4)$$

Evidently minimizing \mathcal{E}_1 over variables $u_1^{k_1}, \dots, u_s^{k_1}$ only, does not change the value of \mathcal{E}_2 and therefore the whole value of \mathcal{E} is also decreased. If s is small then optimizing \mathcal{E}_1 becomes a computationally feasible task (e.g. for Newton's method). This is the main idea standing behind the 2D Leap Frog. Optimizing

\mathcal{E}_1 in our case corresponds to the optimization of \mathcal{E} over each SQ_{ij}^{lm} with all other values of u frozen. As shown by Noakes and Kozera [9] an iterative sequence of local area optimizations over different overlapping snapshot yields a suboptimal solution of \mathcal{E} over the entire image Ω .

2.1 2D Leap Frog Local Optimizer

There are nine different types of subspaces, also called grids (or *snapshots*), as shown in Fig. 1. These subspaces are called: *top-left*(1), *top*(2), *top-right*(3), *middle-left*(4), *middle*(5), *middle-right*(6), *bottom-left*(7), *bottom*(8), and *bottom-right*(9), respectively. Note that for synthetic images we can assume (as explained before) that image is rectangular. Otherwise for the real images (with shadows) nine cases have to be combined with the visible regions of images (left for future work). There are five types of pixels in each grid as follows:

1. *Type 0* – pixels that are only used to preserve the rectangular shape of a grid, and not used for computation.
2. *Type 1* – pixels that are locked, and have constant value throughout the computation of the grid. These pixels are also excluded from the energy cost function computation. They are used to compute the central-difference derivative approximation for some pixels around them.
3. *Type 2* – these pixels are also locked, and are included in the energy function.
4. *Type 3* – these pixels are not locked, and are used as variables while searching for the minimum of the energy cost function. They are also included in the energy cost function.
5. *Type 4* – these pixels are neither locked nor included in the energy function.

The nine grid types can be further reduced to three main types: side grids (touching only 1 border), corner grids (touching 2 borders excluding top-left case) and mid grids (touching none of the borders). This is because grids of the

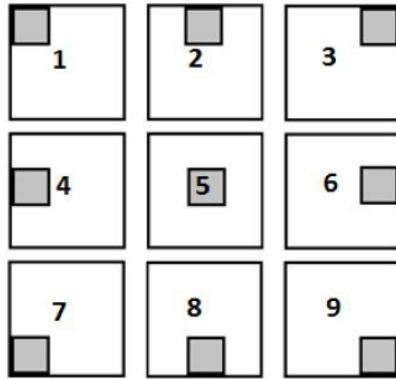


Fig. 1. All 9 types of snapshots

same type can be obtained from each other by rotation. Firstly, side grids are described as shown in Fig. 2.

The central pixels of the border type grid are of type 3. Type 4 pixels are found directly between the pixels of type 3 and the border. This is because the border pixels must be included in the overall energy function, but it is impossible to approximate their central difference derivatives, and therefore their energies cannot be computed. Type 3 pixels are also surrounded by pixels of type 2 because minimizing the cost function has an impact on their surrounding pixels due to their derivative approximation. Lastly, there are type 1 pixels surrounding all others pixels. They are used just to evaluate the derivatives of other pixels and have no further impact on the algorithm. Corner grids are localized next to two borders as shown in Fig. 2. Type 3 pixels are again in the middle of the grid with type 4 pixels situated between them and the borders. Type 2 pixels also surround type 3 pixels, and type 1 pixels surround everything. The only pixel worth mentioning is the corner pixel. This pixel needs to be of type 0 because there are no surrounding dependent pixels. This implies that the strict corners of the space never changes and remains at the initial guess.

Mid grids are the simplest cases as shown in Fig. 2. Type 4 pixels are not used because energy of all pixels in the grid can be calculated. There is a rectangle of type 3 pixels in the middle surrounded by type 2 and type 1 pixels. This case is straightforward and does not require any further explanation.

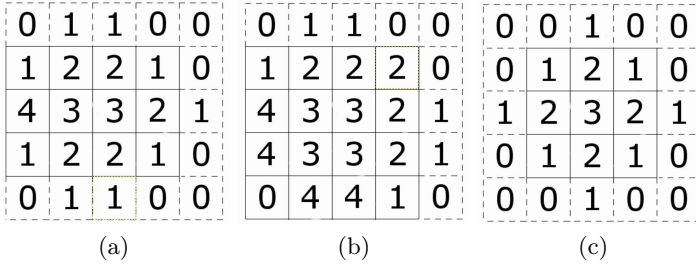


Fig. 2. Main three types of grids

2.2 2D Leap Frog Algorithm

The non-linear 2D Leap Frog Algorithm is constructed as follows:

1. Obtain an initial guess u_0 (e.g. using Lawn Mowing Algorithm [10]). For the need of our paper we added a large noise to the ideal continuous solution u .
2. Divide the space into grids of fixed size, overlapping in such a way that 3 columns and rows of pixels are common for grids that are neighbors.
3. Start first iteration of the algorithm with the grid on the left-top corner, and apply a minimization algorithm to obtain new values of the unlocked variables.
4. Move to the right neighbor of the previous grid. Apply minimization algorithm to the function created by the top grid case. Repeat this process until the right border is reached.

5. In the last grid of the current row, the right-top grid case is used for optimization.
6. Move to the row below. The first grid uses the left grid case for optimization.
7. Move to right neighboring grid and minimize using the mid grid case. Repeat until the right boundary is reached.
8. The last grid in this row is minimized using the right grid case.
9. Repeat 3 previous steps of the algorithm for all rows except the bottom one.
10. First grid of the last row uses the bottom-left grid case to minimize \mathcal{E} .
11. All consecutive grids except of the right most grid use the bottom grid case.
12. The last grid uses the right-bottom grid case.
13. Repeat all previous steps until one of the stopping condition is fulfilled.

In most cases the steps differ only in the type of grid cases that are used. 2D Leap Frog requires that, in every iteration, all pixels (except for the corner pixels) are unlocked at least once. Otherwise the algorithm does not work properly as some variables of the cost function \mathcal{E} never change. In addition this guarantees that Leap Frog converges to critical point of \mathcal{E} .

There are *three possible stopping criteria* for the 2D Leap Frog Algorithm:

1. *Timeout*: the algorithm computation stops upon exceeding a priori established time limit.
2. *Maximum iterations limit*: the algorithm has a cap on the number of iterations that is computed.
3. *Epsilon limit*: the global cost function is not decreasing anymore; this is the most difficult criterion possible in the algorithm because the iterations can have uneven decrease. For example there can be a slowdown in decrease for a few iterations in the middle of computation; therefore the best way of measuring the decrease is for at least four iterations. In this case if the decrease is close to zero for the chosen number of iterations, the algorithm stops.

2.3 Ambiguities and Relaxation

There are two kinds of ambiguities in our discrete settings namely, *standard* and *strong ambiguities*. We explain now the difference. Recall that three images are obtained from three linearly independent light-sources. With these data a system of three image irradiance equations is generated, and independent Gaussian noise is added to the corresponding images. Assume now that C denotes the table of constant entries c : which can be obtained by deleting corner values from the matrix of dimension $n \times n$. Clearly we have:

$$\mathcal{E}(u + C) = \mathcal{E}(u).$$

Therefore adding a constant value to the reconstructed surface in the discrete problem does not change the energy \mathcal{E} . This is called a *standard ambiguity in discrete case*. We demonstrate now, the so-called *strong ambiguity* which also

arises in the discretization of u . To see it let us take tables shown in Fig. 3. When adding these tables $u_i (i = 1, 2, 3, 4)$ to the reflectance maps, due to the central difference method, we also have:

$$\mathcal{E} \left(u + \sum_{1 \leq k \leq 4} \check{c}_k * u_k \right) = \mathcal{E}(u).$$

This phenomenon is called a *strong ambiguity*. Such scenario can be described in terms of finding the minima of a function of four variables. If the function u is shaped like a valley, with the lowest place on the same plane, it is impossible to find only one minimum, as the function has then same value along the valley. This creates the problem of having infinitely many solutions that have the same energy. The simplest solution is to freeze four pixels in positions: (1; 2); (1; 3); (2; 2); (2; 3). This guarantees that the central differences are uniquely determined (for proof see Noakes and Kozera [9]). When the algorithm finishes its work, these four pixels need relaxation, meaning that their minimum is found. It is important to add that, Leap Frog converges to sub-global solution (if initial noise is small then it converges to the global solution which is close to correct solution). For further details refer to Noakes and Kozera [9].

$$\begin{aligned}
 u_1 &= \begin{pmatrix} 1 & 0 & 1 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & \dots & 0 & 1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & \dots & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \end{pmatrix} & u_2 &= \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & \dots & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 & 1 & \dots & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & \dots & 1 & 0 & 1 & 0 \end{pmatrix} \\
 u_3 &= \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & \dots & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 1 & 0 & \dots & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & \dots & 0 & 1 & 0 & 1 \end{pmatrix} & u_4 &= \begin{pmatrix} 0 & 1 & 0 & 1 & \dots & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & \dots & 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & \dots & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \end{pmatrix}
 \end{aligned}$$

Fig. 3. Four tables showing strong ambiguities

2.4 Outlier Removal

In this section we discuss possible approaches to outliers removal. During reconstruction, the most unstable parts are the borders of the image. At the borders, continuity of the function ends. This pushes the pixels at the borders away from their proper values. An attempt is made, in this paper, using a statistical approach to remove these outliers. There are many efficient algorithms for *outlier*

removal [1], but they require a large sample of data to work correctly. We apply here a mask (i.e. a 3x3 pixels shifted over Ω) to identify potential outliers.

The *first approach* (called also *Algorithm 1*) used for outlier removal is based on Chauvenet's criterion [2]. This criterion states that any data that differ by more than two standard deviations from the mean of the data set is an outlier. The data set is created from the 9 closest non-border pixels to test one. Once an outlier is detected, it is replaced with the mean of the data set. The important feature of this test is that potential outliers are not considered as part of the sample (because if it is an outlier then it will disrupt the value of the mean). This test is done for all border pixels. Such approach unfortunately is not sufficient enough on its own (see Ex. 1).

The *second approach* (called *Algorithm 2*) is to modify 2D Leap Frog. The biggest problem with outliers was the starting edge of the image (the same for every iteration). At this point in the frame there are two borders that are unknown, and two borders that are not yet optimized. Later along this edge we have one unknown border, one optimized and two not optimized. This creates the most difficult local optimization problem in whole Ω . To remove this problem we decided that our implementation should start from different points for proceeding iterations. The change is done as follows:

1. The first iteration starts from left upper corner and proceed as described in previous paragraph.
2. Next start from right upper corner and proceed down from starting point.
3. Next start from right bottom corner and proceed left from starting point.
4. Next start from left bottom corner and proceed up from starting point.

3 Examples

This section demonstrates the difference between proposed outlier removal algorithms (i.e Algorithm 1 and 2). The last test shows also the performance of 2D Leap Frog Algorithm with respect to the size of frame. Newton's method is used here for each snapshot optimization. The Gaussian noise added to surface pictures is the same in all tests (i.e. with mean equal to 0 and deviation equal to 0.8).

3.1 Example 1

(i) The first surface S_1 is described by the graph of the function u_1 (see Fig. 4):

$$u_1(x, y) = \frac{(\cos x + \cos y)^3}{8000},$$

defined over the domain $[0, 10] \times [0, 10]$. The light-source directions used here are $p = (1, 2, 3)$, $q = (1, 8, 5)$ and $r = (4, 2, 4)$. Gaussian noise is generated with a mean equal to 0 and deviation equal to 0.2 and added to u_1 to obtain

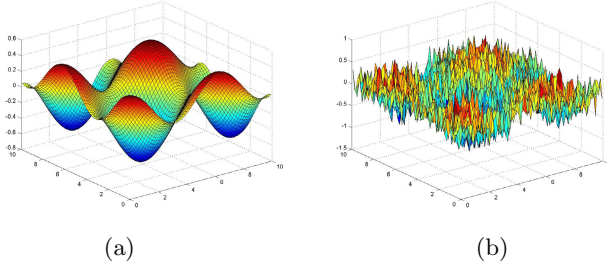


Fig. 4. (a) The ideal surface $S_1 = \text{graph}(u_1)$ and (b) the initial guess

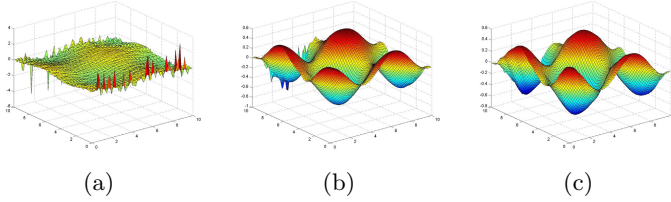


Fig. 5. Results for u_1 : (a) 2D Leap Frog without outlier removal (b) Algorithm 1 - with statistical approach and (c) Algorithm 1 and 2 with both approaches

initial guess. Therefore the noise is significant as compared to the base surface plot (see Fig. 4), having values within the range of -0.6 to 0.6 .

(ii) The second surface S_2 is described by the graph of the function u_2 (see Fig. 6):

$$u_2(x, y) = \cos(5 * x) + \sin(5 * y),$$

defined over the domain $[-0.5, 0.5] \times [-0.5, 0.5]$. The light-source directions used here are $p = (2, 2, 3)$, $q = (1, 3, 2)$ and $r = (6, 2, 4)$. Gaussian noise is generated with a mean equal to 0 and deviation equal to 0.5 (see Fig. 6) and added to u_2 to obtain initial guess. The base surface plot have values is within the range of -0.2 to 0.2 .

From those tests we can clearly see that our algorithm performs very good in terrain like surfaces. Also those tests shows perfectly all problems with outliers in surface reconstruction. In first case (see Fig. 5 and 7) two types of outliers were observed, ones that are on the edges of the reconstructed surface (type 2) and ones that are anywhere else (type 1). In outcome from second implementation (having outlier removal Algorithm 1 - see Fig. 5 and 7) we observe that we removed most of outliers from borders (only the strongest remain). This is because algorithm, using statistical approach, to work properly needs mean and standard deviation of the test area. Those values are taken for small area, and thus the strong outliers can disrupt them in such a manner that it will be possible to form them within the scope of accepted values. In the third approach i.e. using Algorithms 1 and 2 (see Fig. 5 and 7) we can clearly see that no big outliers are created during reconstruction process. Hence combination of Algorithms 1 and 2 for outlier removal succeeds.

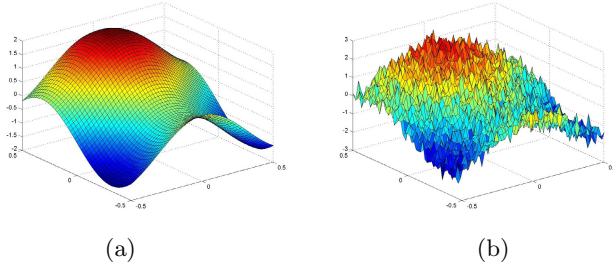


Fig. 6. (a) The ideal surface $S_1 = \text{graph}(u_2)$ and (b) the initial guess

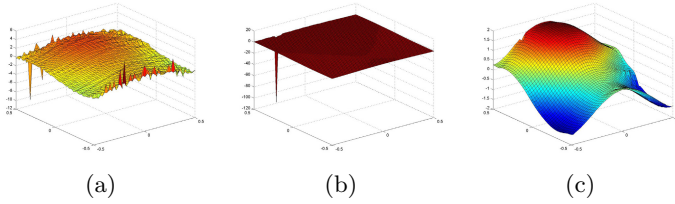


Fig. 7. Results for u_2 : (a) 2D Leap Frog without outlier removal (b) Algorithm 1 - with statistical approach and (c) Algorithm 1 and 2 with both approaches

3.2 Example 2

The surface S_3 chosen for this test is described by the graph of the function u_3 :

$$u_3(x, y) = \frac{1}{\left(1 - \tanh\left(\frac{25}{6(4-6x+3x^2-6y^2+3y^2)}\right)\right)},$$

defined over the domain $[0, 2] \times [0, 2]$. The light-source directions used here are $p = (2, 4, 3)$, $q = (3, 3, 2)$ and $r = (6, 2, 1)$. Gaussian noise is generated with a mean equal to 0 and deviation equal to 0.2 (see Fig. 8) and added to u_3 to obtain initial guess. The base surface plot have values is within the range of 0 to 0.8.

Table I shows that 6x6 is the optimum grid size for this implementation. This size provides the ideal ratio of free variables and the number of grids required to cover the image. The algorithm is run from the smallest to the largest possible grid sizes (5x5 to 8x8) for the implementation. At 9x9, containing 81 pixels, Matlab failed to run the Newton's Algorithm because there were too many unknowns in the function to minimize (about 50 unlocked variables). The next criterion is the quality of the reconstructed surface. Table 1 shows that smaller grid sizes produce reduced energies, therefore better surface reconstructions. This is because smaller functions are used to search for the local area minima. Thus the energy does not stabilize at a local minimum due to intensive grid overlaps. With bigger functions there is less change per iteration (as the whole picture is covered by fewer grids); therefore functions tend to end in its local minima, and do not change significantly in consecutive iterations.

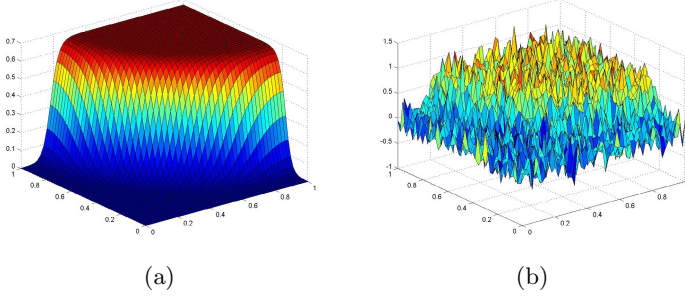


Fig. 8. (a) The ideal surface $S_1 = \text{graph}(u_3)$ and (b) the initial guess

Table 1. Time efficiency for different grid sizes (given in minutes)

	Time needed	Iterations needed	Time per iteration	Finishing energy
5 by 5	14.5	16	0.9	0.015174
6 by 6	12.1	10	1.21	0.019644
7 by 7	17.6	14	1.25	0.093630
8 by 8	20.3	18	1.72	0.132740

4 Conclusions

It should be emphasized that 2D Leap Frog Algorithm is a very *flexible and universal tool* that can be used not only in Photometric Stereo. Clearly this computational method (with or without our modifications) is applicable to any optimization problem which in particular suffers from a large number of free variables. We close this paper with the following observations:

- We tested Leap Frog under the assumption that the Lambertian model is satisfied, and that the character of the Gaussian noise is preserved. The first simplification assures that the examined reflectance is modeled by formula (1) which is well approximated by its discrete analogue (3) (at least for image(s) with high resolution).
- In addition minimizing the cost function (3) addresses the principle of maximum likelihood (see (11)) applied to Gaussian noise added at the image level (where normal random variables denotes measurements of image intensity of each pixel). Therefore under such ideal settings Leap Frog is tested. Of course, with real surfaces, where neither Lambertian model nor Gaussian noise are preserved Leap Frog can still be tested and compared in our experiments (this forms a potential further work).
- 2D Leap Frog Algorithm (combined with outliers removals: Algorithms 1 and 2) works very well in reconstructing different surfaces (in the presence of noise).

- Our our tests clearly demonstrate that most significant changes in the cost function \mathcal{E} (obtained for the whole surface S) are generated during the first 4-6 iterations. After that our cost function \mathcal{E} is marginally decremented. However, it is necessary for an algorithm in question to work further because during those consecutive iterations the biggest surface outliers are removed. Unfortunately, due to space limit we were not able to include this in our tests.
- The 2D Leap Frog Algorithm outputs very good results if equipped with close enough initial guesses. If an initial guess is too far from an ideal solution to the continuous Shape from Shading then the algorithm falls within a wrong potential well of our cost function.
- According to the time criterion the best grid size for this implementation is 6 by 6 pixels snapshot for local area optimization. This grid has best ratio between the amount of grids needed to cover the whole picture with the amount of variables that are needed to be optimized.
- This technique works obviously for an arbitrary number of images. However the more images are taken the smaller Ω becomes. Also, with more images, the computational cost increases. On the other hand more images imposes tighter conditions on u . This should improve the quality of the reconstruction process.
- Possible future work is to implement 2D Leap Frog algorithm that takes into consideration shading on surface pictures (cases when $-1 \leq \cos \Theta \leq 0$ are excluded).

References

1. Atanassov, R., Bose, P., Couture, M.: Algorithms for optimal outlier removal. School of Computer Science, Carleton University, Ottawa
2. Chauvenet, W.: A Manual of Spherical and Practical Astronomy, 5th edn. Lippincott, Philadelphia (1960)
3. Horn, B.: Robot Vision. McGraw-Hill, New York (1986)
4. Horn, B., Brooks, M.: Shape from Shading. The MIT Press (1989)
5. Maruya, M., Nemoto, K., Takashima, Y.: Texture based 3D shape reconstruction from multiple stereo images. In: Proceedings of 11th IAPR International Pattern Recognition Conference A: Computer Vision and Applications, vol. I, pp. 137–140 (1992)
6. Noakes, L., Kozera, R.: A 2D Leap Frog algorithm for optimal surface reconstruction. In: Proc. SPIE 1999, Vision Geometry VII–3811, pp. 352–364 (1999)
7. Noakes, L., Kozera, R.: Denoising Images: Non-linear Leap-Frog for Shape and Light-Source Recovery. In: Asano, T., Klette, R., Ronse, C. (eds.) Geometry, Morphology, and Computational Imaging. LNCS, vol. 2616, pp. 419–436. Springer, Heidelberg (2003)
8. Kozera, R.: Existence and uniqueness in photometric stereo. Applied Mathematics and Computation 44(1), 1–104 (1991)
9. Noakes, L., Kozera, R.: Nonlinearities and noise reduction in 3-Source Photometric Stereo. Journal of Mathematical Imaging and Vision 18(II), 119–127 (2003)
10. Noakes, L., Kozera, R., Klette, R.: The Lawn-Mowing Algorithm for noisy gradient vector. In: Proc. SPIE 1999, Vision Geometry VIII–3811, pp. 305–316 (1999)
11. Zubrzycki, S.: Lectures in Probability Theory and Mathematical Statistics. American Elsevier Pub., New York (1972)

Dynamic Signature Recognition Based on Modified Windows Technique

Rafal Doroz and Krzysztof Wrobel

Institute of Computer Science, University of Silesia, Poland,
ul. Bedzinska 39, 41-200 Sosnowiec
{rafal.doroz,krzysztof.wrobel}@us.edu.pl

Abstract. The paper presents the method of signature recognition, which is a modification of the windows technique. This windows technique allows comparing signatures with the use of any similarity coefficient, without the necessity of using additional algorithms equalizing the lengths of the sequences being compared. The elaborated method introduces a modification regarding the repeatability of individual fragments of person's signature. Thus signature verification is performed only on the basis of signature fragments, characterized by the highest repeatability. Studies using the proposed modification have shown that it has a higher efficiency in comparison to the standard method.

Keywords: Signature recognition, windows technique, similarity measure.

1 Introduction

Biometric techniques are currently among the most dynamically developing areas of science. They prove their usefulness in the era of very high requirements set for security systems. Biometrics can be defined as a method of recognition and personal identification based on physical and behavioural features [1,4,5,12]. Physiological biometrics covers data coming directly from a measurement of a part of human body, e.g. a fingerprint, a shape of face, a retina. Behavioural biometrics analyses data obtained on the basis of an activity performed by a given person, e.g. speech, handwritten signature.

Data collection process within a signature recognition process can be divided into two categories: static and dynamic. The static system collects data using off-line devices [11]. A signature is put on paper, and then is converted into a digital form with the use of a scanner or a digital camera. In this case, the shape of the signature is the only data source, without the possibility of using dynamic data. On the other hand, dynamic systems use on-line devices, which register, apart from the image of the signature, also dynamic data connected with it. The most popular on-line devices are graphics tablets. Thanks to tablets, a signature can be recorded in the form of an n -point set [5,6]. Values of individual features such as: position, inclination, and pressure of a pen are determined in each point. Fig. 1 presents an example of signature S_i .



Fig. 1. Sample of signature S_i and its selected points

When analysing signatures of the same person, it can be noticed that they differ from each other. Certain fragments of signatures are more similar to each other (repeatable) and other ones may differ considerably from one another. Example of such situation is presented in Fig. 2.

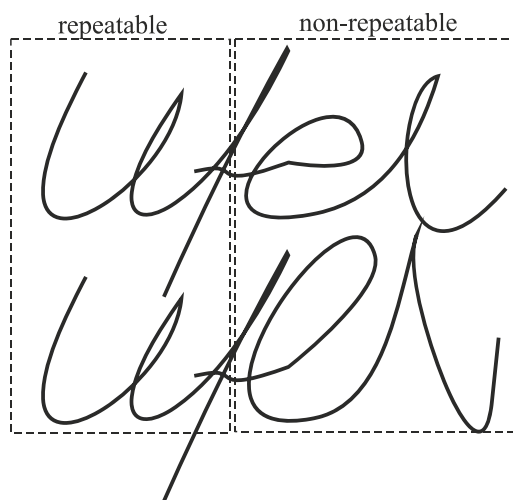


Fig. 2. Two signatures of the same person with a repeatable beginning

Many methods of determining the similarity between signatures require that the signatures being compared have the same length. This requirement is not always fulfilled, since the mentioned earlier differences between the signatures may result in their various lengths. This causes a necessity to use a method of equalizing the length of signatures.

Lengths of sequences can be equalized using many methods, such as DTW [16] or scaling methods [3]. A disadvantage of these methods is the need to interfere with the analysed data, which in turn may lead to distortion of the signatures being compared. The studies [13,14,15] present a method, called *windows technique*, which allows determining the similarity of signatures without the necessity of initial equalization of their lengths. In this way, the signatures being compared are not distorted as it happened in the case of the DTW algorithm or the scaling method.

The main goal on the investigation was to determine windows parameters to signature recognition level. The newest researches point out that new method of selection of some parameters gives better recognition level compare to previously reported work [13,14,15]. Between two compared signatures some differences can be observed, even for signatures of the same individual. It can also be observed that in many signatures some fragments are similar or not. For example one signer put signature almost the same at the beginning, while for other signer his signatures are very similar at the end. The main idea of the investigations is to find similarities and dissimilarities between fragments of the signatures. It will be more precisely explained in the next paragraphs of this paper.

2 Window Technique

In the *windows technique*, the S_i and S_j signatures being compared are divided into equal fragments. Each fragment contains h signature points. Such fragments are called "*windows*" and are designated as "*win*". The k -th window in the S_i signature is designated as $winS_i(k)$, while l -th window in the S_j signature - as $winS_j(l)$. Next windows in the signature can be shifted in relation to each other by a certain number of points designated as jmp . In the S_i signature this parameter was designated as $jmpS_i$, while in the S_j signature - as $jmpS_j$. Appropriate selection of values of the jmp parameter affects the speed and effectiveness of the method.

The values of the h and jmp parameters affect the number of the windows in the analysed signature. The number of all windows in the S_i signature is designated as nwS_i , while in the S_j signature - as nwS_j . The *windows technique* and the influence of parameters on the operation of this method have been discussed in detail in [14,15]. The division of the signature S_i into windows is shown in Fig. 3.



Fig. 3. Division of signature S_i into windows, where $h=5$, $jmpS_i=1$

The process of comparing signatures consists in successive determination of the similarity between each window in the first signature and all windows in the second signature. An example illustrating a comparison of the first window in the S_i signature with windows in the S_j signature is shown in Fig. 4.

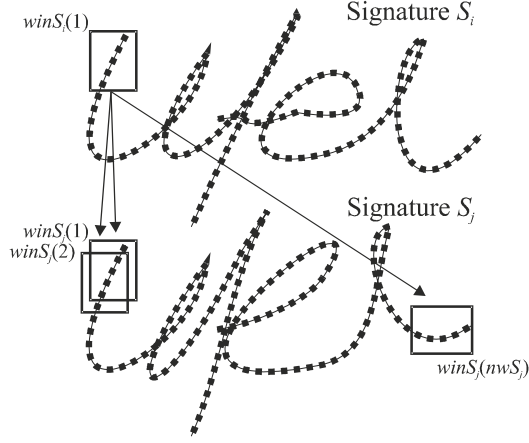


Fig. 4. Comparison of the windows in the two signatures

That same amount of data in the windows being compared allows using any similarity measure M . For each pair of the windows being compared, $winS_i(k)$ in the S_i signature and $winS_j(l)$ in the S_j signature, their similarity is calculated using the following formula:

$$sim_{k,l} = M(winS_i(k), winS_j(l)), \quad (1)$$

where:

- M – similarity measure,
- k – number of the window in signature S_i ,
- l – number of the window in signature S_j .

The result of comparing the k -th window in the S_i signature with the windows determined in the S_j signature is the set of similarity values SIM_k :

$$SIM_k = \{sim_{k,1}, sim_{k,2}, ..., sim_{k,nwS_j}\}, \quad (2)$$

where:

$sim_{k,l}$ – similarity between the compared windows $winS_i(k)$ and $winS_j(l)$, for $l=1, ..., nwS_j$.

The SIM_k set is determined for all windows created in the S_i signature. After the similarity between all windows in the two signatures has been determined, the total similarity between the S_i and S_j signatures can be finally determined:

$$WS(S_i, S_j) = \frac{1}{nwS_i} \sum_{k=1}^{nwS_i} \max(SIM_k), \quad (3)$$

where:

$WS(S_i, S_j)$ – similarity between the S_i and S_j signatures.

The parameter in the *windows technique*, which affects the speed and effectiveness of the method, is *dist*. It narrows down the range of windows in the S_j signature, with which the window analysed in the S_i signature is being compared.

So far the value of the *dist* parameter has been the same for all the signatures of each person. A modification of the *windows technique* has been presented in this study, thanks to which the range of windows being compared is selected individually for signatures of each person. It allowed obtaining better results of the classification. The proposed modification is described in detail in the next section.

3 Modification of the Window Technique

The *Tabdist* arrays constitute a key element in the modification of the *windows technique*. The *PS* set containing the genuine signatures of a given person is required for creating these arrays.

$$PS = \{S_1, S_2, \dots, S_{ns}\}, \quad (4)$$

where:

ns – number of genuine signatures in *PS* set,

S_i – i -th signature of the person, where $i = 1, \dots, ns$.

The *Tabdist* arrays are created separately for each genuine signature $S_i \in PS$. The *PS* set must contain at least three signatures. However the number of elements in the *PS* set should be as large as possible, because it is easier to assess the repeatability of signatures in a larger set.

The algorithm for determining the *Tabdist* array involves comparing the signatures from the *PS* genuine set using the round robin method. The comparison is performed for the parameter values determined in the windows technique ($jmpS_i$). The operation of the algorithm for determining the *Tabdist* array for the $S_i \in PS$ signature can be presented in several steps.

Step 1 – let $k=1$.

Step 2 – determine successively the similarity between the $winS_i(k)$ of the S_i signature in relation to all windows created in the $S_j \in PS \setminus \{S_i\}$ signature, as shown in Fig. 4.

Step 3 – determine the SIM_k similarity set containing the results of comparisons between the k -th window of the S_i signature and the windows created in the S_j signature.

$$SIM_k = \{sim_{k,1}, sim_{k,2}, \dots, sim_{k,nwS_j}\}, \quad (5)$$

where:

$nw S_j$ – number of the window in signature S_j .

Step 4 – determine the maximum value of the similarity from the SIM_k set and remember the number of the ms_k window in the S_j signature, for which this value was determined. This number can be determined from the formula:

$$ms_k = \arg \max_i \{sim_{k,i} \in SIM_k\}, \quad i=1, \dots, nwS_j. \quad (6)$$

Step 5 – determine the number of the S_j signature point, in which the window with the ms_k number begins:

$$pms_k = ((ms_k - 1) \cdot jmpS_j) + 1, \quad (7)$$

where:

pms_k – number of the signature point, in which ms_k window begins.

Step 6 – normalize the pms_k value to the $[0,1]$ interval using the following formula:

$$nmps_k = \frac{pms_k}{m}, \quad (8)$$

where:

m – number of S_j signature points.

Step 7 – write the value of the $nmps_k$ parameter in k -th column of the *Tabdist* array.

A sample $nmps_k$ value, for $k=1$, calculated when comparing the S_i and S_j signatures is presented in Table 1.

Table 1. Table *Tabdist* completed for first window in S_i signature

	$nmps_1$	$nmps_2$	$nmps_3$	\dots	$nmps_{nwS_j}$
$S_i \leftrightarrow S_j$	0.018

Step 8 – repeat steps 2 through 7 successively for $k=2, \dots, nwS_i$.

As a result of carrying out the steps 1 through 8 of the aforementioned algorithm, the first row of the *Tabdist* array is populated with values (Table 2).

Table 2. Table *Tabdist* completed for all windows in S_i signature

	$nmps_1$	$nmps_2$	$nmps_3$	\dots	$nmps_{nwS_i}$
$S_i \leftrightarrow S_j$	0.018	0.027	0.036	...	0.943

Step 9 – repeat the steps 1 through 8, comparing each time the S_i signature with the next genuine signature from the *PS* set. As a result of comparing each pair of the signatures the next row of the *Tabdist* array is obtained.

Sample *Tabdist* array for the S_1 signature from the $PS=\{S_1, S_2, S_3, S_4\}$ set is presented in Table 3.

Table 3. Fragment of the table *Tabdist* completed for S_1 signature

	$nmps_1$	$nmps_2$	$nmps_3$	\dots	$nmps_{nwS_1}$
$S_1 \leftrightarrow S_2$	0.018	0.027	0.036	...	0.943
$S_1 \leftrightarrow S_3$	0.026	0.040	0.053	...	0.917
$S_1 \leftrightarrow S_4$	0.013	0.026	0.039	...	0.918

Step 10 – after the *Tabdist* array for the S_i signature has been created, mean values $nmps$ and standard deviation values σ for individual columns of the *Tabdist* array are determined (table 4).

Table 4. Fragment of the table *Tabdist* completed for S_1 signature

	$nmps_1$	$nmps_2$	$nmps_3$	\dots	$nmps_{nwS_1}$
$S_1 \leftrightarrow S_2$	0.018	0.027	0.036	...	0.943
$S_1 \leftrightarrow S_3$	0.026	0.040	0.053	...	0.917
$S_1 \leftrightarrow S_4$	0.013	0.026	0.039	...	0.918
\overline{nmps}	0.019	0.031	0.043	...	0.926
σ	0.006	0.008	0.009	...	0.015

Step 11 – remove from the *Tabdist* array the columns, in which the standard deviation value is greater than a certain threshold value $\zeta \in [0,1]$. Removing these columns from the *Tabdist* array causes that the non-repeatable signature fragments are not compared with each other.

Thanks to the *Tabdist* array the k -th window in the S_i genuine signature is compared with a sequence of windows in the S_j signature. The number of the first and the last window in the sequence is determined using the following formulas:

$$\begin{aligned}
 pwin(k) &= \text{round}(\overline{npms}(k) \cdot m), \text{ for } k = 1, \dots, nwS_i, \\
 pwin_{\min}(k) &= \text{round}(pwin(k) - \sigma(k) \cdot m), \text{ for } k = 1, \dots, nwS_i, \\
 pwin_{\max}(k) &= \text{round}(pwin(k) + \sigma(k) \cdot m), \text{ for } k = 1, \dots, nwS_i.
 \end{aligned} \tag{9}$$

where:

- $pwin(k)$ – the middle window in the sequence,
- $pwin_{\min}(k)$ – the first window in the sequence,
- $pwin_{\max}(k)$ – the last window in the sequence,
- $\overline{npms}(k)$ – the mean value read from the k -th column of the *Tabdist* array for k -th window in the S_i signature,
- $\sigma(k)$ – the standard deviation read from the k -th column of the *Tabdist* array for k -th window in the S_i signature,
- m – number of the S_j signature point,
- nwS_i – number of the window in signature S_i .

Figure 5 shows the manner of comparing the windows, taking into account the *Tabdist* array.

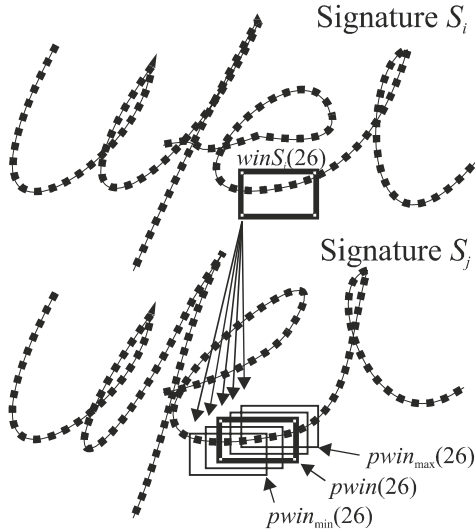


Fig. 5. The example of comparing the windows, taking into account the *Tabdist*

Step 12 – end the algorithm, if the S_i signature has been compared with all the signatures from the *PS* set.

The result of the operation of the modified *windows technique* is the *Tabdist* array generated for each genuine signature. This array is then used in the signature comparison process.

4 Research Results

The purpose of the studies was to determine the effectiveness of the proposed modification and its impact on the signature verification results. The standard windows method was compared with the modified method in the course of the studies. The studies were conducted with the use of signatures from the MCYT database [20]. The database used in the studies contained 1000 signatures of 100 persons. A recognized signature belonging to a given person was compared with 5 genuine signatures of the same person. The set of genuine signatures did not contain any recognizable signatures. The recognizable signatures included 3 other original signatures of a given person and 2 forged signatures of this person.

The studies were conducted for the following ranges of parameter values:

- $dist=[0.1, \dots, 0.5]$ with a step of 0.1,
- $h=[10, \dots, 50]$ with a step of 10,
- the standard deviation $\sigma=[0.05, \dots, 0.2]$ with a step of 0.05.

All combinations of the above parameters were examined. During the research, the several similarity coefficients have been used [14, 15]. The best results were obtained with the use of R^2 ratio [1]. During the studies, the following signature features were examined: coordinates (x,y) of signature points, pen pressure p at the point (x,y) . EER was calculated for each measurement. The lower the value of EER, the lower error of a given measurement is. Table 5 shows the results of the studies obtained for the standard and modified windows method.

Table 5. The best measurements results for standard and modified window technique

	Number of points in window h	Standard deviation values σ	EER [%]
Standard window technique	40	0.3	6.59
Modified window technique	40	0.2	3.20

5 Conclusions

The use of the modified windows technique allows to reduce verification error rate in comparison to standard windows technique. The obtained result $EER = 3.20\%$ is also competitive in comparison to other methods, known from the literature. Table 6 summarizes the well-known methods of signature recognition published in recent years. The methods shown in Table 6 were tested by their authors, as in the present study, using the MCYT signature database.

Table 6. Different online signature verification methods

Authors	Results (EER [%])
Presented method	3.20
Fierrez J., Ortega - Garcia J., Ramos D., Gonzalez - Rodriguez J. [2]	0.74
Lumini A., Nanni L. [7]	4.50
Maiorana E. [8]	8.33
Nanni L., Lumini A. [9]	21.00
Nanni L., Maiorana E., Lumini A., Campisi P. [10]	3.00
Vargas J. F., Ferrer M. A., Travieso C. M., Alonso J. B. [17]	12.82
Vivaracho - Pascual C., Faundez - Zanuy M., Pascual J. M. [18]	1.80
Wen J., Fang B., Tang Y. Y., Zhang T. [19]	15.30

The important advantages the proposed method is to determine of the parameter *dist*. It should be noted that this parameter is automatically selected. This selection is possible on the basis of analyzing of the *Tabdist* array. The obtained classification results encourage the further modification of the presented technique. In the next investigations stages using more complex methods of data analysis are planned. Additionally, time and memory complexity will be also estimated.

References

1. Doroz, R., Porwik, P., Para, T., Wróbel, K.: Dynamic Signature Recognition Based on Velocity Changes of Some Features. *International Journal of Biometrics* 1(1), 47–62 (2008)
2. Fierrez, J., Ortega-Garcia, J., Ramos, D., Gonzalez-Rodriguez, J.: HMM-based On-line Signature Verification: Feature Extraction and Signature Modeling. *Pattern Recognition Letters* 28(16), 2325–2334 (2007)
3. Foley, J.D.: *Introduction to Computer Graphics*. Addison-Wesley (1993)
4. Impedovo, S., Pirlo, G.: Verification of Handwritten Signatures: an Overview. In: 14th International Conference on Image Analysis and Processing, pp. 191–196 (2007)
5. Kamel, M.S., Ellis, G.A., Sayeed, S.: Glove-based Approach to Online Signature Verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1109–1113 (2008)
6. Khan, M.K., Khan, M.A., Khan, M.A.U., Ahmad, I.: On-Line Signature Verification by Exploiting Inter-Feature Dependencies. In: 18th International Conference on Pattern Recognition, pp. 796–799 (2006)
7. Lumini, A., Nanni, L.: Ensemble of On-line Signature Matchers Based on Over Complete Feature Generation. *Expert Systems with Applications* 36(3), 5291–5296 (2009)
8. Maiorana, E.: Biometric Cryptosystem Using Function Based On-line Signature Recognition. *Expert Systems with Applications* 37(4), 3454–3461 (2010)
9. Nanni, L., Lumini, A.: Ensemble of Parzen Window Classifiers for On-line Signature verification. *Neurocomputing* 68, 217–224 (2005)
10. Nanni, L., Maiorana, E., Lumini, A., Campisi, P.: Combining local, regional and global matchers for a template protected on-line signature verification system. *Expert Systems with Applications* 37(5), 3676–3684 (2010)
11. Porwik, P.: The Compact Three Stages Method of the Signature Recognition. In: 6th IEEE International Conference on Computer Information Systems and Industrial Management Applications, pp. 282–287 (2007)
12. Porwik, P., Para, T.: Some Handwritten Signature Parameters in Biometric Recognition Process. In: 29th IEEE International Conference on Information Technology Interfaces, pp. 185–190 (2007)
13. Porwik, P., Doroz, R., Wrobel, K.: A New Signature Similarity Measure. In: 8th IEEE Int. Conf. on Computer Information Systems and Industrial Management Applications, Coimbatore, India, pp. 1022–1027 (2009)
14. Porwik, P., Doroz, R., Wróbel, K.: A New Signature Similarity Measure Based on Windows Allocation Technique. *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)* 2, 297–305 (2010)
15. Porwik, P., Wrobel, K., Doroz, R.: Signature Recognition Method by Means of the Windows Technique. *An International Journal of Image Processing & Communications* 14(2-3), 43–50 (2009)
16. Shanker, A.P., Rajagopalan, A.N.: Off-line signature verification using DTW. *Pattern Recognition Letters* 28, 1407–1414 (2007)
17. Vargas, J.F., Ferrer, M.A., Travieso, C.M., Alonso, J.B.: Off-line Signature Verification Based on Grey Level Information Using Texture Features. *Pattern Recognition* 44(2), 375–385 (2011)
18. Vivaracho-Pascual, C., Faundez-Zanuy, M., Pascual, J.M.: An Efficient Low Cost Approach for On-line Signature Recognition Based on Length Normalization and Fractional Distances. *Pattern Recognition* 42(1), 183–193 (2009)
19. Wen, J., Fang, B., Tang, Y.Y., Zhang, T.: Model-based Signature Verification with Rotation Invariant Features. *Pattern Recognition* 42(7), 1458–1466 (2009)
20. <http://atvs.ii.uam.es/databases.jsp>

Rigid and Non-rigid Shape Matching for Mechanical Components Retrieval

Andrea Albarelli, Filippo Bergamasco, and Andrea Torsello

Dip. di Scienze Ambientali, Informatica e Statistica, Università Ca' Foscari Venezia

Abstract. Reducing the setup time for a new production line is critical to the success of a manufacturer within the current competitive and cost-conscious market. To this end, being able to reuse already available machines, toolings and parts is paramount. However, matching a large warehouse of previously engineered parts to a new component to produce, is often more a matter of art and personal expertise rather than predictable science. In order to ease this process we developed a database retrieval approach for mechanical components that is able to deal with both rigid matching and deformable shapes. The intended use for the system is to match parts acquired with a 3D scanning system to a large database of components and to supply a list of results sorted according with a metric that expresses a structural distance.

1 Introduction

While the exact pipeline for bringing new goods into production varies significantly between markets, some common fundamental steps can be identified. Specifically, the most important milestones are the concept formulation, the design phase, the creation of a final prototype and the engineering of the production process [1]. Given the increasing competitiveness in the field of production, both in terms of time and cost, it is not surprising that a great effort is made to develop approaches that allow for a more efficient process for each of these four distinct phases.

The concept formulation alone could consume up to 50% of the time required for the whole cycle, thus a wide range of time saving approaches have been proposed in literature to reduce its impact [2, 3]. However, in this paper, we concentrate on the remaining three phases, since they are all dependant on technical judgements that can be eased by exploiting an automated tool.

The design phase is about the translation of the requirements emerging from the concept in a full product definition that can be implemented in a prototype. Depending on the market, the focus of the design step can be on the functionality of the product or on aspects related to fashion. Either way, the choices made at this stage can not be decoupled from the following phases. In Fact, the early verification of the relations between the design, the associated technical challenges and the actual capabilities of the available production system, could lead to significant reductions in the overall production cost and time to market [4, 5].

The prototyping is the bridge between the design and the assembly of the actual production process. At this stage, the requirements expressed during the design phase are casted into a physical object (usually handcrafted) and the limitation in the feasibility of some of the planned features may appear. At the same time, the requirements for the tooling (i.e. the set of machine configurations and custom tools needed for actual production) start to become apparent. Of course the quandary between changes in the design made with the goal of production simplification and implementation of new toolings must be managed. To this end, a partial overlapping and a regulated communication between phases has consistently been shown to be both useful and necessary [6-8].

The automatic tools available to help with the described process are many and diverse. In this paper we are focusing on a specific but very critical topic: the reuse of toolings.

Specifically, the most convenient way of reusing toolings is to adapt those made for the production of a component that is very similar (in shape and materials) to one already engineered in the past history of the factory. While this could seem a straightforward task, it must be taken in account that within, a medium to large factory, thousands of different new components can easily be introduced into the production each year. Currently, the most widely adopted method to solve this problem (at least in the factories we surveyed) is to resort to experts that have been working in the production department for a long time. Such experts are able to recall (by memory or by consulting a large archive of drawings) if a component similar to the one at issue have already been produced. Needless to say, this kind of approach can not be deemed as reliable or dependable for several reasons. To begin with, there is no guarantee that the experts are able to exhibit a good enough recall rate, moreover as the knowledge is not an asset of the company, but rather of individuals, the transfer of such assets as personnel turnover happens is difficult and very prone to errors.

The solution we are introducing automates this selection in a semi-supervised manner, making available to experts a structured tool to guide the crawling through a large product database. The proposed system (which will be described in depth in Sec. 2) is mainly based on shape-based recognition. Roughly speaking, shape matching is a technique widely developed in the field of Computer Vision and Pattern Recognition whose goal is to find the alignment or deformation transformation that relates a *model* object with one or more *data* scenes. There exist many different classifications of matching techniques, however, for our purposes, we break down them in two application domains and two transformation models. The application domains are respectively images (2D data) and surfaces (3D data). The transformation models which we are interested in are rigid, where model and data must align exactly and non-rigid, where a certain degree of elastic deformation is permitted.

Image-based shape matching is primarily performed by finding correspondences between point patterns which can be extracted from images using detectors [9-11] and descriptors [12, 13] that are locally invariant to illumination, scale and rotation. The matching itself happens through a number of different

techniques that ranges from Procrustes Alignment [14] to Graphical Models [15] for the rigid scenario, and from Relaxation Labeling [16] to Gaussian Mixture Models [17] for non-rigid matching.

In the first phase of our investigation we evaluated the adoption of image-based methods applied over shots of the components under a set of different angles. Unfortunately this approach was not robust enough to grant a reasonable performance, additionally, the building of the database through image capturing was very time consuming and error prone. For this reason we resorted to the use of 3D matching techniques. Surface matching, albeit being addressed by literature for a long time, is recently boldly emerging due to the availability of cheaper and more accurate 3D digitizing hardware and to the increasing processing power of computer systems that allows for a feasible handling of the more complex volumetric information. In the last decade, more and more problems traditionally tackled with different techniques have been proven to be addressable by exploiting surface matching. For instance, in the field of biometric, the use of 3D surfaces as a substitute for images has shown to attain a far superior performance [18]. Exceptionally good results have been obtained in particular in the recognition of strongly characterizing traits such as ears [19] or fingers [20]. In classical biometric challenges, such as expression-independent face recognition, methods that are able to tell the difference even between the faces of two twins have been demonstrated [21]. In the industry, surface matching has been used extensively for defect analysis [22], reverse engineering [23] and motion capture [24].

Most 3D matching approaches work best when the transformation between model and data is rigid, as when a deformation is applied most Euclidean metrics can not be exploited. Recently, some effective non-rigid registration techniques that can be applied to 3D surfaces have begun to appear. Some of them are based on Graph Matching [25] or Game Theory [26]. Others perform the needed deformation by optimizing the parameters of Thin Plate Splines [27] or of a flexible surface regulated by a set of quadratic equations [28].

With respect to the pipeline described in the following sections, we resorted to the use of the very standard ICP rigid registration technique [29] since it is a fast algorithm that works well over the almost noiseless data that we are acquiring with a structured light scanner. By contrast, we designed a specially crafted method to tackle the non-rigid search. In fact, the knowledge of the restricted problem dominion, allows for an ad-hoc solution that specifically addresses the constraints we are dealing with.

2 A Shape-Based Pipeline to Maximize Tooling Reuse

The approach presented in this paper is the result of a study commissioned by Luxottica, a world leading designer and producer of eyewear. The goal of the study was the design and the implementation of a full system able to search for components similar to a given part into a database of nearly 150.000 components extracted from the about 30.000 eyeglasses model produced during the half a

century of company history. The main challenges were the sheer number of objects to scan, which would imply several years of work with manual 3D scanners, and the need for a fast and accurate search system able to deal with rigid and non-rigid matching. In fact, eyeglasses frames are made up of both unarticulated parts, such as the front bar or the rims of the lenses, and of bending components such as the earpiece or the nose pads. In Fig. 1 an overall view of the proposed system is shown. As a preliminary step, the whole warehouse of available previously produced models is scanned using a structured light scanner customized to perform a fast and unattended digitizing of an eyeglass frame. The acquired models must be splitted into their basic components to be useful for the part search engine. Unfortunately, this kind of segmentation can not be done automatically, since the heterogeneity in component shapes and positions over the model is too high to grant an algorithm the ability to tell where actually a rim ends or a nosepad starts. To solve this problem in the most efficient manner we designed a semi-supervised segmentation tool that works by operating a greedy region growing regulated by the first derivative of the surface curvature. Using this tool a human operator is able to perform a complete model segmentation in a couple of minutes. Each part is subsequently labelled with its type. The component to be used as a query object is created in a similar way, by scanning the prototype frame instead of an archived eyepiece and by segmenting the part of choice. The matching is performed by first choosing the type of search (rigid or non-rigid) and by specifying the metric to be used (in the case of non-rigid matching). The query component is then compared with a suitable algorithm against all the item in the database. While this could seem to be a gargantuan task, both in terms of computing time and memory usage, it is indeed feasible without the need of a multi-level index structure. As a matter of fact, the size of a model component is within a few kilobytes and the typical amount of memory available on a modern server can easily handle hundred of thousands of components. Moreover, all the matching algorithms used can be executed in a few milliseconds, allowing for a full database scan less than a minute (as the

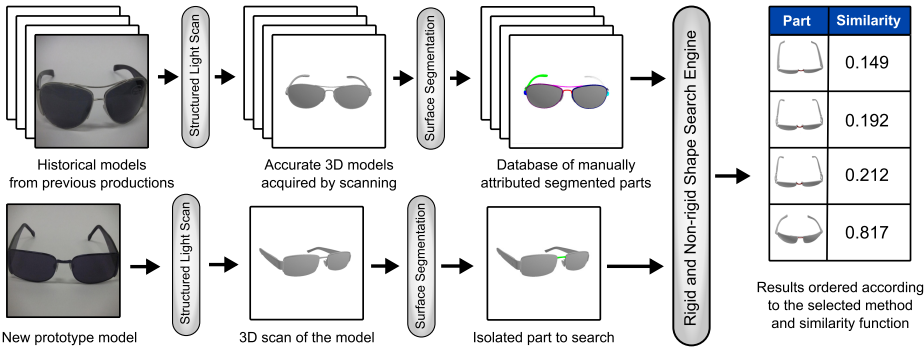


Fig. 1. Overview of the pipeline described in this paper for shape-based rigid and non-rigid component retrieval

search is narrowed for type of component). In practice, however, we narrowed even further the search by filtering out components whose total surface area deviates more than 30% with respect to the surface of the pursued part. This latter optimization provided, on a standard Intel based server, an average query time of about 10 seconds.

2.1 Capturing the 3D Shape of the Components

The building of the components database has been performed by means of a dedicated 3D scanner made up of a specialized support, an automated turntable, and a structured light scanning head (see Fig. 2). The support has been designed to be able to hold an eyewear frame with minimal occlusion and to present it to the scanning head with an angle that allows for a complete and watertight acquisition while turning on a single axis. The scanning head comprises two cameras and a DLP projector used to create light fringes according to the Phase Shift coding variant presented in [30]. A single range image requires about 3 seconds to be captured, and a total of 24 ranges are needed for the complete surface reconstruction. If the time required for the turntable rotation is also accounted, the whole process can be carried on in about two minutes.

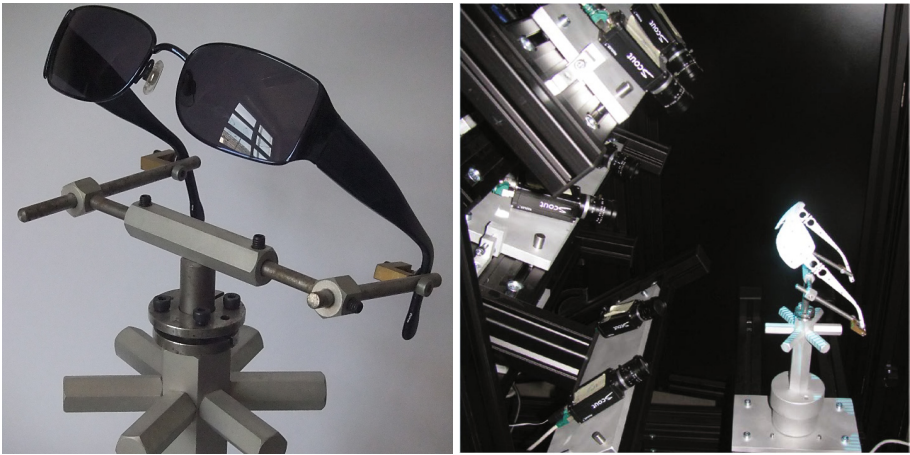


Fig. 2. The custom eyeglasses frame holder and the structured light scanner

Once the scans are completed, the resulting range images are coarsely placed in a common frame by exploiting the knowledge of the step angle of the electric motor that drives the turntable. Subsequently they are pairwise aligned with IPC [29] and globally aligned using Dual Quaternion Diffusion [31]. Finally, a standard reference frame must be imposed to the newly acquired object. This is necessary for obtaining a common orientation of each of the components that will be produced by the following segmentation step, which in turn is needed as an initialization for both rigid and non-rigid search. To this end, a simple Principal Components Analysis (PCA) is performed on the 3D data.

2.2 Semi-supervised Object Segmentation

To help the user in the component segmentation, we designed a semi-supervised tool that is able to separate the single components from a scanned model starting from a limited number of initial seeds supplied by a human operator. Those seeds are used as a hint that indicates areas that belong for certain to different parts. Each area is then grown in a greedy manner until it hits another area. The main idea is that the growing becomes slower when abrupt changes in surface normals are encountered and thus notches on the surface (that are typically associated to small gaps between parts) act as a containment border.

This tool does not work directly on the surface of the object, but rather on an apt dual graph representation [32]. As shown in Fig. 3 each node of this graph corresponds to the a triangle of the mesh. There are no geometrical relations between these nodes and the absolute position of the triangles in space. For this reason we do not need any attribute on the graph nodes. By converse, we are interested in the relations between adjacent faces, thus we are going to define a scalar attribute for the graph edges. Specifically, we want to assign to each edge a weight that is monotonical with the “effort” required to move between the two barycenters of the faces. This effort should be higher if the triangles exhibit a strong curvature with a short distance between their centers and it should be low if the opposite happens. To this extent, given two nodes of the graph associated to faces i and j , we define the weight between them as:

$$\omega(i, j) = \frac{1 - \langle n_i, n_j \rangle}{|p_i - p_j|} \quad (1)$$

where $\bar{p} = (p_1, p_2 \dots p_k)$ is the vector of the barycenters of the faces and $\bar{n} = (n_1, n_2 \dots n_k)$ are the respective normals. $\langle \cdot, \cdot \rangle$ denotes the scalar product and $|\cdot|$ the Euclidean norm.

In Fig. 3 (c) edge weight is represented by using a proportional width in the drawing of the line between two nodes. It can be seen how edges that connect faces with stronger curvatures exhibit larger weight.

Once the weighted graph has been created, the segmentation can happen. In our framework the surface is segmented starting from one or more hints

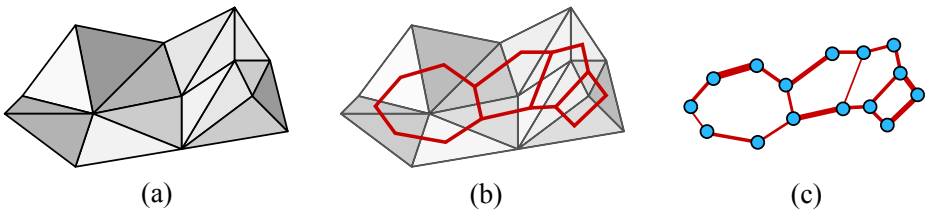


Fig. 3. Steps of the graph creation process. From the initial mesh (a) the dual graph is built creating a vertex for each face and connecting each pair of adjacent faces (b). Finally, each edge of this graph is then weighted according to the dot product between the normals of the connected faces.

provided by the user. This human hint expresses a binary condition on the mesh by assigning a small fraction of all the nodes to a set called *user selected green nodes* and another small portion to a set called *user selected red nodes*. We call *green nodes* the faces (nodes) belonging to the segment of interest and *red nodes* the ones that are not belonging to it, regardless of the fact that those nodes have been manually or automatically labeled. The proposed algorithm distributes all graph nodes in the *green nodes* and *red nodes* sets in a greedy way.

We define a seed as triple $\langle n, t, w \rangle$ where n is the graph node referred by this seed, t is a boolean flag that indicates if n has to be added to green or red nodes, w is a positive value in \mathbb{R}^+ . At the initialization step, for each initial green and red node selected by the user, a seed is created and inserted into a priority queue with an initial weight value $w = 0$. All nodes are also added to the *unassigned nodes* set. At each step, the seed $\langle n, t, w \rangle$ with lowest value of w is extracted from the priority queue and its referred node n is added to *green nodes* or *red nodes* according to the seed's t flag. The node is also removed from *unassigned nodes* to ensure that each node is evaluated exactly once during the execution of the algorithm. For each node $n' \in \text{unassigned nodes}$ connected to n in the graph, a new seed $\langle n', t' = t, w' = \omega(n, n') \rangle$ is created and added into the queue. It has to be noted that it is not a direct consequence of such insertion that the final type of n' (either green or red) is determined by the type t' of this seed. At any time multiple seeds referring the same node can exist in the queue, with the only condition that a node type can be set only once. During the execution of algorithm either the region of green nodes and the region of red ones expands towards the nodes that would require less weight to be reached. Once all nodes in the same connected component are visited, the result of this assignment is shown to the user who can either refine his initial hint or accept the proposed segmentation. Of course the procedure can be iterated to obtain a hierarchical segmentation. In any condition, the algorithm will run in $O(N)$ time since, with the described greedy approach, each node is visited once.

In Fig. 4 and example of the segmentation produced by the tool is shown.

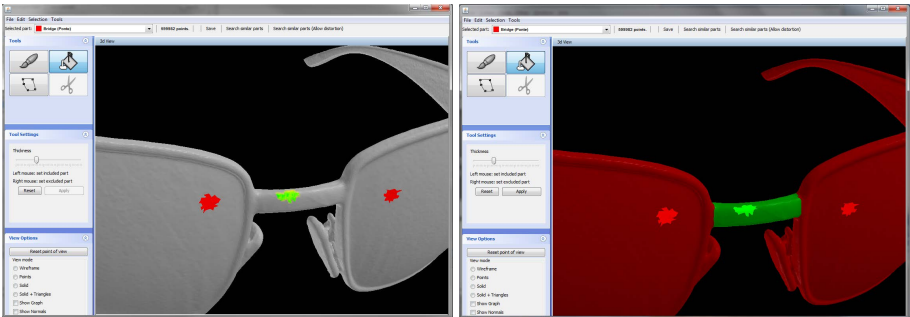


Fig. 4. An example showing an initial manual seeding and the associated semi-supervised segmentation

2.3 Rigid Components Matching

The rigid matching consist in an alignment of each part in the database with the query part using an efficient variant [33] of the ICP (Iterative Closest Point) algorithm [29]. In detail, each component in the database is sampled and exactly 1000 3D points are extracted. The surface of the query object is coarsely aligned with those samples through the PCA previously computed over the whole model. Thus, for each sample point, the intersection between the vertex normal and the surface of the query object is found (if it exists). This intersection generates a new point, that is associated to the original vertex as the ideal mate on the query surface. After all the intersections have been computed a closed-form optimization is used to align mating points to model vertices [34]. The process is iterated many times, until the relative motion of the query with respect to the model becomes negligible. The idea of this algorithm is that the positional error committed when adopting the points generated by normal shooting as true correspondences becomes smaller at each iteration. In practice, the ability of ICP to converge to a correct global minimum strongly depends on the initial coarse alignment. In fact, a less than good initial estimation can easily lead to completely wrong final alignment, even with perfectly correspondent 3D objects. In our validation, however, the coarse registration supplied by the global PCA consistently allowed to obtain a local minimum when the object to be compared was the same or a component similar enough to be effectively used as a tooling source. After an optimal aligned is obtained, a last normal shooting is performed and the RMS of the distance between each vertex and its virtual mate is computed. This RMS is used as the metric distance for the ordering of the results. In Fig. 5 we show some results obtained by searching respectively for a bridge and a front hinge (highlighted in red in the figure). On the side of each result the corresponding RMS associated to the best alignment is reported.









Query	Result 1	Score	Result 5	Score	Result 10	Score
		0.155		0.209		0.266
		0.508		0.715		1.0

Fig. 5. Results obtained by the rigid search engine for some query objects (best viewed in colors)

2.4 Non-rigid Components Matching

We decided to address the problem of non-rigid components matching as the process of finding the best global or local alignment between two linear sequences. Every component is first sliced into a vector of equally-spaced slices along its median axis.

The slicing procedure starts by roughly aligning the first two principal vectors of the component along x and z axis using the PCA. After that, starting from the farthest vertex along the negative side of x axis, a set of n equally-spaced planes parallel to yz plane are used to intersect the component defining n different closed planar contours. For each of those contours, the size w_i and h_i respectively along z and y axes are used to characterize each slice with the quadruple $s_i = \langle w_{i-1}^s, h_{i-1}^s, w_i^s, h_i^s \rangle$. For each component, the vector containing all slices defined above is stored into the database, together with the component id and its size. After that, we defined the similarity between two slices s_i and t_i as

$$s(s_i, t_i) = \begin{cases} m(s_i) - m(t_i) \Leftrightarrow C_i^{s,t} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $C_i^{s,t} = (w_{i-1}^s \leq w_{i-1}^t \leq w_i^s) \wedge (h_{i-1}^s \leq h_{i-1}^t \leq h_i^s) \vee (w_{i-1}^t \leq w_{i-1}^s \leq w_i^t) \wedge (h_{i-1}^t \leq h_{i-1}^s \leq h_i^t)$ and $m(s_i) = \frac{w_{i-1}^s + w_i^s}{2} + \frac{h_{i-1}^s + h_i^s}{2}$.

Once the similarity measure is defined, we search for a best global alignment by exploiting the well known Needleman-Wunsch algorithm first presented in [35] and the local alignment using Smith-Waterman [36]. Each alignment is performed iteratively between all stored components and the results are sorted by similarity. Depending of specific application requirements, local alignment may offer better results than the global approach, vice-versa. In this extent, we decided to show both ordering to the user, along with the similarity measure. In Fig. 6 an example of the ordering obtained respectively with Needleman-Wunsch and Smith-Waterman algorithm is shown.











Query	Result 1	Score	Result 5	Score	Result 10	Score	Result 20	Score
		6.781		7.499		8.380		8.540
		0.182		0.254		2.737		5.799

Fig. 6. Results obtained by the non-rigid search engine using Needleman-Wunsch (top) and Smith-Waterman (bottom) algorithms (best viewed in colors)

3 Evaluation of the Implemented System

An initial evaluation of the system was performed by acquiring different instances of the same objects and, after manual segmentation, by automatically searching for each single part, which was replicated several times in the test database. To test the robustness of the matching process, the components were modified by adding random Gaussian noise to their vertices. The goal of this validation step was simply to assess the ability of the system to produce an ordering were consistent clusters of components appear before the remaining results. While this test was successful, obtaining a 100% recognition rate, it is only meaningful to check that the search algorithm is correctly working as a pure object retrieval tool, but it does not really indicates if similar, but not identical, parts can be retrieved with a reasonable ordering. In order to obtain a measure of how well the system works for the intended purpose, we let the experts use it for some months. During this trial, the experts have been asked to annotate, after each component search, the position in the result list of the first occurrence that they could consider to be similar enough for tooling reuse purposes. If a similar enough component is not found at all (or the expert knows that it does not exists in the database) the annotation does not happen and the event is deemed to be a true negative. The results of this test are given in Fig. 7 as the percentage of cases where the sought component is found among the first N answers supplied by the engine. The first graph shows the performance of the rigid matcher. As expected, in most cases the correct component can be found as the first match and in no cases more than 50 results must be scanned by the user. The capability of the non-rigid matcher, shown in the second graph of Fig. 7, is a bit lower. In detail, the local sequence alignment performed by Smith-Waterman algorithm seems to behave a little better than the global alignment obtained by Needleman-Wunsch. However, for both algorithms, the correct result can be found most of the times immediately and within the first 50 extracted components in more than 90% of the cases. This level of performance is clearly good enough for an effective use in production environment and arguably better than what could be attained without the assistance of the system.

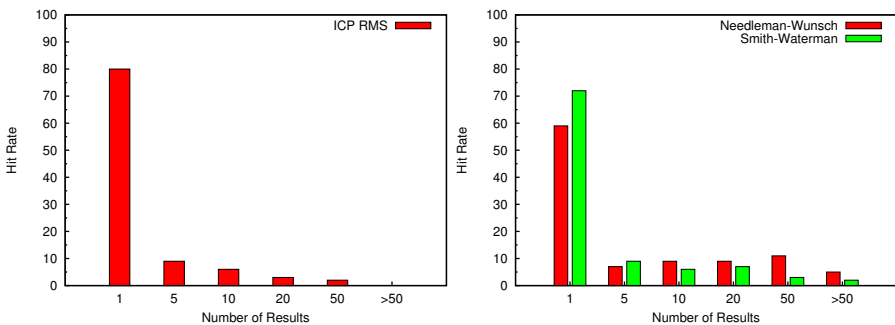


Fig. 7. Evaluation of the hit rate of the search system with respect to the number of results inspected by a human expert for validating purposes

4 Conclusions

In this paper we described a complete pipeline to help field experts during the design process for new production lines. The presented approach is based on both well-known methods and novel domain-specific techniques. In details, the approach allows to easily find toolings and process definition to be reused in new productions by comparing the shape of the newly designed component with a large database of previously engineered parts. This comparison needs to be performed both in a rigid fashion and allowing for some deformation, depending on the type of part to be searched and on the good sense of the expert that is using the tool. Each search produces an ordering of the parts in the database, which can be computed with respect to a set of different metrics. Overall, a sizeable set of trials performed with a database of sunglasses components have been deemed successful by the users, as the most relevant parts were consistently found in the first few results produced by the search engine.

References

1. Ulrich, K.T., Eppinger, S.D.: *Product Design and Development*. McGraw-Hill (1995)
2. Kim, J., Wilemon, D.: Focusing the fuzzy front-end in new product development. *R&D Management* 32, 269–279 (2002)
3. Millson, M.R., Raj, S., Wilemon, D.: A survey of major approaches for accelerating new product development. *Journal of Prod. Innov. Manag.* 9, 53–69 (1992)
4. Dieter, G.E., Schmidt, L.C., Azarm, S.: *Engineering design*, 4th edn. *Journal of Mechanical Design* 131, 056501 (2009)
5. Nihtila, J.: R and D production integration in the early phases of new product development projects. *Journal of Eng. and Tech. Manag.* 16, 55–81 (1999)
6. Takeuchi, H., Nonaka, I.: The new new product development game. *Harvard Business Review* 64, 137–146 (1986)
7. Mabert, V.A., Muth, J.F., Schmenner, R.W.: Collapsing new product development times: Six case studies. *Journal of Prod. Innov. Manag.* 9, 200–212 (1992)
8. Zha, X.F., Sriram, R.D.: Platform-based product design and development: A knowledge-intensive support approach. *Know-Based Syst.* 19, 524–543 (2006)
9. Shi, J., Tomasi, C.: Good features to track. In: 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1994), pp. 593–600 (1994)
10. Smith, S.M., Brady, J.M.: Susan—a new approach to low level image processing. *Int. J. Comput. Vision* 23, 45–78 (1997)
11. Rosten, E., Porter, R., Drummond, T.: Faster and better: a machine learning approach to corner detection. *CoRR* abs/0810.2434 (2008)
12. Lowe, D.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 20, 91–110 (2003)
13. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.J.: Speeded-up robust features (surf). *Computer Vision and Image Understanding* 110, 346–359 (2008)
14. Luo, B., Hancock, E.R.: Matching point-sets using procrustes alignment and the EM algorithm. *Computer*, 43–52 (1999)
15. Caetano, T.S., Caelli, T., Schuurmans, D., Barone, D.A.C.: Graphical models and point pattern matching. *IEEE Trans. PAMI* 28, 2006 (2006)

16. Lee, J.H., Won, C.H.: Topology preserving relaxation labeling for nonrigid point matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 427–432 (2011)
17. Jian, B., Vemuri, B.C.: Robust point set registration using gaussian mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 1633–1645 (2011)
18. Woodard, D.L., Faltemier, T.C., Yan, P., Flynn, P.J., Bowyer, K.W.: A comparison of 3D biometric modalities. In: *Proceedings of CVPR 2006*, pp. 57–64. IEEE Computer Society, Washington, DC (2006)
19. Chen, H., Bhanu, B.: Contour matching for 3D ear recognition. In: *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION 2005)*, vol. 1, pp. 123–128. IEEE Computer Society, Washington, DC (2005)
20. Woodard, D.L., Flynn, P.J.: Finger surface as a biometric identifier. *Comput. Vis. Image Underst.* 100, 357–384 (2005)
21. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Expression-invariant representations of faces. *IEEE Trans. PAMI*, 1042–1053 (2007)
22. Li, Q., Wang, M., Gu, W.: Computer vision based system for apple surface defect detection. *Computers and Electronics in Agriculture* 36, 215–223 (2002)
23. Park, S.C., Chang, M.: Reverse engineering with a structured light system. *Computers and Industrial Engineering* 57, 1377–1384 (2009)
24. Plänkner, R., Fua, P.: Articulated soft objects for multiview shape and motion capture. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 1182–1187 (2003)
25. Zeng, Y., Wang, C., Wang, Y., Gu, X., Samaras, D., Paragios, N.: Dense non-rigid surface registration using high-order graph matching. In: *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2010*, pp. 382–389. IEEE (2010)
26. Emanuele, R., Alex, B., Andrea, A., Filippo, B., Andrea, T.: A game-theoretic approach to deformable shape matching. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*. IEEE (2012)
27. Zou, G., Hua, J., Muzik, O.: Non-rigid Surface Registration Using Spherical Thin-Plate Splines. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) *MICCAI 2007, Part I*. LNCS, vol. 4791, pp. 367–374. Springer, Heidelberg (2007)
28. Salzmann, M., Moreno-Noguer, F., Lepetit, V., Fua, P.: Closed-Form Solution to Non-rigid 3D Surface Registration. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part IV*. LNCS, vol. 5305, pp. 581–594. Springer, Heidelberg (2008)
29. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 239–256 (1992)
30. Lilienblum, E., Michaelis, B.: Optical 3D surface reconstruction by a multi-period phase shift method. *JCP* 2, 73–83 (2007)
31. Torsello, A., Rodolá, E., Albarelli, A.: Multiview registration via graph diffusion of dual quaternions. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011*, pp. 2441–2448. IEEE (2011)
32. Filippo, B., Albarelli Andrea, T.A.: A graph-based technique for semi-supervised segmentation of 3D surfaces. *Pattern Recognition Letters* (2012)
33. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: *Proc. of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pp. 145–152 (2001)
34. Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. *J. of the Optical Society of America A* 4, 629–642 (1987)
35. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* 48, 443–453 (1970)
36. Smith, T., Waterman, M.: Identification of common molecular subsequences. *Journal of Molecular Biology* 147, 195–197 (1981)

Embedding of the Extended Euclidean Distance into Pattern Recognition with Higher-Order Singular Value Decomposition of Prototype Tensors

Bogusław Cyganek

AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
cyganek@agh.edu.pl

Abstract. The paper presents architecture and properties of the ensemble of the classifiers operating in the tensor orthogonal spaces obtained with the Higher-Order Singular Value Decomposition of prototype tensors. In this paper two modifications to this architecture are proposed. The first one consists in embedding of the Extended Euclidean Distance metric which accounts for the spatial relationship of pixels in the input images and allows robustness to small geometrical perturbations of the patterns. The second improvement consists in application of the weighted majority voting for combination of the responses of the classifiers in the ensemble. The experimental results show that the proposed improvements increase overall accuracy of the ensemble.

Keywords: Pattern classification, ensemble of classifiers, Euclidean Distance, IMED, HOSVD.

1 Introduction

This paper is an extension of our previous work on development of the image classification with the ensemble of tensor based classifiers [4]. The method showed to be very robust in terms of accuracy and execution time, since many existing methods do not account for the multi-dimensionality of the classified data [19][21][22].

Processing and classification of the multi-factor dependent data can be addressed with help of methods operating with tensors and their decompositions. One of the pioneered methods from this group is the face recognition system, coined tensor-faces, proposed by Vasilescu and Terzopoulos [22]. In their approach tensors are proposed to cope with multiple factors of face patterns, such as different poses, views, illuminations, etc. Another tensor based method for handwritten digits recognition was proposed by Savas *et al.* [17][15]. Their method assumes tensor decomposition which allows representation of a tensor as a product of its core tensor and a set of unitary mode matrices. This decomposition is called Higher-Order Singular Value Decomposition (HOSVD) [1][14][11]. A similar approach was undertaken by Cyganek in the system for road signs recognition [3]. In this case, the input pattern tensor is built from artificially generated deformed versions of the prototype road sign

exemplars. All aforementioned systems, which are based on HOSVD, show very high accuracy and high speed of response. However, computation of the HOSVD from large size tensors is computationally demanding since the algorithm requires successive computation of the SVD decompositions of matrices obtained from tensor flattening in different modes [13]. In practice, these matrices can be very large since they correspond to the products of all dimensions of the input tensor. In many applications this can be very problematic. To overcome this problem an ensemble with smaller size pattern tensor was proposed by Cyganek [4]. In the proposed methods tensors are of much smaller size than in a case of a single classifier due to the bagging process. However, despite the computational advantages, the proposed ensemble based method shows better accuracy when compared to a single classifier.

In this paper two modifications to the previously presented method are proposed. The first one is embedding of the Extended Euclidean Distance metric, recently introduced by Wang *et al.* [23]. This allows robustness to small geometrical perturbations of the input patterns since the new metric accounts for the spatial relationship of pixels in the input images. The second improvement consists in application of the weighted majority voting for combination of the responses of the classifiers in the ensemble. The experimental results show that in many cases the proposed improvements allow an increase of the overall accuracy of classification.

The rest of the paper is organized as follows. In section 2 properties of the Euclidean Image Distance are presented. In Section 3 the architecture of the proposed ensemble of the HOSVD multi-classifiers is discussed. Pattern recognition by the ensemble of the tensor classifiers is discussed in section 4. Experimental results are presented in section 5. The paper ends with conclusions in section 6.

2 Embedding Euclidean Image Distance

Images are 2D structures in which a scalar, vector (color) or multi-dimensional (MRI) value of a pixel is as important as its position within image coordinate space. However, the second aspect is not easy to be accounted for due to geometrical transformation of images of observed objects. On the other hand, image recognition heavily relies on comparison of images for which the Euclidean metric is the most frequently used one, mostly due to its popularity and simplicity in computations. However, Wang *et al.* proposed a better metric than Euclidean which takes into account also spatial relationship among pixels [23]. The proposed metric, called IMage Euclidean Distance (IMED), shows many useful properties, among which the most important is its insensitivity to small geometrical deformations of compared images.

More specifically, instead of the Euclidean metric between the two images \mathbf{X} and \mathbf{Y} of dimensions $M \times N$ each, given as follows

$$D_E(\mathbf{X}, \mathbf{Y}) = \sum_{k=1}^{MN} (x^k - y^k)^2 = (\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y}), \quad (1)$$

Wang *et al.* propose to use the following extended version

$$D_G(\mathbf{X}, \mathbf{Y}) = \sum_{k,l=1}^{MN} g_{kl} (x^k - y^k)(x^l - y^l) = (\mathbf{x} - \mathbf{y})^T \mathbf{G}(\mathbf{x} - \mathbf{y}), \quad (2)$$

where \mathbf{x} and \mathbf{y} are column vectors formed by the column- or row-wise vectorization of the images \mathbf{X} and \mathbf{Y} , respectively, and g_{kl} are elements of the symmetric nonnegative matrix \mathbf{G} of dimensions $MN \times MN$, which defines the metric properties of the image space.

Thanks to the above formulation, information on spatial position of pixels can be embedded into the distance measure, through the coefficients g_{kl} . In other words, the closer the pixels are, the higher value of g_{kl} should be, reaching its maximum for $k=l$. The distance between pixel positions (not values) is defined on an integer image lattice simply as a function of the 'pure' Euclidean distance between the points, as follows

$$g_{kl} = f(|\mathbf{P}_k - \mathbf{P}_l|) = \frac{1}{2\pi\sigma^2} e^{-\frac{|\mathbf{P}_k - \mathbf{P}_l|^2}{2\sigma^2}}, \quad (3)$$

where $\mathbf{P}_i = [p_i^1, p_i^2]^T$ denotes position of the i -th pixel in the image, while σ is a width parameter, usually set to 1 [23]. Finally, incorporating (3) into (2) the IMED distance among image \mathbf{X} and \mathbf{Y} is obtained, as follows

$$D_{IMED}(\mathbf{X}, \mathbf{Y}) = \frac{1}{2\pi\sigma^2} \sum_{k,l=1}^{MN} e^{-\frac{(p_k^1 - p_l^1)^2 + (p_k^2 - p_l^2)^2}{2\sigma^2}} (x^k - y^k)(x^l - y^l). \quad (4)$$

The D_{IMED} image metric given in (4) can be used for a direct comparison of images, such as in the case of the k -nearest neighbor method, etc. It can be also incorporated into other classification algorithms, such as the discussed HOSVD. This can be achieved substituting D_{IMED} into all places in which the D_E was used.

However, for large databases of images direct computation of (4) can be expensive. An algorithm to overcome this problem was proposed by Sun *et al.* after observing that computation of D_{IMED} can be equivalently stated as a transform domain smoothing [18]. They developed the Convolution Standardized Transform (CST) which approximates well the D_{IMED} . For this purpose the following separable filter was used

$$\mathbf{H} = \mathbf{h} \otimes \mathbf{h}^T = \mathbf{h}\mathbf{h}^T, \quad (5)$$

where \otimes denotes the Kronecker product of two vectors \mathbf{h} and \mathbf{h}^T with the following components

$$\mathbf{h} = \begin{bmatrix} 0.0053 & 0.2171 & 0.5519 & 0.2171 & 0.0053 \end{bmatrix}^T. \quad (6)$$

The filter \mathbf{h} given by (6) was also used in our computations since it offers much faster computations than direct application of (4).

Fig. 1 shows examples of application of the Standardizing Transformation for selected pictograms of the road signs in implementation with the filter \mathbf{h} in (6). It is visible that ST operates as a low-pass filter (lower row). This way transformed patterns are fed to the classifier system.



Fig. 1. Visualization of Standardizing Transform applied to the road sign pictograms. Original pictograms (upper row). After transformation (lower row).

Finally, it should be noticed that the IMED transformation should not be confused with the Mahalanobis distance or the whitening transformation [6][5]. Specifically, in equation (2) we do not assume computation of any data distribution nor probabilistic spaces. In other words, the main difference lies in definition of the matrix \mathbf{G} in (2) which elements, given by (3), convey information on mutual positions of the points. In contrast, for the Mahalanobis distance \mathbf{G} would be an inverse of the covariance matrix which elements are computed directly from the values of \mathbf{x} and \mathbf{y} disregarding their placement in the images.

3 Architecture of the Ensemble of HOSVD Multi-classifiers

Multidimensional data are handled efficiently with help of the tensor based methods since each degree of freedom can be represented with a separate index of a tensor [1][2]. Following this idea, multidimensional training patterns can be efficiently represented by a prototype tensor [3]. For the purpose of pattern recognition the prototype patterns tensor can be further decomposed into the orthogonal components which span a prototype tensor space. For the decomposition the Higher-Order Singular Value Decomposition can be used [1][13][11]. This way obtained orthogonal bases are then used for pattern recognition in a similar way to the standard PCA based classifiers [5][20][21].

The HOSVD method allows any P -dimensional tensor $\mathcal{T} \in \Re^{N_1 \times N_2 \times \dots \times N_m \times \dots \times N_n \times \dots \times N_P}$ to be equivalently represented in the following form [13][14]

$$\mathcal{T} = \mathcal{Z} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \dots \times_P \mathbf{S}_P, \quad (7)$$

where \mathbf{S}_k are $N_k \times N_k$ unitary mode matrices, $\mathcal{Z} \in \Re^{N_1 \times N_2 \times \dots \times N_m \times \dots \times N_n \times \dots \times N_P}$ is a core tensor. \mathcal{Z} fulfills the following properties [13][14]:

1. (Orthogonality) Two subtensors $\mathcal{Z}_{n_k=a}$ and $\mathcal{Z}_{n_k=b}$ for all possible values of k for which $a \neq b$ it holds that

$$\mathcal{Z}_{n_k=a} \cdot \mathcal{Z}_{n_k=b} = 0. \quad (8)$$

2. (Energy) All subtensors of \mathcal{Z} for all k can be ordered according to their Frobenius norms, as follows

$$\|\mathcal{Z}_{n_k=1}\| \geq \|\mathcal{Z}_{n_k=2}\| \geq \dots \geq \|\mathcal{Z}_{n_k=N_P}\| \geq 0, \quad (9)$$

The a -mode singular value of \mathcal{T} is defined as follows

$$\|\mathcal{Z}_{n_k=a}\| = \sigma_a^k. \quad (10)$$

An algorithm for computation of the HOSVD is based on successive computations of the SVD decomposition of the matrices composed of the flattened version of the tensor \mathcal{T} . The algorithm requires a number of the SVD computations which is equal to the valence of that tensor. The detailed algorithm can be referred to in the literature [1][13][14].

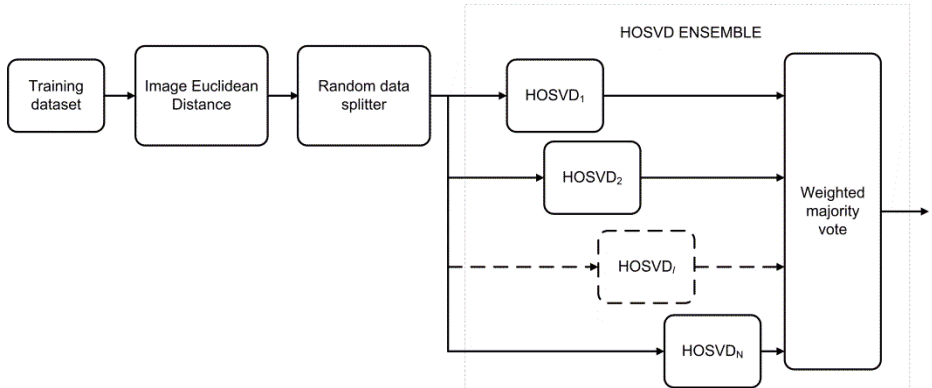


Fig. 2. Architecture of the ensemble with the HOSVD classifiers. Data preprocessed with the Image Euclidean Transformation. Bagging method used for training. Outputs are combined with the weighted majority voting.

Fig. 2 shows architecture of the proposed ensemble of the HOSVD classifiers. All training and testing data are preprocessed with the Image Euclidean Transformation described in section (2). Then, each HOSVD is trained with only a partition of the training dataset obtained in the bagging process.

In the next step, accuracies of each of the classifiers in the ensemble are assessed using the *whole* training dataset. These are then used to compute the weights of the classifiers in the ensemble. In the run-time their outputs are combined with the weighted majority voting scheme, as will be described in the next section.

4 Pattern Recognition by the Ensemble of Tensor Classifiers

It can be observed that thanks to the commutative properties of the k -mode tensor multiplication [17], the following sum can be constructed for each mode matrix \mathbf{S}_i in (7)

$$\mathcal{T} = \sum_{h=1}^{N_p} \mathcal{T}_h \times_p \mathbf{s}_p^h. \quad (11)$$

In the above the tensors

$$\mathcal{T}_h = \mathcal{Z} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \dots \times_{P-1} \mathbf{S}_{P-1} \quad (12)$$

form the basis tensors, whereas \mathbf{s}_p^h denote columns of the unitary matrix \mathbf{S}_p . Because each \mathcal{T}_h is of dimension $P-1$ then \times_p in (11) is an outer product, i.e. a product of two tensors of dimensions $P-1$ and 1. Moreover, due to the orthogonality properties (8) of the core tensor \mathcal{Z} in (12), \mathcal{T}_h are also orthogonal. Hence, they can constitute a basis which spans a subspace. This property is used to construct a HOSVD based classifier.

In the tensor space spanned by \mathcal{T}_h , pattern recognition can be stated as a measuring a distance of a given test pattern \mathbf{P}_x to its projections into each of the spaces spanned by the set of the bases \mathcal{T}_h in (12). This can be written as the following minimization problem [17]

$$\min_{i, c_h^i} \left\| \underbrace{\mathbf{P}_x - \sum_{h=1}^H c_h^i \mathcal{T}_h^i}_{Q_i} \right\|^2, \quad (13)$$

where the scalars c_h^i denote unknown coordinates of \mathbf{P}_x in the space spanned by \mathcal{T}_h^i , $H \leq N_p$ denotes a number of chosen dominating components.

To solve (13) the squared norm Q of (13) is created for a chosen index i . Assuming further that \mathcal{T}_h^i and \mathbf{P}_x are normalized the following is obtained (the *hat* mark indicates tensor normalization)

$$\rho_i = 1 - \sum_{h=1}^H \left\langle \hat{\mathcal{T}}_h^i, \hat{\mathbf{P}}_x \right\rangle^2. \quad (14)$$

Thus, to minimize (13) the following value needs to be maximized

$$\hat{\rho}_i = \sum_{h=1}^H \left\langle \hat{\mathcal{T}}_h^i, \hat{\mathbf{P}}_x \right\rangle^2, \quad (15)$$

Thanks to the above, the HOSVD classifier returns a class i for which its ρ_i from (15) is the largest.

Table 1. Structure of a matrix of partial accuracies for each classifier and each training prototype pattern

Digit	0	1	2	3	4	5	6	7	8	9
HOSVD ₀	p_{00}	p_{00}	p_{00}	...						
HOSVD ₁	p_{10}	p_{11}	p_{12}	...						
HOSVD ₂	p_{20}	p_{21}	p_{22}	...						
...						

In this work also different fusion methods were tested. Especially, the majority voting scheme was substituted for the weighted majority vote [10][16]. As alluded to previously, we proposed to use bagging to train the HOSVD classifiers from the ensemble which allows efficient memory usage. However, the partitions used for bagging contain less exemplars than all available for each prototype pattern. Therefore we further propose to use the whole training dataset to test each classifier trained with only fraction of that dataset for recognition of each pattern. This way we can assign some weight accuracies p_{kl} for each classifier k and for each trained class l . These are defined as follows

$$p_{kl} = \frac{N_{TP}^l}{N_{TP}^l + N_{FP}^l}, \quad (16)$$

where N_{TP}^l denotes a number of true positive responses and N_{FP}^l false positives, respectively. Table 1 visualizes this process for ten training patterns, such as digits.

Further, it is assumed that the classifiers are independent and each is endowed with its individual accuracies p_{kl} . If their outputs are combined with the weighted majority voting scheme, then accuracy of the ensemble is maximized by assigning the following weights [12][9]

$$b_{kl} = \log \frac{p_{kl}}{1 - p_{kl}}, \quad (17)$$

where p_{kl} are given by (16). On the other hand, for a test pattern \mathbf{P}_x each of the HOSVD classifiers in the ensemble responds with its class and assigned vote strength, as follows

$$d_{kl} = \begin{cases} \hat{\rho}_{kl}, & \text{if } \text{HOSVD}_k \text{ labels class } l \\ 0, & \text{otherwise} \end{cases}, \quad (18)$$

where $\hat{\rho}_{kl}$ denotes a maximal value of $\hat{\rho}_i$ in (15) and for the l -th classifier in ensemble and for pattern class $k=i$. Finally, the following discriminating function is computed

$$g_k(\hat{\mathbf{P}}_x) = \log(P_k) + \sum_{l=1}^L d_{kl} b_{kl}, \quad (19)$$

where P_k denotes the prior probability for the k -th class. However, the latter is usually unknown, so in the rest of experiments the first term in (19) was set to 0.

5 Experimental Results

The presented method was implemented in C++ using the HIL library [2]. Experiments were run on the computer with 8 GB RAM and Pentium® Quad Core Q 820 (clock 1.73 GHz).

For the experiments the USPS dataset was used [8][24]. The same set was also used by Savas *et al.* [17], as well as in the paper [4]. This dataset contains selected and preprocessed scans of the handwritten digits from envelopes of the U.S. Postal Service. Fig. 3 depicts some digits from the training and from the testing sets, respectively. The dataset is relatively difficult for machine classification since the reported human error is 2.5%. Therefore it has been used for comparison of different classifiers [15][17]. Originally the test and train patterns from the ZIP database come as the 16×16 gray level images.

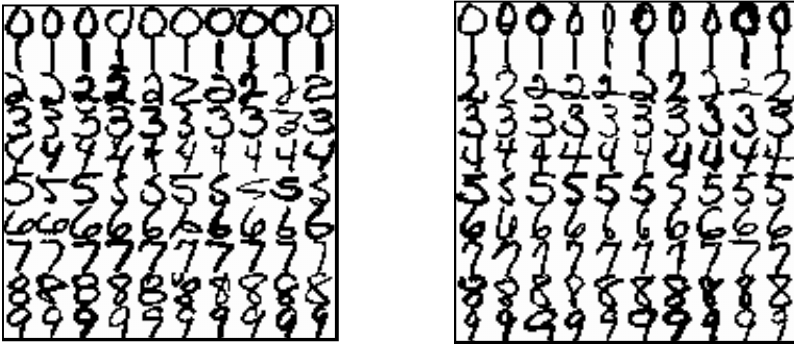


Fig. 3. Two data sets from the ZIP database. Training set (a), and testing set (b).

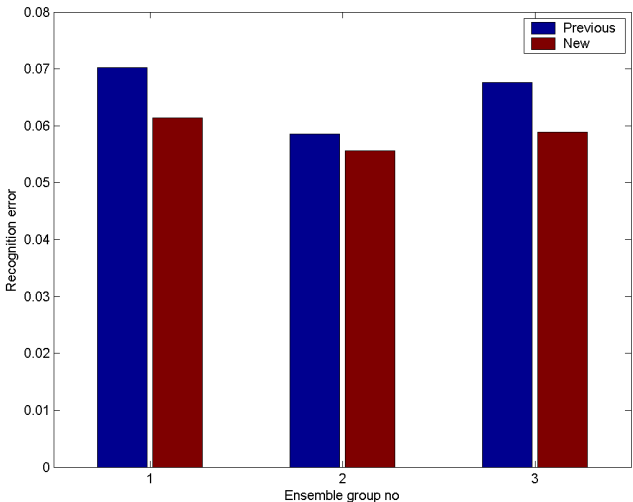


Fig. 4. Comparison of the accuracies of the three different ensemble settings without and with modifications proposed in this paper (the lower, the better). Blue bars for reference settings from [4]. Red bars relate to the method proposed in this paper, i.e. with the IMED and weighted majority voting.

The database is divided into the training and testing partitions, counting 7291 and 2007 exemplars, respectively.

Each experimental setup was run number of times and an average answer is reported. In all cases the Gaussian noise was added to the input image at level of 10%, in accordance with the procedure described in [2]. An analysis of different combinations of data partitions, number of classifiers in ensemble, number of dominating components, as well as input image size is discussed in our previous work [4]. In this paper we choose the best settings described in experimental results of the mentioned paper [4] and tested influence of the new IMED based preprocessing method, as well as new output fusion method. Results for three different settings are shown in the bar graph in Fig. 4.

For the experimental setups in this work were chosen three best setups from our previous work [4]. These are summarized in Table 2.

Table 2. Three best experimental setups of the ensemble from [4]

No.	Param.	Number of experts	Data in samples	Important components	Image resolution	Noise [%]
1		11	64	16	16x16	10
2		15	192	16	32x32	10
3		33	64	16	16x16	10

The first experiments were run with configurations of the ensembles from Table 2. Then new propositions of IMED and weighted majority voting were introduced and run again. Each experiment setup was run 25 times and average parameters are reported. In all cases the proposed modifications allowed better results of 0.2-1% with negligible time penalty due to separability of the IMED filter.

6 Conclusions

In this paper an extended version of our previous work on image classification with the ensemble of tensor based classifiers is presented [4]. In this method, thanks to the construction of the ensemble of cooperating classifiers, tensors are of much smaller size than in a case of a single classifier. Each classifier in this ensemble is trained with data partition obtained from bagging. Such approach allows computations with much smaller memory requirements. However, the method shows also better accuracy when compared to a single classifier. In this paper two modifications to this formulation were discussed. The first is to apply input pattern preprocessing with embedding of the Extended Euclidean Distance metric. This allows robustness to small geometrical perturbations thanks to the metric which accounts for the spatial relationship of pixels in the input images. The second improvement consists in application of the weighted majority voting for combination of the responses of the classifiers in the ensemble. The experimental results show that in many cases the proposed improvements allow an increase of the overall accuracy of classification in order of 0.2-1%. The method is highly universal and can be used with other types of patterns.

Acknowledgement. The work was supported in the years 2011-2012 from the funds of the Polish National Science Centre NCN, contract no. DEC-2011/01/B/ST6/01994.

References

1. Cichocki, A., Zdunek, R., Amari, S.: Nonnegative Matrix and Tensor Factorization. *IEEE Signal Processing Magazine* 25(1), 142–145 (2008)
2. Cyganek, B., Siebert, J.P.: *An Introduction to 3D Computer Vision Techniques and Algorithms*. Wiley (2009)
3. Cyganek, B.: An Analysis of the Road Signs Classification Based on the Higher-Order Singular Value Decomposition of the Deformable Pattern Tensors. In: Blanc-Talon, J., Bone, D., Philips, W., Popescu, D., Scheunders, P. (eds.) *ACIVS 2010, Part II. LNCS*, vol. 6475, pp. 191–202. Springer, Heidelberg (2010)
4. Cyganek, B.: Ensemble of Tensor Classifiers Based on the Higher-Order Singular Value Decomposition. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) *HAIS 2012, Part II. LNCS*, vol. 7209, pp. 578–589. Springer, Heidelberg (2012)
5. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*. Wiley (2001)
6. Fukunaga, K.: *Introduction to Statistical Pattern Recognition*, 2nd edn. Academic Press (1990)

7. Grandvalet, Y.: Bagging equalizes influence. *Machine Learning* 55, 251–270 (2004)
8. Hull, J.: A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(5), 550–554 (1994)
9. Jackowski, K., Woźniak, M.: Algorithm of designing compound recognition system on the basis of combining classifiers with simultaneous splitting feature space into competence areas. *Pattern Analysis and Applications* 12, 415–425 (2009)
10. Kittler, J., Hatef, M., Duing, R.P.W., Matas, J.: On Combining Classifiers. *IEEE PAMI* 20(3), 226–239 (1998)
11. Kolda, T.G., Bader, B.W.: Tensor Decompositions and Applications. *SIAM Review*, 455–500 (2008)
12. Kuncheva, L.I.: *Combining Pattern Classifiers. Methods and Algorithms*. Wiley Interscience (2005)
13. de Lathauwer, L.: *Signal Processing Based on Multilinear Algebra*. PhD dissertation, Katholieke Universiteit Leuven (1997)
14. de Lathauwer, L., de Moor, B., Vandewalle, J.: A Multilinear Singular Value Decomposition. *SIAM Journal of Matrix Analysis and Applications* 21(4), 1253–1278 (2000)
15. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE on Speech & Image Processing* 86(11), 2278–2324 (1998)
16. Polikar, R.: Ensemble Based Systems in Decision Making. *IEEE Circuits and Systems Magazine*, 21–45 (2006)
17. Savas, B., Eldén, L.: Handwritten digit classification using higher order singular value decomposition. *Pattern Recognition* 40, 993–1003 (2007)
18. Sun, B., Feng, J.: A Fast Algorithm for Image Euclidean Distance. In: *Chinese Conference on Pattern Recognition, CCPR 2008*, pp. 1–5 (2008)
19. Szeliski, R.: *Computer Vision. Algorithms and Applications*. Springer (2011)
20. Theodoridis, S., Koutroumbas, K.: *Pattern Recognition*, 4th edn. Academic Press (2009)
21. Turk, M., Pentland, A.: Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 3(1), 71–86 (1991)
22. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear Analysis of Image Ensembles: TensorFaces. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002, Part I. LNCS*, vol. 2350, pp. 447–460. Springer, Heidelberg (2002)
23. Wang, L., Zhang, Y., Feng, J.: On the Euclidean Distances of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(8), 1334–1339 (2005)
24. <http://www-stat.stanford.edu/~tibs/ElemStatLearn/>

DTW and Voting-Based Lip Print Recognition System

Piotr Porwik and Tomasz Orczyk

Department of Computer Systems, University of Silesia, Katowice, Poland
{piotr.porwik,tomasz.orczyk}@us.edu.pl

Abstract. This paper presents a method of lip print comparison and recognition. In the first stage the appropriate lip print features are extracted. Lip prints can be captured by police departments. Traces from lips may also found at a crime scene. The approach uses the well known DTW algorithm and Copeland vote counting method. Tests were conducted on 120 lip print images. The results obtained are very promising and suggest that the proposed recognition method can be introduced into professional forensic identification systems.

Keywords: Lip print, matching, voting, image pre-processing, DTW, Copeland's voting method.

1 Introduction

Today we now know that not only fingerprints, footprints or pistol bullets with their ballistic traits, but also hair, voice, blood and the fibers from clothes can be treated as criminal identifiers. Some of identifiers mentioned above are captured by police technicians directly from crime scenes. Latent lip prints can be also successfully collected [9, 10, 11, 13]. If properly captured and examined, the material left at a crime scene can contain useful data leading to personal identification. The factual materials collected can be successfully analyzed and recognized by the police or by medical professionals. Last year, a new type of identification trait was included in the practical repertoire of the police and the judiciary: the lip print. Lip prints are the impressions of human lips left on objects such as drinking glasses, cigarettes, drink containers and aluminum foils. The serious study of human lips as a means of personal identification was publicized in the 1970's by two Japanese scientists – Yasuo Tsuchihasi and Kazuo Suzuki [2, 3,4]. The forensic investigation techniques in which human lip prints are analyzed are called a cheiloscopy [1,3]. The area of interest focused on by cheiloscopy is the system of furrows (lines, bifurcations bridges, dots, crossings and other marks) on the red part of human lips [1,2,3]. The uniqueness of lip prints makes cheiloscopy especially effective whenever appropriate evidence, such as lipstick blot marks, cups, glasses or even envelopes, is discovered at a crime scene. Even though lip print analysis is not substantially developed, this new form of identification is slowly becoming accepted and being introduced into practice all over the world. Nowadays, technique of lip print image analysis can be used for human identification, for example when a post mortem identification is needed [10]. It should be emphasized that cheiloscopy has already been successfully used as an evidence in lawsuits [12, 13]. Unfortunately, use of the lip prints in criminal cases is often limited because

the credibility of lip prints has not been firmly established in many courts. It should be mentioned here that a literature review shows that there are already a small number of works in which lip print technology (acquisition, automatic capture, analysis and recognition) have been mathematically described or practically utilized. Lip prints of a person are unique and are quite similar to finger imprints. Additionally, similar to fingerprint patterns, lip prints possess the following specific properties: permanence, indestructibility and uniqueness [1,2,3]. Lip prints are determined by one's genotype, and are therefore unique and stable throughout the life of a human being. In the solution preferred by forensic laboratories, lip imprints are captured using special police materials (white paper, a special cream and a magnetic powder). This technique is also preferred by police researchers. Here, in the first step, the lip print is rendered onto a durable surface using a fingerprint powder (a different type of powder can be used depending on the surface type). Special backing tapes or specialized papers (cards) can be used as a background surface. The image, thus developed, is then converted into a digital image by a scanner. This is the method of capture that we utilized during our investigations, as it allows for the lip images to capture more precise textures. These techniques have been described in a number of papers [2, 6-13]. The recognition results reported in the aforementioned papers have now been significantly improved.

2 Lip Print Pre-processing

For the proper preservation of evidence, lip imprints should be carefully captured and then copied to appropriate materials by means of a special cream and a magnetic powder. The process of capturing an image of the lips starts from one corner of the mouth and ends at the opposite site of the mouth, thus all the details of the lips being copied onto a surface. In successful prints, the lines on the lips must be recognizable, not smudged, neither too light nor too dark, and the entire upper and lower lip must be visible. In the next step, this trace needs to be digitalized because a digital lip print preparation is absolutely necessary for the subsequent techniques. Lip imprints often have many artefacts: fragments of skin, hairs and others, all of which generate unwanted noise. For this reason, the lip print image, I , should be enhanced and all of its artefacts removed. Many lip print feature extraction algorithms have been reported in the literature [1, 6-13]. In this paper, a new approach to lip print extraction and analysis is presented. The proposed method includes introduction of a sequence involving image normalization, lip pattern identification and feature extraction. The normalization procedure allows for the standardization of lip print images; it involves a horizontal alignment and then the separation of the image into the upper and lower lip. Lip pattern extraction separates the furrows from their background, thus forming the lip pattern. The main goal of the feature extraction is to convert the lip pattern image into projections (specialized histograms) which can then be compared using the DTW algorithm [5].

2.1 Image Normalization

In the first stage, after obtaining the original lip image I of dimension $M \times N$, its histogram is stretched, thus a new image, I^{new} , is obtained.

Let $I = \{p_i\}_{i=1}^{M \cdot N}$ be a set of pixel values p_i which form the image I , and $p_i \equiv I(x, y) \in \{0, \dots, L\}$. Given this, the histogram stretching operation is performed via a simple mathematical operation:

$$I^{new}(x, y) = \frac{255}{\max - \min} (I(x, y) - \min), \quad (1)$$

where:

min = minimum value of the elements in the set I ,
max = maximum value of the elements in the set I .

After stretching all source images are gray-scale images, black pixels have assigned a value of 0 and white pixels – a value of 255. Other pixels take intermediate values. The process of determining the lip area in a lip print image consists of several steps (Fig. 1). In the first step, the background of the original image should be removed. This can be performed by the following operation:

$$I^*(x, y) = \begin{cases} 255 & \text{for } I^{new}(x, y) > 180 \\ I^{new}(x, y) & \text{otherwise} \end{cases}. \quad (2)$$

So pixels with a value greater than a threshold level (180) are converted into the colour white, while the remaining pixel values remain unchanged (Fig. 1a). Next, a median filter with a 7×7 mask is used to blur the image I^* (Fig. 1b). The median filter has been chosen as it is much better at preserving sharp edges than a mean filter. This blurring operation forms the image I^{**} . In the last step, a binary image is prepared. The process of forming a binary image is:

$$I_{BIN}(x, y) = \begin{cases} 0 & \text{for } I^{**}(x, y) = 255 \\ 1 & \text{otherwise} \end{cases}. \quad (3)$$

The final, binary image is shown as Fig. 1c.



Fig. 1. Steps of lip area detection: a) the image after evaluating pixels against a threshold, b) after blurring, c) the detected lip area

The upper and lower lip separation is determined by a curve that runs through the centre of the space between the lips (Fig. 2a). This designated curve divides the region of interest of the lip print area into an upper and a lower lip.

A straight line equation can be established by means of a linear regression. It should be noticed that the normalized image presented in Fig. 2b has been obtained from a high quality lip print. In practice, images captured can be of a poor quality –

especially when the image has been captured as trace directly from a crime scene. For such images some intractable artefacts of the background may be difficult to eliminate. This means that the image cropping can not be performed perfectly. This can be seen in Fig. 3 in which normalized images are presented. The first image (Fig. 3a) is a good quality image, is perfectly normalized and thus leads to a properly cropped imprint. This image could be, for example, captured in a police department under ideal conditions.

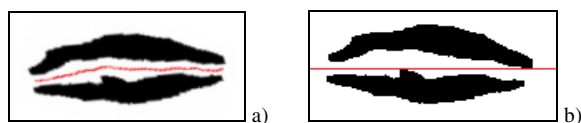


Fig. 2. Upper and lower lip separation line

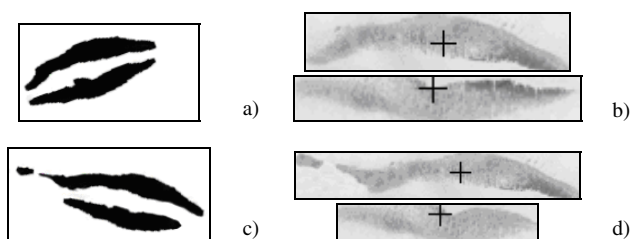


Fig. 3. Lip prints of the same person a) perfectly captured, b) the normalized and cropped image, c) lip print with artefacts and d) the imperfect print normalized and cropped

2.2 Lip Pattern Extraction

In the first stage of lip pattern extraction, the original and directionally normalized lip print images are smoothed. This process aims to improve the quality level of the lines forming the lip pattern. This smoothing can utilize convolution filtering. In the proposed solution, special smoothing masks have been designed, each of 5×5 pixels. These masks are depicted in Fig. 4. Elements of the each mask can take one of two binary values, 0 or 1. The white squares (elements) of the mask have the value of 0. The mask's other elements have the binary value of 1.

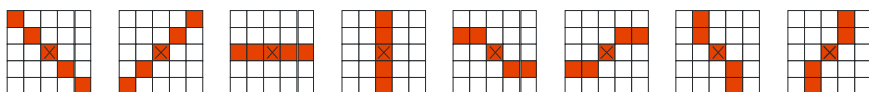


Fig. 4. Masks used in the lip print image smoothing procedure with their central points

Within the image, the masks in Fig. 4 traverse the image from its upper left corner, being applied to each pixel of the lip image. It should be noted that no element of the

mask is placed outside the source lip print image I . To calculate the value of the central point of the mask, here marked with a cross, the values of the pixels of the original image I covered by the mask are convolved with the mask's elements.

Let the eight smoothing masks be denoted as S^J , where $J = 1, \dots, 8$. Then, for each position of the mask S^J , the value of its central pixel can be calculated:

$$p^J = \sum_{a=0}^4 \sum_{b=0}^4 I(x+a-2, y+b-2) \cdot S^J(a, b). \quad (4)$$

Because the area of the mask S^J must always be located entirely inside the image I , the procedure (4) begins at the point $x = y = 2$ of image I . For each pixel, the values p^J of the mask J are successively calculated via the convolution of each mask with the image I . For each pixel, the mask S^b with the largest cumulative value is selected:

$$b = \arg \max_J \{p^J, J = 1, \dots, 8\}. \quad (5)$$

The value $p^b / 5$ now replaces the appropriate element of the image $I(x, y)$.

The convolution procedure (4) is repeated for each pixel of the image I over the range $x \in [2, M-2]$ and $y \in [2, N-2]$. The effect of the image smoothing within the region of interest is shown in Fig. 5b. To compare the results obtained, fragments of the selected images are shown both before and after the smoothing procedure.

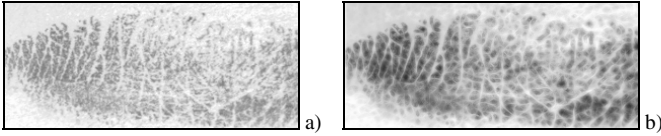


Fig. 5. The result of lip prints smoothing: a) a fragment of a lip print before smoothing, b) the same fragment after smoothing

In the second stage, the top-hat transformation is applied to the lip print image. This transformation extracts highlights (the small elements and details in a given image). In practice, the image is processed by each structural element and the processed image then subtracted from the original image. In the proposed approach, flat, disk-shaped structural elements were applied [9]. The purpose of this procedure is to emphasize the lines of the lip pattern and to separate them from their background. To increase the effectiveness of this algorithm, this transformation was applied twice using two different sizes of structural masks. Two masks were used: a 5×5 mask to highlight thin lines (of up to 3 pixels); and an 11×11 mask to highlight the thick lines (of more than 3 pixels). The effects of these operations are depicted in Fig. 6a and 6b. After the top-hat (T-H) transformation, the image I^{T-H} is obtained.

In the subsequent stage of the lip pattern extraction process, the image is converted into a binary representation. This procedure involves the application of a formula (6) to each of the two images resulting from the top-hat transformation. For the image

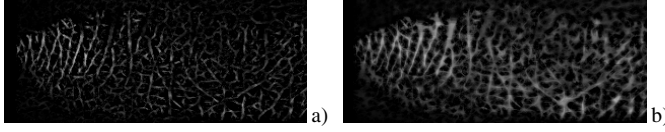


Fig. 6. Fragment of the lip print after the top-hat transformation: a) the thin lines emphasised (the 5×5 mask), b) the thick lines emphasised (the 11×11 mask)

with the thin lines emphasised, the conversion threshold value was set to be $t = 15$, while for the thick line image this parameter became $t = 100$.

$$I_{BIN}(x, y) = \begin{cases} 1 & \text{for } I^{T-H}(x, y) > t \\ 0 & \text{for } I^{T-H}(x, y) \leq t \end{cases} \quad (6)$$

The effect of the conversion of the lip print image into a binary representation is seen in Fig. 7.

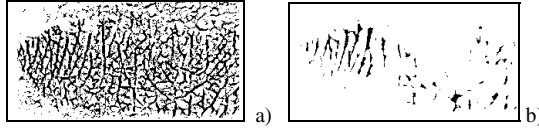


Fig. 7. Fragment of the lip print converted into a binary representation: (a) the thin lines emphasised (threshold of 15), (b) the thick lines emphasised (threshold of 100)

In the last stage, the sub-images of the thin and thick lines are combined into one single image, and this unified image is de-noised. For the noise reduction, appropriate 7×7 dimensional masks have been designed. Each of the 20 different masks is depicted in Fig. 8. Each mask, M_J , where $J = 1, \dots, 20$, consists of a different number of elements e_i as indicated on Fig. 8. The two masks M_6 and M_{11} consist of $e_5 = e_{11} = 5$ elements. Each of the other masks has 7 elements. Each element of each mask can take any value because these values are not important. Only the direction of the mask's elements needs to be considered. To simplify the mathematical formulas, the elements e_j of each mask M_J can take the value of 1. Thus, both, the image I_{BIN} and the masks M_J , have binary representation. Thus, each element of each mask M_J nominates corresponding pixels from the source image I_{BIN} to be counted.

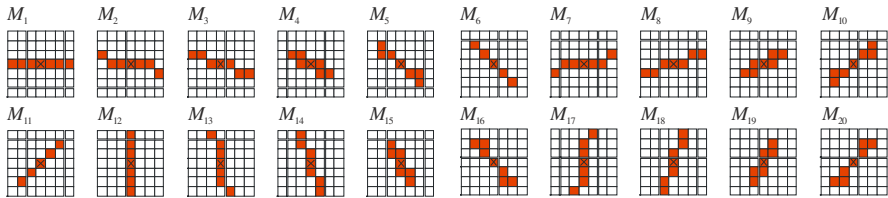


Fig. 8. Masks used for de-noising the binary image

The M_J masks are applied to each of the pixels of the source binary image I_{BIN} . Similarly to the previous masks, they are shifted from the left to the right and from the top of the lip print image I_{BIN} downwards. For each given mask, the black pixels of the image I_{BIN} that lie underneath the mask are counted. This process can be described more formally:

$$p_J = \sum_{a=0}^6 \sum_{b=0}^6 I_{BIN}(x+a-3, y+b-3) \cdot M_J(a, b). \quad (7)$$

If the number p_J of the black pixels of the image I_{BIN} associated with the number of elements in the mask M_J satisfies the condition $p_J < e_J$ then the analyzed pixel of the image $I_{BIN}(x, y)$ is converted to white. Otherwise the value of the pixel remains unchanged. The exemplary effects of this noise reduction method are shown in Fig. 9.

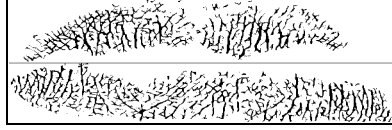


Fig. 9. Lip print image after the lip pattern extraction process

3 The DTW-Based Comparison of Lip Imprints

This process relies on a feature analysis and consists of determining the vertical, horizontal and diagonal projections of the lip pattern image. These projections create a special type of histograms.

Let the monochrome image I of dimension $M \times N$, consist of pixels p_i :

$$I = \{p_i\}_{i=1}^{M \cdot N}, \text{ where } p_i \equiv I(x, y) \in [0, 1]. \quad (8)$$

Let there be a set of projections:

$$A) \quad H_c^{0^0} = \text{card}\{(x, y) \in I : I(x, y) = 1 \wedge I(x, y) \in l_c\},$$

$$l_c : y = c, \quad c = 1, \dots, N, \quad (9)$$

$$B) \quad H_c^{90^0} = \text{card}\{(x, y) \in I : I(x, y) = 1 \wedge I(x, y) \in l_c\},$$

$$l_c : x = c, \quad c = 1, \dots, M \quad (10)$$

$$C) \quad H_c^{+45^0} = \text{card}\{(x, y) \in I : I(x, y) = 1 \wedge I(x, y) \in l_c\},$$

$$l_c : y = -x + c, \quad c = 1, \dots, \left\lfloor \sqrt{M^2 + N^2} \right\rfloor, \quad (11)$$

$$D) \quad H_c^{-45^0} = \text{card}\{(x, y) \in I : I(x, y) = 1 \wedge I(x, y) \in l_c\},$$

$$l_c : y = x + c, \quad c = 1, \dots, \left\lfloor \sqrt{M^2 + N^2} \right\rfloor, \quad (12)$$

where l_c is an appropriate straight line equation.

The directional projections of the example lip print image are shown in Fig. 10.

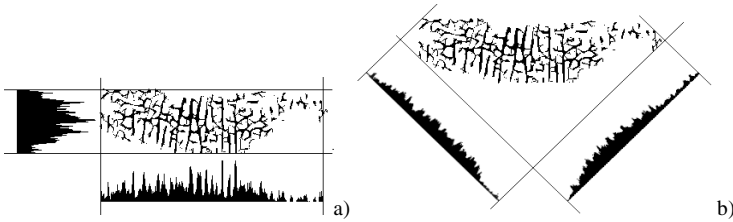


Fig. 10. Projections (histograms) obtained from an example lip pattern: (a) vertical and horizontal projections, (b) diagonal projections at angles of $+45^\circ$ and -45°

Each projection shows the number of black pixels that lie perpendicular to each axis direction: horizontal, vertical, and oblique at $\pm 45^\circ$. In order to evaluate the similarity between any two lip images, an appropriate measure of similarity needs to be defined. For image retrieval systems a variety of similarity measures and coefficients have been reported. It is known that the choice of similarity measure is related to the variability of the images within the same class. The method detailed in this paper demonstrates that similarity can be determined directly from the histograms produced. These histograms form discrete sequences composed of the histogram's elements H_c^a . Hence, the conformity of any two images can be determined by applying the DTW method. This approach was described in detail and successfully applied in a previous paper [7,9]. For the two images that are to be compared, first their histograms are generated. Each histogram can be treated as a series of numbers N_+ . In the first step, the local cost matrix C of the alignment of the two sequences A and B is formed:

$$C \in N_+^{M \times N} : c_{ij} = |a_i - b_j| \text{ and } a_i \in A, b_i \in B, i = 1, \dots, M, j = 1, \dots, N \quad (13)$$

Many different alignment paths can be created over matrix's points, from c_{11} to c_{MN} . The algorithm being summarized here finds the alignment path that runs through the low-cost areas of the cost matrix: see Fig. 11. The path with the minimal alignment cost will be referred to as the optimal alignment path. Let this path be denoted as L . Then:

$$\text{cost}(A, B) = \sum_{l=1}^L c_{ij} : c_{ij} \neq 0. \quad (14)$$

In other words, the similarity of two images can be defined as their appropriate histograms' matching cost. If the matching cost is lower, then the images are more similar. This procedure is performed for each type of projection (histogram) collected. For the

two images being compared four costs values are computed. The costs obtained are then summarized and their average values calculated. The best DTW paths matching projections from two different sample lip prints are graphically depicted in Fig.11.

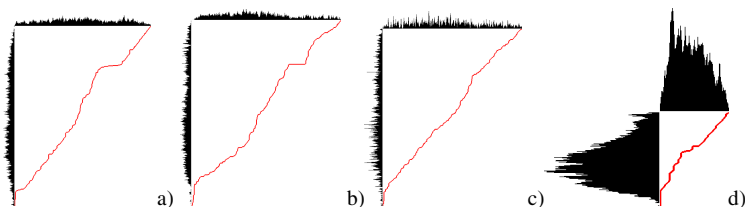


Fig. 11. Comparison of lip pattern projections using a DTW methodology: (a) oblique at $+45^\circ$, (b) oblique at -45° , (c) horizontal, (d) vertical

This biometric recognition system has been tested in a closed set of trials by the evaluation of the CMC curves. In our case, for proposed database, ERR factor achieved value $EER \approx 21\%$. The results obtained are presented in the chart of Fig. 12. Precise details of the methodology just described can be found in previously published work [9].

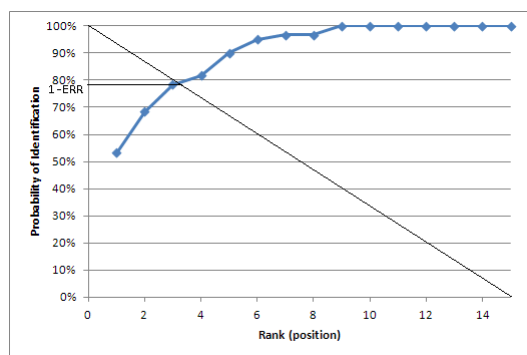


Fig. 12. The CMC curve with approximated 1-ERR point

It should be noted that lip recognition systems have not yet been properly and seriously investigated thus the literature on this topic is still very poor. For this reason, the results announced here must be regarded as good. Unfortunately, from a modern identification system's point of view, the recognition level achieved is still insufficient. For this reason, a modification of presented technique of image comparison has been introduced.

3.1 Modification of the Basic Method

Instead of summarizing and calculating the average matching cost of all collected histograms, each matching cost value can be individually analysed to build a rank list, which gives 8 rank lists for each lip print – 4 for upper lip, and 4 for lower lip. In the new approach three methods of building rank lists were analyzed:

- Class score is equal to the lowest (best) matching cost among class' samples.
- Class score is equal to the highest (worst) matching cost among class' samples.
- Class score is equal to the average matching cost of all class' samples.

According to other investigations reported [13,14] it is possible to utilize vote counting methods for rank level fusion of biometric data, so the rank lists based on matching cost of each of the 8 histograms are used as a voting ballots to elect a Condorcet winner in a single round of voting performed using the Copeland's method as a vote counting algorithm.

Basic Description of the Copeland Vote Counting Method.

Let are 3 classifiers and 4 known objects (A, B, C and D), stored in the database. A new object O_{new} should be classified. Each of the classifiers C_i , $i = 1, 2, 3$ separately points out the list, sorted in descending order, of objects from the database according to their similarity $M_i = \{sim(O_{new}, A), \dots, sim(O_{new}, D)\}$ to the object O_{new} . This idea is presented on Fig. 13a. In the next stage classifiers return the sorted, with descending order, database objects which are most similar to the object O_{new} .

Class	M 1	M 2	M 3
A	0,8	0,6	0,7
B	0,6	0,8	0,4
C	0,1	0,1	0,9
D	0,5	0,2	0,1

C 1	C 2	C 3
A	B	C
B	A	A
D	D	B
C	C	D

a)

Pair	Winner	Looser	Ratio
A, B	A	B	2:1
A, C	A	C	2:1
A, D	A	D	3:0
B, C	B	C	2:1
B, D	B	D	3:0
C, D	D	C	2:1

Class	Victories	Defeats	Points
A	3	0	3
B	2	1	1
D	1	2	-1
C	0	3	-3

b)

Fig. 13. Principles of the rank list creation process (a) and pair-wise contest Condorcet's vote counting method (b)

In the next step a list of all possible pairs (combinations without repetitions) of classes is created (Fig. 13b). Class which has a higher position in most rank lists gains a point. Class which has a lower position loses a point. If there is a tie no points are granted. This, so-called pair wise contest, is illustrated on Fig. 13b. The last step is sorting the classes list according to points they gained in the pair-wise contest.

Voting and Rank Technique

As in previous case the tests were performed using the histograms and DTW method on a closed set of samples using CMC curves, and estimation of EER. Recognition accuracy of this method was significantly improved, Fig. 14 shows that the best results were achieved when voting and rank lists based on average matching cost across the class (ERR= $\sim 11,5\%$) were applied. The second best result was performed for voting on rank list based on a worst class representative (sample with highest matching cost across the class) with ERR= $\sim 13\%$ and the worst results among voting

trials were achieved when using samples with lowest matching cost as a class representatives (ERR= $\sim 17,5\%$) but it was anyways a better result than when ordering classes on a base of an average DTW matching cost.

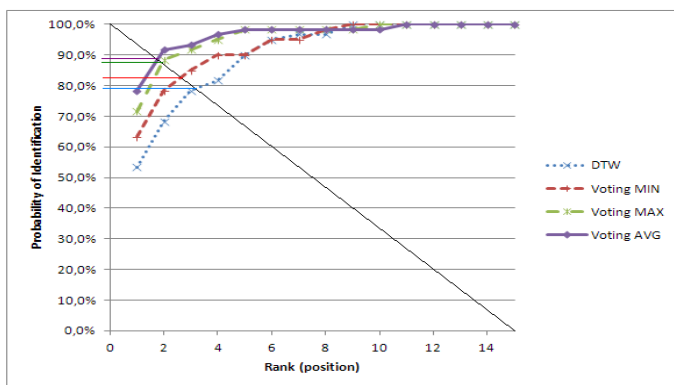


Fig. 14. CMC curves for modified algorithm

One thing worth noticing is the fact, that the basic DTW method and voting using best (minimal matching cost) class representatives have probability of identification of 100% at rank 9, when using voting with worst (maximal matching cost) class representative probability of identification of 100% is reached at rank 10 and when using class-average matching cost probability of identification of 100% is reached at rank 11.

4 The Results Obtained

A recognition technique measures some properties of an individual and stores that information as a template. This procedure is repeated for a number of individuals, and a database of templates is formed. We then need an identification technique to correctly discriminate between individuals. Two such techniques have been presented in this paper. In the study described here, 120 lip prints from 30 individuals (4 lip prints per person) were examined. These prints were entered into the database. Next, in a round-robin type procedure, one image was removed from the database and this extracted image was compared with all the remaining images in the database. From this dataset we used similarity scores (8 per one individual). As the voting results does not contain any score, just an order is important so it is not possible to introduce any threshold value to draw a ROC curves for identification in an open set environment, so the methods' efficiency was compared in a closed set environment using cumulative match characteristic curves (Fig. 14), where possibility of identification (PoI) and rank are defined as:

$$PoI = \frac{TP}{TP + FP} \cdot 100\% \quad \text{and} \quad rank = FP - 1 \quad (15)$$

where: TP – true positive match and FP – false positive match.

5 Conclusions

The results of experiments demonstrate that presented here model of biometric system and a new kind of similarity scores give a good lip print recognition level. As similarity scores the DTW together with Copeland voting technique have been applied. The process for collecting lip prints was the same procedure as that used in forensic laboratories for collecting lip prints from suspects. First, latent lip prints were collected on paper using a cheiloscopy stamp. The visible lip prints were then scanned and converted into grayscale images with a resolution of 300 dpi. Proposed studies make it clear that if lip features are appropriately captured, the lip print might become an important tool for identification. For example, this procedure can be applied during forensic investigations (post mortems, at crime scenes, in relation to medicine, etc.).

References

1. Kasprzak, J., Leczynska, B.: Cheiloscopy. Human Identification on the Basis of a Lip Trace, pp. 11–29 (2001) (in Polish)
2. Sharma, P., Saxwina, S., Rathod, V.: Comparative reliability of cheiloscopy and palatoscopy in human identification. *Indian Journal of Dental Research* 20(4), 453–457 (2009)
3. Tsuchihashi, Y.: Studies on personal identification by means of lip prints. *Forensic Science*, 127–231 (1974)
4. Dougherty, E.: An introduction to morphological image processing (1992)
5. Rath, T.M., Manmatha, R.: Word image matching using dynamic time warping. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 521–527 (2003)
6. Smacki, L., Wrobel, K.: Lip Print Recognition Based on Mean Differences Similarity Measure. In: Burduk, R., Kurzyński, M., Woźniak, M., Żołnierek, A. (eds.) *Computer Recognition Systems 4. AISC*, vol. 95, pp. 41–49. Springer, Heidelberg (2011)
7. Smacki, L.: Lip traces recognition based on lines pattern. *Journal of Medical Informatics and Technologies* 15, 53–60 (2010)
8. Smacki, L., Porwik, P., Tomaszewski, K., Kwarcinska, S.: Lip print recognition using the Hough transform. *Journal of Medical Informatics and Technologies* 14, 31–38 (2010)
9. Smacki, L., Wrobel, K., Porwik, P.: Lip print recognition based on the DTW algorithm. In: *Proc. of the Nature and Biologically Inspired Computing, Conf. Salamanca, Spain*, pp. 601–606 (2011)
10. Utsuno, H., et al.: Preliminary study of post mortem identification using lip prints. *Forensic Science International* 149(2-3), 129–132
11. Segui, M.A., et al.: Persistent lipsticks and their lip prints: new hidden evidence at the crime scene. *Forensic Science International* 112(1), 41–47 (2000)
12. Boole, R.M., et al.: *Guide to biometrics*. Springer Science+Buisnes Media Inc. (2004)
13. Kim, J.O., et al.: Lip print recognition for security systems by multi resolution architecture. *Future Generation Computer Systems* (20), 295–301 (2004)
14. Monwar, M., Gavrilova, M.L.: Robust Multimodal Biometric System using Markov Chain based Rank Level Fusion. In: *VISAPP 2010 - Proceedings of the Fifth International Conference on Computer Vision Theory and Applications, Angers, France*, vol. 2, pp. 458–463 (2010)

Advances in the Keystroke Dynamics: The Practical Impact of Database Quality

Mariusz Rybniak¹, Piotr Panasiuk², Khalid Saeed², and Marcin Rogowski³

¹ University of Bialystok, Bialystok, Poland

² AGH University of Science and Technology, Cracow, Poland

³ King Abdullah University of Science and Technology, Thuwal, Kingdom of Saudi Arabia

mariuszrybniak@wp.pl, {panasiuk,saeed}@agh.edu.pl,

marcin.rogowski@kaust.edu.sa

Abstract. This paper concerns database quality in the Keystroke Dynamics domain. The authors present their own algorithm and test it using two databases: the authors' own *KDS* database and *Keystroke Dynamics - Benchmark Data Set* online database. Following problems are studied theoretically and experimentally: classification accuracy, database representativeness, increase in typing proficiency and finally: time precision in samples acquisition. Results show that the impact of the database uniqueness on the experimental results is substantial and should not be disregarded in classification algorithm evaluation.

Keywords: keystroke dynamics, identification, authentication, behavioral science, biometrics, computer security.

1 Introduction

With the recent expansion of Internet and the constant development of social networks, a lot of sensitive personal data circulate in the worldwide web. Frequently it is important to maintain *data security*, by limiting the access to a specific trusted group of individuals. It is therefore essential to determine or confirm person identity. The task is known as *authentication* - determining if the specific person's identity conforms to its claim. *Reference authentication data* has to be stored inside an *authentication database*, in order to be compared with *authentication data* provided by the user. After positive *authentication* the user is granted access to the *sensitive data* or *services*. The difference between *authentication* and *identification* is that *identification* is to determine the user's identity, without a claim who it is. In the case the whole *authentication database* has to be searched and the best matching user may be given access (if his *authentication data* is trustworthy enough).

The traditional taxonomy of the human *authentication methods* has been proposed by H. Wood [1] and (after slight modifications) it distinguishes three groups of methods:

- **a proof by knowledge** - something that the user knows and remembers, such as passwords, PIN numbers, lock combinations, answers to secret questions;

- **a proof by possession** - a unique item (token) that the user possesses, e.g. keys, chip cards, magnetic cards, hardware or software tokens;
- **biometrics** - behavior or physical body properties unique for the user, such as fingerprints, signature, keystroke dynamics, eye retina pattern, hand shape, ear shape, etc.

Proof by knowledge is the most popular method of securing digital data, usually referred to as *passwords*. Regarding security it is important to create an efficient *multiple-use password* (as opposed to one-time passwords), which should follow three properties listed by Burnett and Kleiman [2]: *complexity*, *uniqueness* and *secrecy*. In practice, unfortunately, most users ignore at least one of the rules, e.g.: (i) a password is unique and complex, but written on an easily-accessed memo; (ii) a password is complex and secret but the same for every service; (iii) a password is unique and secret, but very simple to guess. As reported by Bruce Schneier [3] about 25% of the passwords can be guessed using a 1000-word dictionary with 100 common suffixes. Larger dictionary along with biographical data brings success rate to 55-65%.

Techniques that use *proof by possession* guarantee neither high security nor availability. As tokens are physical objects, they can be possibly handed over, stolen, misplaced or broken. If they are not secured by an additional password or a PIN code one can assume that the thief will easily access the sensitive data.

The *biometric methods* can be used for both *authentication* and *identification*. *Biometrics* is a science concerning measurements of living organism features. In the past few decades there was a noticeable increase in biometrics popularity, especially in the domain of *data security*. *Biometric methods* vary greatly in terms of uniqueness, classification accuracy and acceptability. Measured *features* can be classified on the basis of their origin as physical or behavioral features. *Physical features* are those that are derived from the way in which our body is built. The most popular and proved physical feature is fingerprint. Among physical biometric features one can also distinguish: face image, iris or retina scan, hand geometry. *Behavioral features* originate from a way user performs certain activities. The most known and the oldest behavioral biometric feature is handwritten signature. Examples of other behavioral features are: voice, gait and – the main subject of this paper – keystroke dynamics.

Keystroke dynamics, like *gait analysis*, has significant advantages over other biometric features. It is non-invasive, highly acceptable and it does not need specialized hardware (in its basic form). There are also some disadvantages of *keystroke biometrics*: (i) efficient features interpretation can be problematic; (ii) limitations of present Operating Systems can affect the data quality. Reference [4] correctly points out that the researchers often overlook an important disadvantage of many biometric methods – acceptability. Obtaining fingerprints or an iris scan may be considered insulting by some people.

Sometimes a particular biometric feature cannot be obtained from the user (e.g. a finger blessing altering the fingerprint), thus systems based on more than one feature are desirable. In [5] voice, hand geometry and face image are used together.

This paper is organized as follows: section 2 describes state of the art in *keystroke dynamics*, section 3 presents two databases: database *KDS* created by the authors and

Keystroke Dynamics - Benchmark Data Set [6] database available online, section 4 presents briefly the fundamentals of the authors' classification approach to *keystroke dynamics* (for details please refer to papers [7]-[9]), section 5 presents classification results and problems related to database quality regarding *keystroke dynamics*, finally section 6 concludes the paper.

2 State of the Art

Keystroke dynamics is a *behavioral biometric feature* that describes human *keyboard typing pattern*. This method is dated as far as the invention of the telegraph and its popularization in the 1860s [10]. *Keystroke dynamics* is not as accurate method as *fingerprint* pattern so it cannot be used for forensic purposes [11], as the method does not meet the European access control standards such as EN-50133-1. It specifies that FRR should be less than 1% and FAR should be no more than 0.001%. However, if one includes other features of typing, the keystroke dynamics results will definitely be improved. Similarly, as it is with handwritten signature, when existing (off-line) signature is analyzed, only two features are considered – its dimensions. On the other hand, when on-line signature is analyzed, additional features such as the pressure of the pen, its angle and the position in time can be extracted. This gives five features to analyze and improves accuracy significantly. A good idea is also to analyze the pressure of the keystroke.

Keystroke dynamics itself is not likely to give satisfying results, unless merged with some other biometric features, preferably non-invasive physiological ones in a multimodal system. An example of multifactor systems could be keystroke dynamics merged with face image recognition used to verify user identity while inserting PIN number at the ATM.

2.1 Latest Achievements and Other Possible Directions

Latest research focuses in general on the *user authentication* in order to secure personal computers. There are only a few works on the topic of *user identification*. Artificial Neural Networks are one of the most common tools for classification. The main disadvantage of ANN is the high dependence on the training database and high cost of retraining. Also, it is a *black-box model*, so no information about the specific attributes is available. Researchers mainly focus on the algorithms that are ready and known to work well, but in general the number of untested approaches is constantly decreasing.

With many of the algorithm ideas tested, researches started looking for new features that would improve the classification accuracy. One of the ideas is to use pressure sensitive keyboards. Microsoft is working on the hardware [12] and a student team contest was organized using the prototypes, searching for new ideas [13]. It is shown that *pressure* is even more important characteristics than the dynamics itself [14]. In [15] the authors constructed their own keyboard and used pressure as an additional feature, which turned out to be very helpful for the *user authentication*. This

should not surprise anyone since i.e. *on-line signature recognition* is generally more reliable than *off-line*. The results suggest that the use of *pressure information* would greatly help in *user identification*. The main problem with this approach is very low availability of pressure sensitive keyboards.

Some research has been done using mobile phone keyboards as input devices [16]-[18]. The motivation behind is the rising popularity of mobile phones and the fact that many users do not even use PIN to protect their devices. The proposed solution is to use *keystroke dynamics* to *authenticate* users as they type text messages and phone numbers. For the standard 9-key keyboard, both numerical and alphabetical inputs have been tested and the error rates are reported to be about 12.8% for 11-digit phone number [16] and 13% using fixed 10-character long alphabetical input [17]. Interestingly, for mobile version of QWERTY keyboard, dwell time for each key did not prove to be a reliable feature and only latency between keys was used [18]. Results were similar as for 9-key keyboard and the error rate was 12.2%.

ATM hardware was also considered [19], but rather than *keystroke dynamics*, keystroke motion and hand shape at different time points were analyzed and the results proved to be very good. Error rate achieved was as low as 1.1% to 5.7% depending on the PIN and exact features used. This approach requires a camera which records hands movements as the PIN is typed. It raises safety issues, as it is generally advised to hide hand movements while typing PIN.

2.2 Database and Sample Validity

The work [20] summarizes all major efforts in *keystroke dynamics* with attention put on database issues. The algorithms in the field are mostly developed using dedicated databases. The main problem is that all those various and 'specialized' databases are very difficult to compare. Some of them were collected in *supervised conditions*. In this case certain samples may be disregarded, i.e., the users who make a lot of mistakes or users that want to sabotage the experiment (by intentionally inserting unnaturally different samples). Samples are gathered with various amounts of characters. One cannot tell if a phrase is as good in discriminating user's identity as the other with the same length [8]. Some of the phrases also need pressing additional special keys in case of typing capital letters or diacritic characters. The size of the users' population matters greatly, especially with *identification* algorithms. Another issue is incomplete or corrected data. That leads to sample inconsistencies that may render the results unreliable. The event timing may be affected by OS clock process queuing. It was examined using arbitrary waveform generator [21] and reported that 18.7% of the keyboard events are registered with 200 μ s latency. However, while using typical PC, samples are limited in precision with OS event clock, which is limited to accuracy of 15.625 ms (64 Hz) using MS Windows and 10 ms using most Linux distributions.

Considering the constraints described above, the authors of [20] released their database online: *Keystroke Dynamics - Benchmark Data Set* that is very accurate and has many samples. The database is available online free of charge [6] and was used by the authors along their own *KDS database* to experimentally test database-related issues.

3 Database Classification

It has been shown that *keystroke dynamics authentication* results highly depend on the database quality [20], [22]. Viable algorithms should deal with noisy samples: the ones with typos or random pauses in user typing. Among the databases the authors can distinguish ones collected in a supervised way, meaning every test subject was individually instructed by a supervisor before the start of the samples acquisition process. The supervisor can also make notes on how the subject types and what influences him. It guarantees samples of good quality. This type of database, however, usually does not reflect real world situations. Databases may have accounts duplicated, for example if the user forgets his password or just wants to have multiple accounts. The typing pattern may be duplicated for two different classes, which may decrease the identification accuracy and in hybrid (rank-threshold) based verification methods it may even increase the FRR. Typing with unnatural manner can also increase FAR.

Another factor is the purpose for which the database is gathered. Authentication requires user ID attached to *keystroke data*. Simulation of hacking requires the same text typed by many users. Passwords are usually short phrases often consisting additional characters like capital letters (that involve *shift* key), dots, semicolons, numbers and symbols. For *identification* samples should be preferably longer, as this application is more complex.

There can be two additional approaches to *keystroke data acquisition*. The first is based on a *fixed text*. The second way is to use *free-text authorization* [22] to continuously monitor user's workstation while trying to *authorize* him/her. There are the following problems with *free-text authorization*: (i) how often *user authentication algorithm* should be run, (ii) more difficulty with data collection, (iii) more samples are needed for learning of the recognition algorithm. Potential noise can be a unique feature that helps to recognize users, so removing it completely – without deeper analysis – would be a loss of valuable information.

3.1 KDS Database Description

The authors' *keystroke dynamics database (KDS database)* was gathered in non-supervised conditions using JavaScript web-browser platform [23]. It is therefore OS independent and globally available, however at a cost of unpredictable latency. Data from over 400 users and total of over 1500 samples is stored in the database.

KDS database is unique, as it stores additional meta-information like *user's name, age, sex, hand used while writing* and *estimated proficiency with keyboard*. This additional information could serve for other purposes than authentication. The samples consist of five phrases, different among language versions (Polish and English). Uppercase and lowercase letters, special characters and key modifiers (Shift, Alt) are registered. The first phrase is a popular sentence, in English it is "To be, or not to be, that is the question. The second phrase is a tongue twister; in English it is "Shy Shelly says she shall sew sheets." The third phrase is an example of simple password: short Polish word: "kaloryfer". The fourth phrase is a user-chosen sentence. The fifth

phrase is a Psylock (commercial keystroke dynamics solution) password “After some consideration, I think the right answer is:” [24].

3.2 Keystroke Dynamics – Benchmark Data Set Database Description

Keystroke Dynamics - Benchmark Data Set database [6] was used for the reference. It was gathered in supervised conditions from 51 subjects, using an external high precision clock. Sample acquisition was divided into eight sessions, 50 samples each. Each user had to type a phrase “.tie5Roanl” 400 times. The data acquisition sessions were separated by at least 24 hours. The database was used to test 14 published classifiers [20]. The database is especially useful for testing *fixed-text* algorithms. It is time-accurate, has a reasonable number of users and many samples per user.

4 The Authors’ Approach

In this section, the authors describe their approach to identification, operating on *fixed-text* samples. Main goal is to compare the results obtained with the two above-mentioned databases. The authors use *k-Nearest Neighbor classifier*, so k value is chosen and a *training dataset* is built, where the amount of samples per user cannot be less than k . The remaining user samples are assigned to the *testing dataset*. The authors’ latest approach [8] was to calculate initial weights for all expected key events. However, during tests with *Keystroke Dynamics - Benchmark Data Set* it turned out that the classification results are better with the use of the former algorithm [7]. The possible explanation of this phenomenon is given in section 5.

Absolute times are processed into *flight times* and *dwell times*. *Flight times* are the times between releasing one key and pressing another. *Dwell time* is the time when a key is in the pressed state. The reason the authors convert *event times* into those two characteristics is because they are more stable. When the user makes a mistake or hesitates on some key, this would only affect the next two keys and not all the remaining times. The distances between samples are calculated using Manhattan metrics between corresponding *keyboard event times*.

Partial distances for two given samples were calculated using Manhattan distance (for corresponding *dwells* and *flights*), as specified in (1) and (2), where d_d is the partial dwell distance, d_f is the partial flight distance, d_{1i} and d_{2i} are the i -th dwells for 1st and 2nd samples, respectively, d_d is the partial dwell distance, f_{1i} and f_{2i} are the i -th flights for 1st and 2nd samples, respectively.

$$d_d = \sum_{i=1}^n |d_{1i} - d_{2i}| \quad (1)$$

$$d_f = \sum_{i=1}^m |f_{1i} - f_{2i}| \quad (2)$$

The total distance d between the two samples is calculated as in (3), where p is the ratio of importance of the *flight time* compared to the *dwell time*.

$$d = p * d_f + (1 - p) * d_d \quad (3)$$

Both *flight* and *dwell* are important as the authors presented in [7]. In the previous experiments the authors had determined the best p ratio value as 0.7. However, due to use of the different database, the authors decided to use the arbitrary value of 0.5.

After the calculation of all distances k samples are labeled with the training author ID and assigned a *rank*. The authors evaluate only *closed-world case*. Among all the results the authors take the k best ones and then conduct *voting* procedure on users (as described in detail in [7]). The shortest distance gets the highest score of k , the longest distance gets the lowest score of 1. The *winner* is the user with the greatest sum of scores.

5 Experimental Results

5.1 On Classification Accuracy

The authors have tried many varieties of combinations of their algorithm, while using the same amount of users in both databases, number of characters in a phrase and amount of training samples. In both experiments the training data sets were created using random samples, $k=2$, training set containing 6 samples for each user and 51 classes. Fig. 1 shows the results of this comparison where the *flight-to-dwell importance* is presented in horizontal axis. As can be seen, the classification results are significantly different. It leads to the claim that the results are incomparable even if the authors test the same algorithm in similar conditions. This supports the conclusion from [20] that the results obtained by research teams on their own databases may be incomparable.

5.2 On Database Representativeness

The main issue with databases collected in the supervised conditions is that they do not refer to the real-world conditions and therefore may lead to false results. Watching a user may be frustrating and lead to the acquisition of corrupted samples. Supervised acquisition, however, eliminates samples intentionally counterfeited. Real-world samples are sometimes corrupted. In *Keystroke Dynamics - Benchmark Data Set* there seem to be no corrupted samples. When using this database samples written with mistakes should be therefore rejected, as they probably cannot be classified.

Keystroke Dynamics - Benchmark Data Set is accurate and has a large amount of samples. It is perfect for testing algorithms for *user authentication*. However, when it comes to *user identification*, samples are too short to obtain satisfactory accuracy. Obtained accuracy of about 67% is satisfactory for such a short phrase and 51 classes.

5.3 On Increase of User's Typing Proficiency

In the *Keystroke Dynamics - Benchmark Data Set* users were asked to type 50 samples each time in 8 sessions separated one from another by at least 24 hours. The authors wondered how the learning process influences the results, so the authors tried to

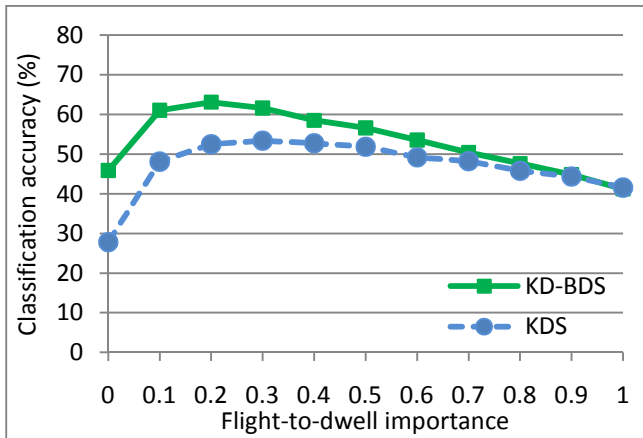


Fig. 1. The authors' approach classification rates, using two various databases, versus *flight-to-dwell importance* factor

test few *learning set* building algorithms. For the first experiment the authors took 8 random samples from all sessions per each user. Those samples should contain the best characteristics of the typing style of each user. In the second experiment the authors selected the first sample from each session. It means that there was at least 24 hour time span in acquisition between any of the samples from any single user used in the experiment. In the third experiment there were only the first 8 samples from the first session. This means that the user was not familiar with the password and has not developed the typing pattern yet. In the fourth experiment the last 8 samples from the last session were selected for training. It means that the users were well trained in typing the password. However, those are the last samples in the 50-sample session, so the users could be already tired. The authors always used 8 training samples per user profile and the authors set k value to 8 in our algorithm. In Fig. 2 one can see that the results vary a lot.

As one can conclude, using the samples collected *early* does not result in satisfactory accuracy. The characteristics obtained from them are differentiated, distorted by the fact that the user was still unfamiliar with the password. The first samples from each session also are not very good training dataset because the user had a long break between inserting them and they differ from the average user's characteristics. The last inserted samples are better, however, the user seemed to be tired typing so many samples and they may be not as stable as the samples from the middle of the session.

Many of *keystroke dynamics* methods are based on Artificial Neural Network (ANN) algorithms. The authors' experiments show that the first samples of the user are the noisiest ones, and using them to train the ANN yields poor results similar to those shown in [20].

The next problem the authors examined was the time of the sample typing. Fig. 3 presents the average time of phrase acquisition of 5 randomly selected users in each session. One can observe the gradual decrease in the mean time values.

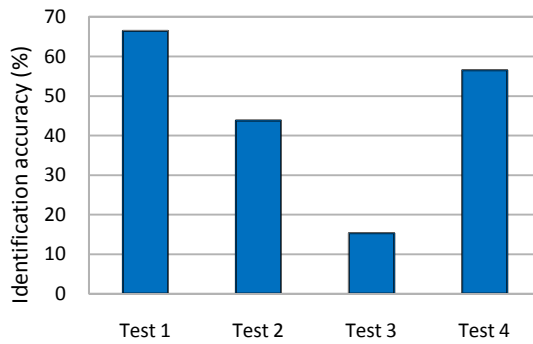


Fig. 2. Sample data selection using different methods. Test 1 – random samples. Test 2 – first sample from each session, Test 3 – first 8 samples from the first session, Test 4 – last 8 samples from the last session.

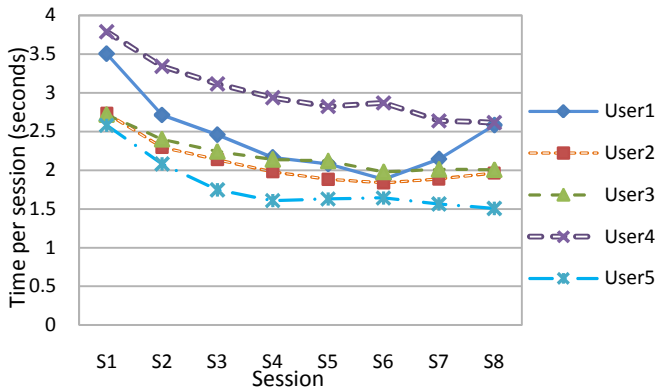


Fig. 3. The mean sample total typing time for five randomly selected users, versus sessions

The standard deviation of each keystroke decreases over time, as could be seen in Fig. 4 Gradually the users insert samples in a more consistent manner. Inner-class differences decrease, which helps in classification. It reduces FAR in verification systems.

5.4 On Time Precision in Samples Acquisition

In [21] keyboard functioning using 15MHz function and arbitrary waveform generator was examined. It was noticed that 18.7% of keystrokes were acquired with a $200\mu\text{s}$ error. Therefore, the keyboard was calibrated and the database was collected using higher precision. Data have been gathered with an accuracy of $100\mu\text{s}$. This experiment has shown that databases gathered using different machines may not be comparable because of *the lack of the main bus clock calibration*. Moreover, when

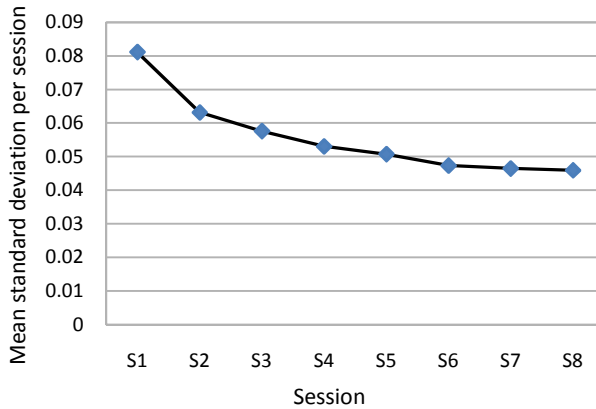


Fig. 4. The average of the keystrokes *standard deviation* for all samples, versus session number

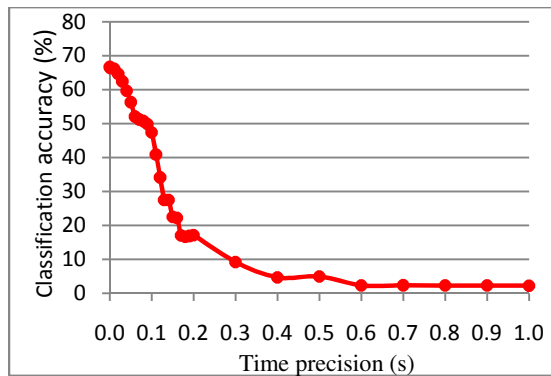


Fig. 5. Influence of time precision on algorithm's classification accuracy

CPU(s) is under load, delays in keystroke acquisition occur, as they are usually handled by *the message queue*. There is also a difference between the lengths of keyboard clock frames of Linux/Unix (*10ms* frame) and Windows (*15ms* frame, 64 ticks per second) operating systems. The influence of time resolution on the algorithm classification accuracy was tested using the authors' approach.

Fig. 5 shows that in the typically used time frames (*10-15 ms* – operating system clock intervals) there is no concern about data precision. As long as the resolution is smaller than *1ms* there is no significant difference in the algorithm classification accuracy. However, it is easily noticeable, with a resolution greater than around *20ms*, that the accuracy drops significantly. With precision of *0.6s* almost random results are obtained.

6 Conclusions

There has been a lot of research done in the field of keystroke dynamics during past decades. Many classification methods were researched in order to improve *keystroke dynamics* classification accuracy for both *authentication* and *identification* tasks. Obtained results are however hardly comparable due to the use of various database acquiring procedures and non-availability of databases.

The authors have tested two databases. The *KDS database* has worse time precision than *Keystroke Dynamics - Benchmark Data Set* due to the acquiring procedure performed over the Internet with use of JavaScript and web browser. It is however more universal as it could be used remotely (however user identity would not be guaranteed in this case). *KDS database* is more suitable for user *identification* because the samples are longer. It contains users' mistakes and corrections, what could be used in further experiments.

As the authors have indicated experimentally - using the same algorithm and conditions - specifics of *training database* affects *classification accuracy*. The samples obtained later with greater users' proficiency are of better consistency and distinguish users more reliably. Training set should also contain imperfect samples as they increase FRR error margin. An algorithm for updating the training set should be considered, as using only the initial samples would affect the classification accuracy.

The observations lead to the conclusion that the selection of database for the tests has the vital meaning for the reliability of results. The tests of classification algorithms should be run on the same database without any modifications and with a fixed *training dataset* building. If the conditions are not satisfied, the obtained results are hardly comparable with others.

Acknowledgements. This work was supported by AGH University of Science and Technology, grant no. 11.11.2010.01.

References

1. Wood, H.M.: The use of passwords for controlling access to remote computer systems and services. In: Proc. of the National Computer Conf., New York, pp. 27–33 (1977)
2. Burnett, M., Kleiman, D.: Perfect Passwords. Syngress, Rockland (2005)
3. Schneier, B.: Secure Passwords Keep You Safer (January 11, 2007), <http://www.wired.com/politics/security/commentary/securitymatters/2007/01/72458>
4. Li, X., Maybank, S.J., Yan, S., Tao, D., Xu, D.: Gait Components and Their Application to Gender Recognition. IEEE Trans. Syst. Man Cybern. C, Appl. Rev. 38(2), 145–155 (2008)
5. Veeramachaneni, K., Osadciw, L.A., Varshney, P.K.: An adaptive multimodal biometric management algorithm. IEEE Trans. Syst. Man Cybern. C, Appl. Rev. 35(3), 344–356 (2005)
6. Killourhy, K., Maxion, R.A.: Keystroke Dynamics - Benchmark Data Set (June 29, 2009), <http://www.cs.cmu.edu/~keystroke/>

7. Rybnik, M., Panasiuk, P., Saeed, K.: User Authentication with Keystroke Dynamics Using Fixed Text. In: IEEE-ICBAKE 2009 International Conference on Biometrics and Kansei Engineering, Cieszyn, Poland, pp. 70–75 (2009)
8. Panasiuk, P., Saeed, K.: A Modified Algorithm for User Identification by His Typing on the Keyboard. In: Choraś, R.S. (ed.) *Image Processing and Communications Challenges 2*. AISC, vol. 84, pp. 113–120. Springer, Heidelberg (2010)
9. Rybnik, M., Tabedzki, M., Saeed, K.: A keystroke dynamics based system for user identification. In: IEEE-CISIM 2008 – Computer Information Systems and Industrial Management Applications, Ostrava, Czech Republic, pp. 225–230 (2008)
10. Checco, J.C.: Keystroke Dynamics and Corporate Security. WSTA Ticker (2003)
11. CENELEC. European Standard EN 50133-1: Alarm systems. Access control systems for use in security applications. Part 1: System requirements, Standard Number EN 50133-1:1996/A1:2002, Technical Body CLC/TC 79, European Committee for Electrotechnical Standardization, CENELEC (2002)
12. Dietz, P.H., Eidelson, B., Westhues, J., Bathiche, S.: A practical pressure sensitive computer keyboard. In: Proc. of the 22nd Annual ACM Symposium on User Interface Software and Technology, New York (2009)
13. UIST. Student Innovation Contest Results (October 6, 2009), <http://www.acm.org/uist/uist2009/program/sicwinners.html>
14. Saevanee, H., Bhattarakosol, P.: Authenticating User Using Keystroke Dynamics and Finger Pressure. In: Consumer Communications and Networking Conference, Las Vegas, NV, pp. 1–2 (2009)
15. Loy, C.C., Lai, W.K., Lim, C.P.: Keystroke Patterns Classification Using the ARTMAP-FD Neural Network. In: *Intelligent Information Hiding and Multimedia Signal Processing*, Kaohsiung, pp. 61–64 (2007)
16. Clarke, N.L., Furnell, S.M.: Authenticating mobile phone users using keystroke analysis. *International Journal of Information Security* 6(1) (2006)
17. Campisi, P., Maiorana, E., Lo Bosco, M., Neri, A.: User authentication using keystroke dynamics for cellular phones. *IET Signal Processing* 3(4) (2009)
18. Karatzouni, S., Clarke, N.L.: Keystroke Analysis for Thumb-based Keyboards on Mobile Devices. In: Venter, H., Eloff, M., Labuschagne, L., Eloff, J., von Solms, R. (eds.) *New Approaches for Security, Privacy and Trust in Complex Environments*. IFIP, vol. 232, pp. 253–263. Springer, Boston (2007)
19. Ogihara, A., Matsumura, H., Shiozaki, A.: Biometric Verification Using Keystroke Motion and Key Press Timing for ATM User Authentication. In: *International Symposium on Intelligent Signal Processing and Communications*, Tottori, Japan, pp. 223–226 (2006)
20. Killourhy, K.S., Maxion, R.A.: Comparing Anomaly-Detection Algorithms for Keystroke Dynamics. In: *Dependable Systems & Networks*, Lisbon, Portugal, pp. 125–134 (2009)
21. Killourhy, K.S., Maxion, R.A.: The Effect of Clock Resolution on Keystroke Dynamics. In: Lippmann, R., Kirda, E., Trachtenberg, A. (eds.) *RAID 2008*. LNCS, vol. 5230, pp. 331–350. Springer, Heidelberg (2008)
22. Panasiuk, P., Saeed, K.: Influence of Database Quality on the Results of Keystroke Dynamics Algorithms. In: Chaki, N., Cortesi, A. (eds.) *CISIM 2011*. CCIS, vol. 245, pp. 105–112. Springer, Heidelberg (2011)
23. Panasiuk, P.: Keystroke Dynamics System (March 15, 2012), <http://www.kds.miszu.pl>
24. Psylock (February 08, 2011), <http://www.psylock.com>

Advanced Intracardial Biosignal Processing

Marek Penhaker¹, Petr Klimes¹, Jakub Pindor¹, and David Korpas²

¹ VSB - Technical University of Ostrava, Faculty of Electrical Engineering and Computer Science, Ostrava, Czech Republic

² Silesian University, Faculty of Public, Policies Institute of Nursing, Opava, Czech Republic

{marek.penhaker, petr.klimes, jakub.pindor}@vsb.cz,
david.korpas@seznam.cz

Abstract. In this work deals about the efficient intracardial ECG analysis algorithm. The main focus was design an optimal detection method for intracardial signals based on essential intracardial measuring methods and basic principles of gathering and processing data from invasive catheters. Intracardial ECG analysis is further important step in heart behavior understanding, especially it's electric manners. Detailed signal description generated in heart, together with heart function knowledge can provide us with an useful information about heart's condition or it's diseases. Designed detection method is able to mark significant points in intracardial records, compute it's elemental parameters and important time intervals. Incurred algorithm was designed and tested on intracardial records provided by Cardiology Clinic of Hospital IKEM – Praha, and Electrophysiology Laboratory of Hospital Podlesí – Trinec.

Keywords: iECG, Detection, Filtering, Processing.

1 Introduction

By entering this work was, among other things, analysis and detection of significant characters in the intracardiac ECG recordings with subsequent processing of such information. Data from measurements of intracardiac ECG was provided in a hospital Podlesí Trinec. These are the signals measured at different types of system operations and subsequently Exported Cardiolab in *.txt. Each data set is also associated *.inf file that contains additional information about the record length, sampling frequency, number of leads, etc. The development application was trying to get signals of various parameters from different places with varying degrees of heart hampering to determine a success or approximate boundaries of functionality algorithm.

1.1 Source Intracardial Data

Data is stored in a file with the extension *.txt, and can therefore be viewed as a classic notepad in Windows. After the opening we see that this is a table of numbers, where each column represents a seduction. The measured signals were recorded with

a sampling frequency of 977Hz. It follows that for such a long signal for 30 seconds for each lead-recorded 29310 samples. If recorded three intracardiac leads, for example, the first of the CS catheter ablation, and two, that the total surface leads recorded with 15 leads, it means that the file of the thirty-second measurement has 439,650 samples, or figures in the table. This comprehensive text file on your computer takes up about 5MB of memory.

The benefits file with the *.txt is the fact that it is very easy to load into MATLAB as a multidimensional variable. Indexing, we were able to separate each lead, respectively. vector and store it as a separate variable. Thus it can be done with all leads and each store separately under its own pre-chosen name. This prepared data from individual leads are ready for further processing.

2 Signal Preprocessing

In this work, the data are processed intracardiac leads see. Figure 1 shows the loaded signal from the ablation catheter accurately record the first 2.5 seconds. Signals from the chambers, which are clearly visible sign precedes the P wave, a signal from the hall. This wave, along with other manifestations of tissue, however, at this point redundant information. Our goal is to become independent and to detect ventricular signal.

For recognition of deformed QRS complex need to adjust the signal first. In the first phase of minor cropping potential and then using the wavelet transform.

Setting the threshold at the loaded signal seen in Figure 2 Red lines show the threshold values that are set both in positive terms and in the negative. Everything in this limited area is then ignored and reset the chart. However, it is necessary for this operation to maintain an identical length of the signal. It is therefore a direct replacement of values below the threshold at zero.

In the algorithm shown below is copied into a variable with a length of the original signal is modified in the wake.

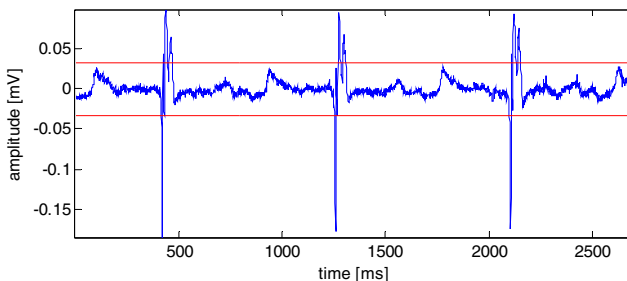


Fig. 1. From the loaded signal is displayed intracardiac catheter ablation of seduction

The next step is transformation modified signal using wavelets. Specifically, the selected wavelet bior1.5. As explained in section 4.3, determine the type of waves

preceded the experimental procedure is based on knowledge parameters of wavelets and its use in other applications. Subsequent levels of decomposition setting, or scale, using this formula was based on wavelets for detecting QRS complexes in the surface ECG 02.05.

Finishing the signal is conducted continuous wavelet transform (CWT), wavelets bior1.5, with the fourth level of decomposition. The result of this transform shown in Figure 2.

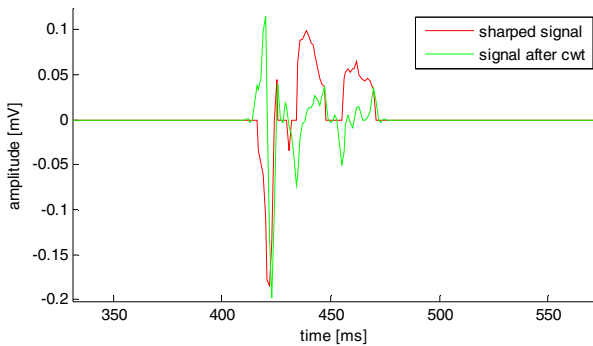


Fig. 2. Adjustment signal continuous wavelet transform - CWT, bior1.5. For clarity, only the first pick was detected excitation signal. It is clearer and shows how the signal continuous wavelet transform based on original changed shape.

2.1 Finding Three Significant Characters in the Intracardiac Recordings

Trimming curve prepared in advance and continuous wavelet transform is now available for the final stage of detection. This is a clear target indication of significant peaks that are detected primarily ventricular cycles and finding the beginnings and ends of these impulses. As already mentioned in Chapters 3 and 4, our task is not interested to follow more closely the morphology of the curve. Valuable data are kept in the information exactly where on the timeline is detected peak of excitement, when the excitement begins and ends, respectively. length and its relationship to neighboring vzruchům detected. First of all, we must clearly identify the detected peak of excitement.

2.2 Finding the Detected Peak Impulse

After adjusting signal quality before building a relatively simple task. The signal is free from all the surrounding potentials which are not interesting for us as well is wavelet transform rid of sharp transitions and changed into a form that presents Figure 3. So the only thing left, and find places where the transformed curve intersects the zero. We know that at the same time when the original curve reaches its local maximum, then the detected peak of excitement.

The problem with this procedure is the possibility of intersection of two lines of zeros. The signal zero crossing can be upward or downward, or may go from negative to positive and vice versa. Is needed in this algorithm into account, otherwise it will only detect one case of two and a high probability of excitation peak detect.

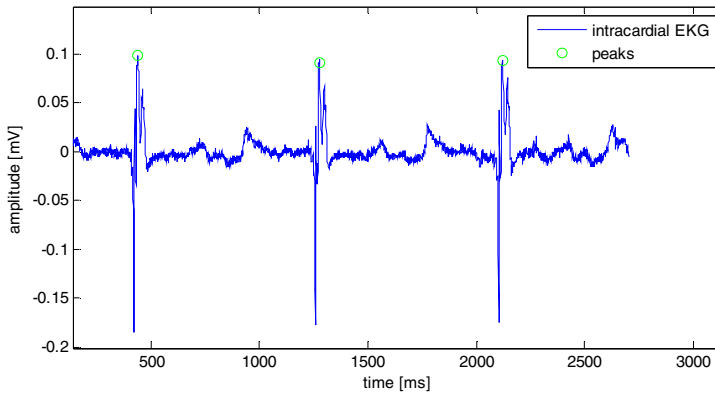


Fig. 3. Detected peaks intracardiac ECG electrical activity generated by ventricular

Another complication lies in the multiple detection. Figure 28 shows that the zero passage of the transformed signal below the top of the original passage is not only zero in the area. Algorithm to actually detect several peaks and is not clearly determined which of them is described as the pinnacle of the impulse, ie as a global summit. Assuming that it would be a simple curve with really only one local maximum to avoid the multiple detection. With this, however, can not count, especially with intracardiac leads that may have a completely unexpected

The solution to this problem is the second part of the displayed code, which provides only the label is always the highest, that is the most prominent peak, which will ensure compliance with the requirement to detect a global peak of excitement. S_{cwt} variable represents the signal after processing the first part of the code and wavelet transforms. In the first phase will split the signal into sections that define the detected impulses. With a condition that can be detected only one peak in each such segment, the algorithm looks for the one with the highest amplitude relative to other peaks found. The definition of this section is based on sampling frequency, while the assumption of a distribution of excitation. As a limit value for setting limits for cropping signal, this value may be modified according to user needs and current signal parameters.

In this way, peaks are detected, or excitation peaks, which are used in the subsequent detection of the beginning and end of the impulse as a starting point, clearly defining where the peak is located.

2.3 Finding the Beginning and End of the Detected Impulse

The next phase of the effort comes reliable indication of the place where the detected excitation begins and ends. Early attempts to design this part of the algorithm contributed to the final solution with satisfactory results. To better understand the description and the impasse of development.

The original idea comes from practice. The doctor looks at the curve and just see the place where the excitement begins. Identifies it as that of the relatively flat line curve begins to rise suddenly to higher values. Likewise, it is also able to detect an algorithm to detect the beginning of excitement, but it has to be defined in accordance with these requirements. It is not possible based on the projected shape of the curve, as is the case with the surface ECG. The shape can take various forms, it is necessary to build on this basic premise: excitement begins where the curve starts from a relatively flat line of steep climb to higher values.

To find this place is first necessary to map the section signal. Rather it is a section from the detected peak before the beginning of oscillation. The signals are then backward from this point marked points that are stored in a matrix for subsequent calculations. Not only do we know their coordinates, or location on the timeline, we also represent the value of the signal amplitude at a given point.

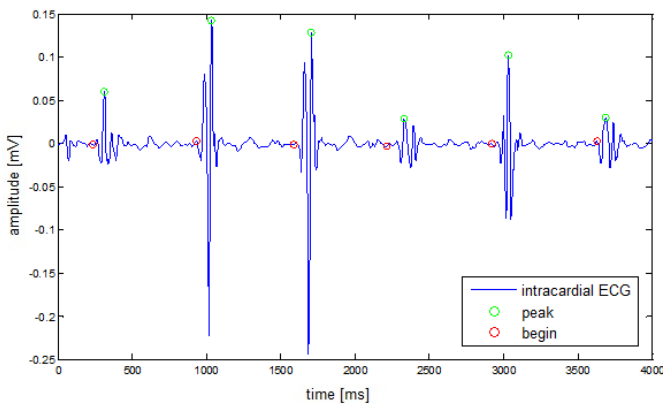


Fig. 4. Marking the beginnings of impulses detected by intracardiac ECG

This section mapped signal was then subjected to search for the smallest difference in amplitude between two neighboring points. The result of this comparison should be to find a flat end and the beginning of the steep climb. However, faced several problems. First and foremost was the smallest difference between points are often found at large distances from the actual beginning of the impulse, ie before it even began to stir. This was mainly due to the fact that the smallest difference amplitudes are often located away from the beginning of the excitement and not directly at him. The second complication occurred when the two adjacent points lie in the steep part, but one on the ascending and descending on the second, at about the same level. The algorithm then evaluate this case as a flat surface and named it as the beginning of excitement, but he saw a local maximum or minimum between these points and was

unable to recognize that the points lie on a steep curve. Including more surrounding pixels into these calculations, the problem is solved.

To ensure a really steep marking the beginning of excitation, respectively. place before the start of the steep change was designed additional code, to ensure early recognition steep based on a predetermined level, the signal must exceed before the impulse. Points are mapped in the first phase of this calculation converted to their absolute values. It is not conclusive, if the beginning is in positive or negative values. Elements are among the neighboring points. Subsequently, the detected maximum value of the matrix marked points. This value is then determined the level of 0.1%.

In the detection of all impulses to proceed similarly, but on the opposite side of the detected peak. At the same time there is an effort to find the sharp end, because the impulses are usually sharply terminated. In most cases disappear more slowly than the rise. The algorithm was therefore counted the detected displacement of the point which was found in the same manner as at the beginning of excitation.

In this way the algorithm was developed to detect peaks, start and end of detected impulses. Beginnings and endings are detected based on detecting peaks. Without a mark on top of the algorithm is able to delimit excitement. The experimental part of the algorithm was able to recognize signs and Q and S waves detected in the chamber complex. Due to the nature of the intracardiac signals, however, from this expansion was abandoned and attention was paid to quality notably through the detection of peaks of individual impulses.

3 Analysis Result

The algorithm was tested on approximately 75 different intracardiac signals from 15 different operations. This was how the signals from the diagnostic catheters, which provide about the best possible signal, measured directly in the heart, as well as therapeutic catheters, especially ablation, which is more therapeutic than hampering because of almost constant motion after cardiac walls.

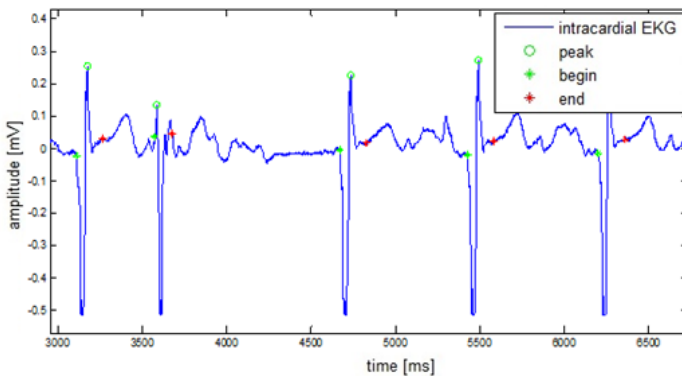


Fig. 5. Slightly hampering signal ablation catheter. Detection of peaks was flawlessly, beginnings and ends are detected for the second excitation reliably identified.

Detection of peaks achieved when the correct setting of limits almost total success even in noisy signals. In cases where the algorithm does not detect any peaks or, conversely, create false points, the curve was so hampering that there were doubts about the actual value of the information curve.

Identifying the beginnings and ends of the excitation peaks detected in the header pipes from the CS catheter almost flawless. For noisy signals success begins to deteriorate, and to mark the beginning is always the case. Only the position of determining the beginning point is not exactly at the point of beginning of excitation.

By classifying peaks, start and end of each detected impulse was obtained by specific dates, which carry information about the amplitude of excitation, the frequency, duration, etc. The processing of this information is not targeted to any particular application. This is a demonstration of versatility of use of the algorithm. Analyzed intracardiac ECG curve has been described and this allows the subsequent use of these data for experimental and other purposes.

4 Test Results and Evaluation by Analysis

Of all ten measured parameters were selected to assess three basic parameters of the signal and then three time intervals for cross checking the accuracy of detection.

Table 1 are measured values from the analysis of the fifteen intracardiac signals. The upper part is shown on the measurement of diagnostic catheters, the lower part presents the signals of therapeutic catheters. HR heart rate indicates the heart rate of total recording time. It is therefore an average value in units of beats / min. The amplitude of the detected impulse Amp informs about the average height of the signal amplitude in mV units.

Table 1. The measured values

<i>Signals from a diagnostic catheter EN</i>						
	<i>HR [bpm]</i>	<i>Amp [mV]</i>	<i>P - P [ms]</i>	<i>B - B [ms]</i>	<i>E - E [ms]</i>	<i>Hit Rate [%]</i>
1	72	1,024	838	838	838	100
2	72	0,5917	838	838	838	100
3	90	0,4053	667	667	667	97,93
4	87	0,4996	686	686	686	100
5	75	0,4566	798	798	798	98,2
<i>Signals from therapeutic catheter ABL</i>						
	<i>HR [bpm]</i>	<i>Amp [mV]</i>	<i>P - P [ms]</i>	<i>B - B [ms]</i>	<i>E - E [ms]</i>	<i>Hit Rate [%]</i>
1	74	0,2676	806	804	806	100
2	106	0,512	567	565	567	75
3	133	0,3798	452	453	452	95
4	133	0,3798	452	453	452	94,04
5	86	0,469	698	698	698	100
6	86	0,4296	698	699	698	91,3
7	86	0,6738	796	796	796	100
8	66	0,4559	910	910	910	86,36
9	85	1,7146	706	706	706	98,46
10	71	1,2949	846	846	846	95,38

The following three values of PP, BB, EE is dedicated to time intervals to assess the mutual time between detected points. This is the time interval between two adjacent peaks, the time interval between the starts of two adjacent impulses, and the interval between the end in units of ms.

The last column lists the values of hit rate achieved success at the individual signals. Word of success in this case think successfully found and marked points of the beginning, peak and end of detected impulses. This value was calculated based on detection by visual inspection for each of the records. The number of unsuccessfully marked or completely unmarked points and the total number of impulses detected catheter has been derived detection rate as a percentage.

5 Signal Processing Ranking

This value has a major impact on all previous calculations. If success is very small, we mean below 50%, does not have a previous column, too deal with. Measured and calculated data in such a case would probably not representative. Values above 90% success rate on the contrary we affirm the accuracy of the calculated parameters.

Percentage success was calculated based on visual inspection of each detected impulse. In the event that the detector has not identified any excitement, the points were its peak, the beginning and end labeled as defective. Likewise, if the detector is evaluated as a potential suitable to describe excitement, while the excitement was not in effect, create a false detection, and also this point is marked as defective. Subsequently, the bad points of detection compared with the total number of impulses, which had a detector to recognize and label.

In the event that marked all the impulses in the signal, its peaks, starts and ends at the same time did not create any false detection, the program has reached the maximum, a 100% success rate. In case of omission of some important points, or marking the wrong places, the success rate has declined in the worst case up to 75%.

Overall, the table shows that the measured signal quality diagnostic catheters is generally achieved a higher success rate, while the ablation catheter, which is often hard to read the signal, detection rate was worse.

Simultaneously with this measurement was performed peak detection statistics of success and detection of beginnings and endings, which is not included in the table. In this measurement is not take into account the type of signal. It did not matter then whether the signal from the therapeutic or diagnostic catheter is taken into account only well or poorly marked points under visual control again seen as successful or faulty. The resulting, average values of success are as follows:

The success of the algorithm for detection of significant characters in the intracardiac signals. Detection rate peaks: 98.85%, The success of detection of beginnings and endings: 95.05%, The success rate in detecting signals from diagnostic catheters: 99.22%, The success rate in detecting signals of therapeutic catheters: 93.55%.

6 Conclusion

The benefit of this work is a new approach to detect points in the ECG signals nonstandard shapes. It introduced an efficient method to recognize, label and border impulses, which are directly from the heart wall observed invasive catheters. This project was created to continue work to further improve the algorithm to detect and use this system to find deposits of arrhythmia in the heart, using calculations of intervals between impulses of two different signals in real time. For processing and design of detection algorithms were used MATLAB, in which they were implemented as mathematical signal processing operations, so statistical analysis of test results.

Acknowledgment. The work and the contribution were supported by the project: Ministry of Education of the Czech Republic under Project 1M0567 “Centre of Applied Cybernetics”, Student grant agency SV 4501141 “Biomedical engineering systems VII” and TACR TA01010632 “SCADA system for control and measurement of process in real time”. Also supported by project MSM6198910027 Consuming Computer Simulation and Optimization. This paper has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic.

References

- [1] Alfaouri, M., Daqrouq, K.: ECG signal denoising by wavelet transform thresholding. *American Journal of Applied Sciences* 5(3), 276–281 (2008)
- [2] Laguna, P., Jané, R., Caminal, P.: Automatic detection of wave boundaries in multilead ECG signal: Validation with the CSE Database. *Computers and Biomedical Research* 27, 45–60 (2004)
- [3] Vašíčková, Z., Penhaker, M., Augustynek, M.: Using Frequency Analysis of Vibration for Detection of Epileptic Seizure. In: Dössel, O., Schlegel, W.C. (eds.) WC 2009. IFMBE Proceedings, vol. 25/IV, pp. 2155–2157. Springer, Heidelberg (2009)
- [4] Krejcar, O., Janckulik, D., Motalova, L.: Complex Biomedical System with Mobile Clients. In: Dössel, O., Schlegel, W.C. (eds.) WC 2009. IFMBE Proceedings, vol. 25/V, pp. 141–144. Springer, Heidelberg (2009)
- [5] Prauzek, M., Penhaker, M.: Methods of comparing ECG reconstruction. In: 2nd International Conference on Biomedical Engineering and Informatics, Stránky, pp. 675–678. Tianjin University of Technology, Tianjin (2009) ISBN: 978-1-4244-4133-4, IEEE Catalog number: CFP0993D-PRT

Multi-constraints Face Detect-Track System

Hazar Mliki¹, Mohamed Hammami², and Hanène Ben-Abdallah¹

¹ MIRACL-FSEG, University of Sfax, 3018 Sfax, Tunisia
mliki.hazar@gmail.com,
hanene.benabdallah@fsegs.rnu.tn
² MIRACL-FS, University of Sfax, 3018 Sfax, Tunisia
mohamed.hammami@fss.rnu.tn

Abstract. This paper presents a new system to achieve face detection and tracking in video sequences. We have performed a combination between detection and tracking modules to overcome the different challenging problems that can occur while detecting or tracking faces. Our proposed system is composed of two modules: Face detection module and face tracking module. In the face detection module, we have used skin color and motion information to extract regions of interest and cut off false positive face. This filtering step has enhanced the next face tracking processing step, as it helps to avoid tracking false positive faces. Regarding tracking module, we have used face detection results to keep the face tracker updated. In order to carry on tracking face we have used particle filter technique which was adapted to track multiple faces. Moreover, each tracked face was described by a defined state: tracked, occluded, entered, left or stopped. The performance of our detect-track system was evaluated using several experiments. This evaluation proved the robustness of our face detection-track system as it supports automatic tracking with no need to manual initialization or re-initialization and reaches best performance to deal with different challenging problems.

Keywords: Face detection, face tracking, particle filter.

1 Introduction

Detecting and tracking faces with any view is an important problem since it has been observed that there is nearly 75% of faces in video are non-frontal [1]. This requires insuring permanent face spotting through the three degree of freedom of human head pose: yaw, pitch and roll. Thereby, detecting and tracking faces with any point of view is a very useful task mainly in visual surveillance applications, human computer interaction, facial biometric security solutions and driving assistance systems.

Actually, both automatic face detection and tracking have to deal with different problems. Concerning face detection task, it has to deal with various challenging problems such as: variation in pose, illumination, scale, and facial expression. With regard to tracking task there is no doubt that using pure tracking technique cannot be applied in real world scenarios since it reveals some deficiencies namely:

- Initialization problem: as soon as the new face enters the scene, we ought to provide this information to the tracker in order to start a new track.
- Lost track problem: when a face has left the scene, we have to update the tracker.
- Occlusion problem: this is a major problem since people tend to walk and interact in groups with others which increase the chances of having partial or complete occlusion.

Many previous studies have been reported on tracking face through video frames. Relying on the face representation, we can classify tracking methods into three main approaches: point-based approach, kernel-based approach and silhouette-based approach. Point-based approach represents the detected face as a set of relevant points. The association of the points is based on the previous face state [2, 3]. This approach requires face detection in every frame which is not always possible since the head is naturally moving within its three degree of freedom, this can made the detection task rough. Nevertheless, it can be used when the face is moving in certain narrow degree of rotation. The category of kernel-based approach is based on face appearance representation, namely templates and density-appearance models. In fact, Template matching is a brute force method which goes over the whole frame, looking for a similar region to the object template defined in the previous frame [4]. Instead of templates, other face representations can be used like color histograms [5] or mixture models [6] which can be computed by using the appearance of pixels inside the rectangular or ellipsoidal face region. The chief goal of a silhouette-based tracker is to find the face region in each frame using a face model generated in the previous frames. This face model can take the form of face shape or face contour. At respect to these face models, silhouette trackers can be divided into two sub-categories, namely, shape matching approaches and contour tracking approaches. Shape matching approaches search for the face silhouette in the current frame [7, 8]. However, the contour tracking approaches raise an initial contour to its new position in the current frame [9, 10].

In this paper, we propose an integrated detect-track system that can overcome the foremost problems usually emerging while detecting and tracking faces. Our proposed approach is composed of two main steps: Face detection step which was widely detailed in [18] and tracking step which will be depicted in this paper. Both of these two steps are running continually in each frame so that they complement each other very well to improve face localization in video sequences. In fact, when detection fails to detect face, tracking will secure face location; conversely when the tracker fails to update the tracker, the detection step will keep the tracker informed if a face has entered, left or reappeared. Concerning detection step, our proposed approach was based on the affluent combination of spatial (skin) and temporal (motion) information in video to filter out and restrict regions of interest. Regarding Tracking step we have used particle filter technique which is well known for its capability to deal naturally with problems arising from non-linearity and non-normality of the motion model under many tracking scenarios. Moreover, we have classified each tracked face into five states: tracked, stopped, entered, left, or occluded, this will help understanding person behavior in the scene. To evaluate our proposed system for face detection and

tracking in video, we have appraised several experiments. Our main contributions can be abridged in two points: adapting the system to detect and track multiple faces without scarifying real time performance and handling multiple complex challenging problems related to detection and tracking tasks.

The rest of the paper is organized as follows: We begin in the next section, by detailing our proposed detect-track system. The results of several experiments and performance evaluation are presented in section 3. We conclude this paper with remarks on potential extensions for future work.

2 Proposed Approach

Figure 1 illustrates our proposed approach for automatic detection and tracking faces in video. It is composed of two successive modules: face detection module and face tracking module.

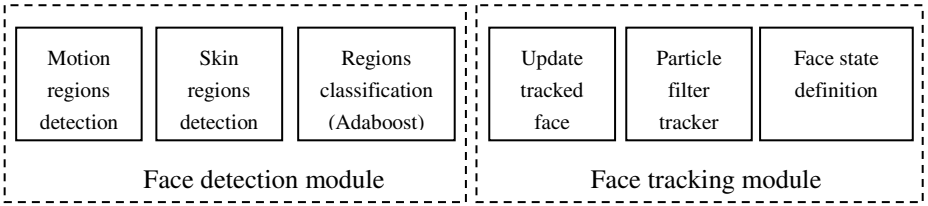


Fig. 1. The proposed detect-track approach

2.1 Face Detection Module

Regarding face detection module, we have proposed in [18] a hybrid approach which benefits from the affluent combination of spatial (skin) and temporal (motion) information in video to improve face detection. In fact, our approach starts by detecting regions in motion using approximate median background subtraction technique. Then, we go over these moving regions to find out pixels having skin color. Skin color detection task was performed using our proposed skin color model HSCr which was developed using a data-mining process. This restriction of regions of interest, by applying motion and skin color filtering, helps to reduce regions of interest and cuts out earlier false positive which can appear jointly with decor or non-skin regions. Finally, we classify these moving-skin color regions into face and non-face using Adaboost algorithm. In [18] we have proved the effectiveness of integrating spatial and temporal information as it helps to avoid tracking false positive faces.

2.2 Face Tracking Module

This module is composed of three main stages: update tracked face stage; particle filter tracker stage and face state definition stage.

Update Tracked Face. The update tracked face stage seeks to update not only the face tracked structure but also the number of tracked faces either there is a new detection or not.

To start this stage, we have to record a first detection shot in order to initialize the tracker. Henceforward, we just applied the update tracked face stage which aims to keep the tracker informed with the detection module results.

Initialization step consists of assigning to the tracking face structure the new measures out came from the detected face structure. Notice here that a face structure (detected or tracked) is defined by its center coordinates (x, y), height, width, distance between the face and the camera and its current state (Leaving, Entering, Stopped, Occluded, Tracked).

In the case when the face detection module fails to detect faces in the scene, we update the tracked face structure and the number of tracked faces in the scene with the previous ones.

Otherwise, when we record a new detection, we have to identify if this new detection belong to an old tracked face or it is for a new entering face. Therefore, we go over each detected face and check if its center belongs to an old tracked face, if it fits this condition we just update the old tracked face structure with the new detected face structure; If not, we verify if there is an old occluded tracked face in the scene. If this is the case, we have to understand that this occluded face has just reappears again and then we update the occluded tracked face structure with the detected face structure without increasing the number of tracked faces.

Otherwise, if we do not find an old face tracked which fits the new detected face and there is no previous occluded face in the scene, we conclude that there is a new face which has just entered to the scene. Thus we have to update the tracked face list by adding the new detected face structure and incrementing the number of tracked faces in the scene.

Particle-Filter Tracker. At time t , each face is described by a state vector S_t . The goal of face tracking is to estimate the face state S_t at time t using a set of observations Z_t . In other words, it consists of constructing the posterior probability density function (pdf) defined as $p(S_t | Z_t)$.

To derive the prior pdf of the current state $p(S_t | Z_{t-1})$, the recursive Bayesian filtering provides the theoretically optimal solution which makes use of the dynamic equation and the already computed pdf $p(S_{t-1} | Z_{t-1})$ of the face state at time $t-1$ in the prediction step. Afterward, in the update step, it employs the face likelihood function $p(Z_t | S_t)$ of the current observation to compute the posterior pdf $p(S_t | Z_t)$. Formally, it can be written as follows:

$$p(S_t | Z_{1:t}) = \frac{p(Z_t | S_t) \int p(S_t | S_{t-1}) p(S_{t-1} | Z_{1:t-1}) dS_{t-1}}{\int p(Z_t | S_t) \int p(S_t | S_{t-1}) p(S_{t-1} | Z_{1:t-1}) dS_{t-1} dS_t} \quad (1)$$

As it showed in equation above, the computation of the posterior probability $p(S_t|Z_{1:t})$ using Bayesian filtering requires a high dimensional integrals computation and can deal with the non-linearity and non-normality of the motion model. High-dimensional integrations cannot be easily computed in a closed analytical form; hence a particle filter tracker was adopted to estimate the posterior probability.

Actually, it is widely accepted that the particle filter is superior to the traditional Kalman filter in terms of tracking performance [12], since the particle filter provides a robust object tracking without being restricted to a linear system model and proves performance to deal with limitations arising from non-linearity and non-normality of the motion model under many tracking scenarios [13, 14].

To perform tracking we have used particle filter technique with adaptive resampling and adjusted it for multiple faces tracking context.

In fact, the particle filter tracker maintains a probability distribution over the tracked face state (location, scale, etc.). Particle filters represent this distribution as a set of weighted particles samples. Each particle is an instantiation illustrating one potential location of the tracked face. Particles having high value of weight reveal the locations where the tracked face is more likely to be. This weighted distribution is propagated through time using the Bayesian filtering equations, the trajectory of the tracked face can be found by using the highest weighted particles. In our case, we have defined the face state as follows: $S_t = \{x, y, s, w, h, \text{histo}\}$, where (x, y) are the face coordinates center, s is the face scale, w and h are respectively the width and the height of the current face, histo is an histogram model of the current region.

The first step in particle filter tracker is initializing distribution which consists of creating N particles on each detected face. Then for each particle, we sample a new face state using second-order autoregressive dynamical model which is a type of random process often used to model and predict an output of a system based on the two previous outputs. Particle weight assignment was performed using Bhattacharyya distance between the predicted face region and the measured one. In fact, particles which have predicted correctly the face location will have high value of weight and conversely the particles which have fail to predict properly the face will have weak value of weight.

Actually, the algorithm consists in making evolve the particle set $P_t = (S_t^n, W_t^n)_{n=1, \dots, N}$, where S_t is hypothetical state of the tracked face and W_t its weight. However, after some iterations, most of particles will have a weak weight close to zero; this is known as the degeneracy problem. The goal of resampling is to focus on high weighted particles and get rid of weak weighted particles by replacing the N particles according to their importance weights.

Unlike the existing works [17, 11] which use particle filter technique for tracking a single object, we have adopted this technique for tracking multiple faces by activating a different set of particles for every new detected face.

Face State Definition. The state definition stage is a significant task since it contributes to update successfully faces, as well as understanding their behavior in the

scene. Hence, we have differentiated five states: Tracked, Occluded, Stopped, Entered or Left.

- *Stopped Face State.*

A face is seen as stopped face if the Euclidean distance between its center in the current frame and the previous one does not exceed two pixels. Furthermore, the current state of the face should not be occluded since an occluded face has often a small amount of movements. Moreover, the distance separating the face from the camera should not be greater than one centimeter among the previous and the current frame (Figure 2).



Fig. 2. Stopped Face State

- Leaving and Entering Face State.

To identify leaving and entering face, we have defined a window of track as an area where the face could appear clearly in the scene. If the tracked face center is out of this window of track, we declare that it is either entering or leaving the scene. This window of track is defined as follows:

$$\begin{cases} X_{TrackWindow} = X_{Frame} - 10 \\ Y_{TrackWindow} = Y_{Frame} - 10 \\ Height_{TrackWindow} = Height_{Frame} - 20 \\ Width_{TrackWindow} = Width_{Frame} - 20 \end{cases} \quad (2)$$

A face is perceived as a leaving face if its center goes beyond the window of track and the distance between the center of current tracked face and the frame center is larger than the distance between the center of previous tracked face and the frame center (i.e. the face is going far from the frame center)

Once we have identified a leaving state we have to update face tracked structure, thus we delete the left face structure from face tracked list and decrease the number of tracked faces in the scene (Figure 3).

A face is defined as an entered face, if its center is out of the window of track and the distance between the center of current tracked face and the frame center is lower than the distance between the center of previous tracked face and the frame center (i.e. the face is moving toward the frame center).



Fig. 3. Leaving Face State

- Ocluded Face State.

A face is classified as an occluded face in two cases: when there is a faces interaction constraint or there is a one-face occlusion. In fact, when the distance between the centers of two tracked faces is lower than the sum of their widths divided by two, we declared that we have a faces interaction case. However, when the number of skin color pixel in the tracked region is lower than 30% of the total face tracked region we asserted that we have a one-face occlusion case (Figure 4).

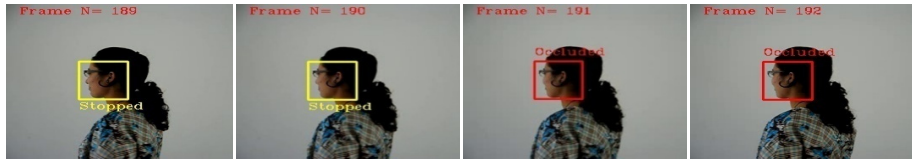


Fig. 4. One-Face Occlusion Case (Total Head Rotation)

In the first instance, when faces interaction occurs, we have to identify which face occludes the other. Therefore, we compare the distance between each interacted face and the camera. The face having a larger distance than the interacted one will be classified as an occluded face. With regard to the face hiding the occluded one, it will be classified as a tracked face (Figure 5).

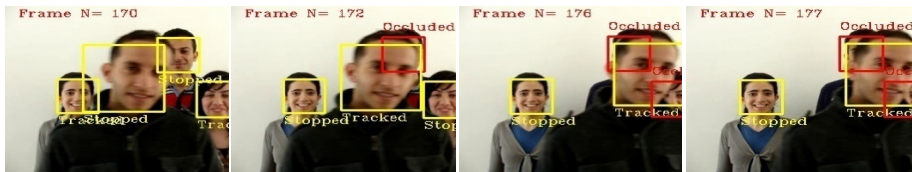


Fig. 5. Faces Interaction Case

- Tracked Face

The tracked face state is the default state. In other words, when a face is not occluded, or stopped, nor entering or leaving, it is obviously moving in the scene and hence it is perceived as a tracked face.

3 Experimental Results

In order to evaluate the performance of our proposed detect-track system, we have carried out two series of experiments. The first one dealt with the evaluation of the proposed detect-track system and the second one seeks to evaluate the performance of our adapted particle filter tracking technique.

3.1 First Serie of Experiments

This set of experiments aims to evaluate the performance of our proposed approach for face detection and tracking in video, as well as traces its behavior when only face detection module is applied.

Experimental study was performed on two video databases sequences, La Casia database [15] and our own recorded database. In fact, we have picked 10 sequences from La Casia database (5 sequences under uniform light and 5 sequences under varying light), also we have captured 15 sequences which deal with various challenging problems that can occur while detecting or tracking faces. Both video databases sequences description are summarized in table 1.

Table 1. Video databases sequences description

Video Databases	Constraints		# Sequences	# Frames	#Faces	Duration (seconds)
La Casia Database	Uniform Light		5	985	985	30
	Varying Light		5	985	985	30
Our Database	One Face	Free Head Rotation	2	266	266	8
		In/Out Face	2	194	141	6
		Partial/Total Occlusion	2	257	257	8
		Total Head Rotation	2	557	321	18
	Multiple Faces	Faces Interaction	3	989	2765	32
		In/ Out Faces	2	509	720	16
		Free Heads Rotation	2	347	694	11
TOTAL			25	5089	7134	129

Table 2 illustrates the obtained results while applying only face detection module and the recorded results by performing our face tracking module jointly with face detection module. To ensure evaluation, the familiar rates of precision and recall were used.

The table 2 shows that with tracking module, both of recall and precision rates were improved. In fact we record height rate of precision rate while applying only detection module or running the whole detect-track system. This was expected, since we process with a filtering stage while detecting faces using spatial (skin) and temporal (motion) information. This filtering stage helps to get rid of false positive

detection in an earlier step. Moreover, we notice that even when there is a total head rotation, we succeed to keep following it, so we jump from 61.9% of recall rate, to 100% thanks to our combination of detection and tracking modules

Table 2. Precision and recall rates of the detection module alone and the whole detect-track system

Video Databases	Constraints		Recall Rate		Precision Rate	
			Only Detection	Detect-Track	Only Detection	Detect-Track
La Casia Database	Uniform Light		82.5 %	100 %	100 %	100 %
	Varying Light		100 %	100 %	100 %	100 %
Our Data-base	One Face	Free Head Rotation	83.3 %	100 %	100 %	100 %
		In/Out Face	100 %	100 %	100 %	100 %
		Partial/Total Occlusion	73.6 %	100 %	100 %	100%
		Total Head Rotation	61.9 %	100 %	100 %	86.0 %
	Multiple Faces	Faces Interaction	93.4 %	100 %	100 %	97.8 %
		Free Heads Rotation	69.5 %	99.3 %	100 %	92.2 %
		In/Out Faces	100 %	100 %	100 %	100 %

Nevertheless, we perceive that the precision rate decrease little bit while tracking an occluded face. In fact, this precision decrease is due to the fast movement of the occluded face, as the face is not only occluded but also moves quickly. Although, this has affected slightly the precision rate but it does not touch the general performance of our detect-track system since a new detection will update naturally the face tracker.

3.2 Second Serie of Experiments

With regard to tracking technique evaluation, we have compared our adapted particle filter face tracker technique with the referenced tracking technique CamShift [16]. To insure such evaluation, we suggest a new distance measure to appraise the performance of each face tracker technique. This distance measure is defined as the difference between X and Y ground truths trajectories and the X and Y tracked trajectories averaged over all frames in each video sequence. Seeing that this distance is an average measure, we have rounded it to the nearest whole unit. In addition, since we have dealt with large databases we cannot display all the computed results, so only one example of each constraint will be exposed. Table 3 sums up this comparative study.

Table 3 depicts the obtained results with our adapted particle filter and CamShift techniques. In fact, these results reveal the sensibility of CamShift tracker when some constraints arise. In particular, when the face is occluded the Camshift tracker fails to follow it and drift away with 22 pixels from the Y ground truth trajectory however the adapted particle filter keeps pursuing correctly this occluded face (only 3 pixels away from the Y ground truth trajectory).

Table 3. Performance comparison of Particle filter and CamShift trackers Techniques

Video Databases	Constraints		Adapted Particle Filter		CamShift		
			Dist_X	Dist_Y	Dist_X	Dist_Y	
La Casia Database	Uniform Light (Seq: jam1)		3	2	4	4	
	Varying Light (Seq: jal5)		5	3	39	6	
Our Database	One Face	Free Head Rotation (Sequence number 2)	3	2	10	5	
		In/Out Face (Sequence number 2)	6	1	3	7	
		Partial/Total Occlusion (Sequence number 2)	3	3	3	22	
		Total Head Rotation (Sequence number 2)	8	1	23	17	
	Multiple Faces	Faces Interaction (Sequence number 2)	Face 1	4	2	9	6
			Face 2	5	2	17	10
			Face 3	3	2	12	14
			Face 4	7	1	15	11
		In/Out Faces (Sequence number 1)	Face 1	3	1	3	4
			Face 2	2	2	3	2
		Free Heads Rotation (Sequence number 2)	Face 1	4	3	9	6
			Face 2	5	4	10	5

Furthermore, our proposed particle filter seems to be more faithful to the ground truth trajectory than CamShift tracker in varying illumination constraint since CamShift tracker discards with 39 pixels from the X ground truth trajectory while the particle filter tracker keep being close (5 pixels) to the X ground truth trajectory. Moreover, regarding faces interaction constraint which is a complex challenge since faces are moving one next the other in such way that they hide each one the other, our adapted particle filter tracker proves more performance than CamShift tracker seeing that particle filter trajectory along X and Y is more closer to the ground truth trajectory than CamShift trajectory. Based on this report, we can conclude that CamShift tracker usually drifts away and in some times loses tracking when tracking error propagates through video frames. This problem is especially relevant when partial or total face occlusion occur or nearby faces appear with similar distributions to the target face.

Figure 6 displays the X-trajectory of the second video sequence under faces interaction constraint. Through this figure, we can see that the second and the third faces enter successively to the scene at the frame number 41 and 102 respectively; then we can notice how the fourth face enters at the frame number 165 and pass in front of the three faces and occludes them; however our face detect-tracker system succeed to keep following them although they are really occluded.

This experimental study has highlighted the robustness and the effectiveness of our proposed approach for face detection and tracking in video. In fact, the combination of the detection and tracking modules supports automatic tracking with no need to manual initialization or re-initialization and reaches best performance to deal different challenging problems which can arise while face detection or tracking stages. This is achieved without affecting the quality of tracking or its computational complexity.

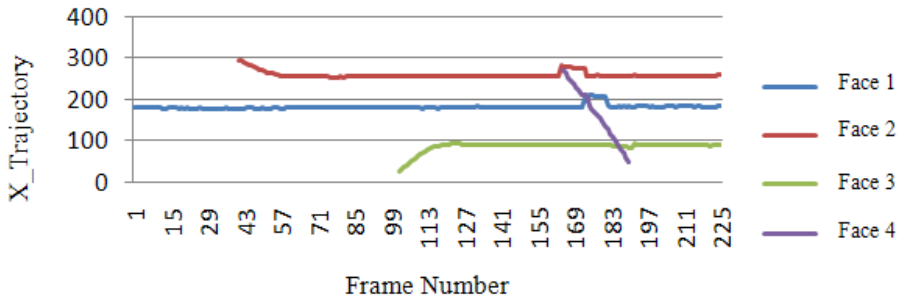


Fig. 6. X-Trajectories of faces in interaction

4 Conclusion

A face detect-track system was proposed in this paper. The face detection module makes use of motion information as well as skin color to earlier get rid of false positive. Based on detection modules results we update the tracking module. Then we adapt the particle filter technique for multiple faces track scenario without affecting its performance. Finally we define a specific state for each tracked face. This state definition step helps to understand each face behavior in the scene; hence we have defined five face states: Tracked, Occluded, Stopped, Entered or Left.

The experimental results have proved the effectiveness and the performance of our proposed combination of face detection and tracking modules to deal with the various face detection and tracking problems. In fact, the detection module helps to handle tracking problems and conversely, the tracking module provide an intelligent way to handle detection challenges.

In our future works, we seek to estimate continually the pose of the tracked faces. Such works will help to capture the best frame having the most reliable pose for a further face recognition task.

References

1. Kuchinsky, A., Pering, C., Creech, M.L., Freeze, D., Serra, B., Gwizdka, J.: Fotofile: Consumer multimedia organization and retrieval system. In: Proc. ACM SIG CHI 1999 Conf. (1999)
2. Li, X., Kwan, C., Mei, G., Li, B.: A Generic Approach to Object Matching and Tracking. In: Campilho, A., Kamel, M.S. (eds.) ICIAR 2006. LNCS, vol. 4141, pp. 839–849. Springer, Heidelberg (2006)
3. Adipranata, R., Ballangan, C.G., Rostianingsih, S., Ongkodjodjo, R.P.: Real-Time Human Face Tracker Using Facial Feature Extraction. In: International Conference on Soft Computing (2007)
4. Romero, M., Bobick, A.: Tracking Head Yaw by Interpolation of Template Responses. In: CVPRW 2004 Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (2004)

5. Comaniciu, D., Ramesh, V., Andmeer, P.: Kernel-based object tracking. *IEEE Trans. Patt. Analy.*, 564–575 (2003)
6. Swaminathan, G., Venkoparao, V., Bedros, S.: Multiple appearance models for face tracking in surveillance videos. In: *Proceedings of AVSS*, pp. 383–387 (2007)
7. Hou, Y., Sahli, H., Ilse, R., Zhang, Y., Zhao, R.-c.: Robust Shape-Based Head Tracking. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) *ACIVS 2007. LNCS*, vol. 4678, pp. 340–351. Springer, Heidelberg (2007)
8. Kohsia, S.H., Mohan, M.T.: Robust Real-Time Detection, Tracking, and Pose Estimation of Faces in Video Streams. In: *The 17th International Conference on Pattern Recognition* (2004)
9. Sobottka, K., Pitas, I.: A novel method for automatic face segmentation, facial feature extraction and tracking. *Signal Processing: Image Communication*, 263–281 (1998)
10. Gunn, S.R., Nixon, M.S.: Snake Head Boundary Extraction using Global and Local Energy Minimization. In: *Proceedings of the 13th ICPR*, pp. 581–585 (1996)
11. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002, Part I. LNCS*, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
12. Chang, C., Ansari, R., Khokhar, A.: Multiple object tracking with kernel particle filter. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (2005)
13. Li, P., Zhang, T., Pece, A.E.C.: Visual contour tracking based on particle filters. *Image Vis. Comput.* 21, 111–123 (2003)
14. Okuma, K., Taleghani, A., de Freitas, N., Little, J.J., Lowe, D.G.: A Boosted Particle Filter: Multitarget Detection and Tracking. In: Pajdla, T., Matas, J.(G.) (eds.) *ECCV 2004. LNCS*, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
15. La-Cascia, M., Sclaro, S., Athitsos, V.: Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models. *IEEE Transactions on Patterns Analysis and Machine Intelligence* (2000)
16. Bradski, G.R.: Computer Vision Face Tracking For Use in a Perceptual User Interface. In: *Proc. IEEE Workshop on Applications of Computer Vision*, pp. 214–219 (1998)
17. Andrew, B., Zulfiqar, H.K., Irene, Y.H.G.: Robust Object Tracking Using Particle Filters and Multi-region Mean Shift. *IEEE Transactions on Circuits and Systems for Video Technology*, 74–87 (2011)
18. Mliki, H., Hammami, M., Ben-Abdallah, H.: Real time face detection based on motion and skin color information. Appeared in the 6th International Conference on Multimedia and Ubiquitous Engineering (2012)

Using a Differential Pressure Sensor as Spirometer

Martin Augustynek, Ondrej Adamec, and David Micanik

VSB– Technical University of Ostrava, FEECS, Department of Cybernetics and Biomedical Engineering, Ostrava, Czech Republic, 17. Listopadu 15, Ostrava – Poruba, 70833
{martin.augustynek, ondrej.adamec}@vsb.cz

Abstract. For a doctor to determine the most accurate diagnosis of diseases of the respiratory tract, it must be as accurate as possible insight into the problem. Imaging technology allows to look into the body, unfortunately for example lung is an organ, where without contrast agent does not buy the picture. Furthermore, the methods that can be used are whole body plethysmography or, a better option, spirometry. A measurement of spirometry is performed by the pneumotachograph or the spirometry. Spirometer measures lung volumes and lung capacity. Pneumotachograph is the flow rate measuring device, but can also be used for indirect measurement of lung volumes and capacities. Spirogram is the result of spirometry measurements.

Keywords: Spirometry, lungs, function lungs parameters, digital communication, Matlab.

1 Introduction

In the past it was possible to perform spirometer only in medical facilities. Older spirometers are voluminous, have high maintenance and their measurement capabilities are very limited. After each patient had to be the instrument properly disinfected to prevent transmission of bacteria. Over time, the dimensions of the device and thus diminished their demands for service and maintenance. The coup came with the advent of pneumotachograph. They also want to specify multiple parameters of lung and especially sharply reduced the size of the device itself. Now, using simple instruments, measurements can be made in domestic environments. Measurement results are displayed on the LCD display device or on a computer screen [8].

This paper describes how to create a demonstration and simple devices for spirometer. It can find usage in a professional environment or in an education area in an university. Modern digital sensor was used with excellent high sensitivity, low power consumption and preprocessing options. It was also used Matlab programming options.

2 Methods

For spirometry is possible to use several different methods like water spirometer, dry-bellowed wedge spirometer or fully electronics spirometer, typically. Electronic

spirometers have been developed that compute airflow rates in a channel without the need for fine meshes or moving parts. They operate by measuring the speed of the airflow with techniques such as ultrasonic transducers, or by measuring pressure difference in the tube [5]. These spirometers have greater accuracy by eliminating the momentum and resistance errors associated with moving parts such as windmills or flow valves for flow measurement. They also allow improved hygiene between patients by allowing fully disposable air flow channels.

3 Measuring Set

Principle of measuring a spirometry was chosen differential manometer. Spirometer body is made up of a biological filter HEPA Light. Electronics device includes a digital component to communicate with a computer. This solution enables the processing of measured data using specialized software without having to perform other calculations.

The sensor is a product of Swiss company Sensirion (see Tab. 1). The sensor output is digital, the I2C bus. Temperature calibration is automatically done by the sensor and electronics output is already linearized. Influence of initial offset aging of the material is given less than 0.1 Pa per year. The default sensitivity resolution of 12 bits (can be 9 to 16 bits) is given $0,2 Pa$. The sensor is recommended, inter alia, in health care. This sensor offers the best fit of desired parameters and especially its digital output, thus allowing easier processing in the computer.

The measurement is started by breathing into the spirometer. Non-electric value is converted to digital signals by sensor. [6]

Table 1. Sensor parameters

Power supply	3 - 3,6 V for < 6 mA
Measurement range	-500 Pa ÷ 500 Pa
Response time	4.6 ms
Temp. calibration range	0 °C – 50 °C
I2C working frequency	100 kHz (max. 400 kHz)

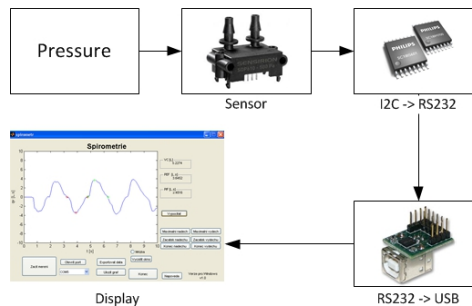


Fig. 1. Measuring set

The digital signal is converted into comprehensible forms for serial communication. The computer information is drawn from the serial line and the measurement is complete interpretation of the measured values. Result can be interpreted as a graph or directly calculated values of lung parameters. The measurement block diagram is displayed in Fig 1.

For easier interpretation of the results, the mathematical program Matlab is used. Matlab allows contact the programming elements and mathematical operations. Facilitates the work by omitting the definition of computing and thus automate the processing of values measured. Ideal measurement software should performs satisfy several following requirements:

- easy operation
- error – free functionality (clean code)
- the possibility to save results

The software (see Fig. 2) is developed as multiplatform for Windows and Mac OS X. For a smooth running program, you must have installed the virtual port driver and the Matlab version at least 2008 for Windows or for Mac OS X, Matlab 2009. For both operating systems, 32-bit and 64-bit architectures are supported.

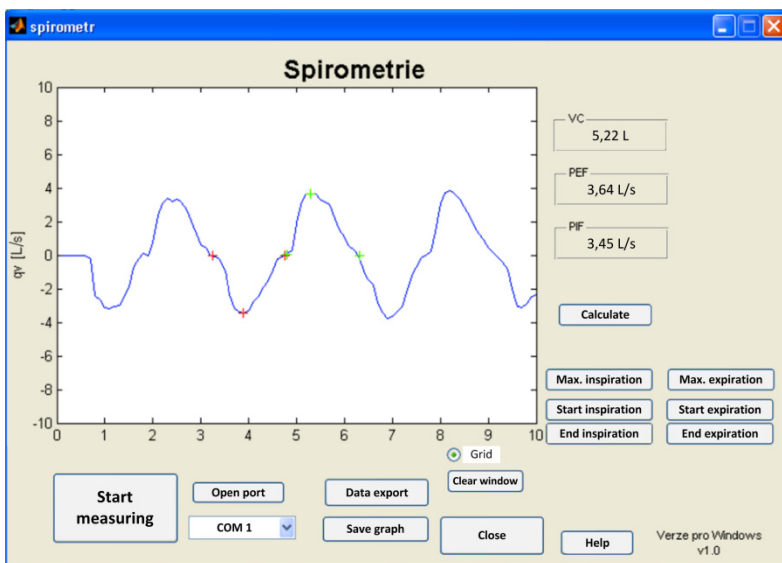


Fig. 2. Software window

4 Calibration

The value of the pressure difference Δp is linearly dependent on the volume q_v and size of air flow resistance R (see (1)).

$$\Delta p = p_2 - p_1 = R \cdot q_v \quad (1)$$

To calculate the correct flow of gas was needed to determine the value of air resistance. The gas flow is expressed first by equation (1) and from already adjusted equation (2) is calculated the gas flow value q_v .

$$q_v = \frac{\Delta p}{R} \quad (2)$$

The value of pressure difference and flow rate were known during the calibration. Therefore equation (2) was modified into (3).

$$R = \frac{\Delta p}{q_v} \quad (3)$$

The gas rate was simulated using a phantom of an air, in the range of $0,3 L \cdot s^{-1}$ to $0,85 L \cdot s^{-1}$. Phantom of air was connected to the spirometer input and the differential pressure was sensed by a computer with a period of 10 s. Readout pressure was calculated by averaging the readings over one minute.

Next table (see Tab. 2) shows results from calibration. The average air resistance R was calculated as the mean of the measured resistance and is equal to $0,2029 Pa \cdot s \cdot L^{-1}$. This value was set as the resistance to air flow. Then it is easy to compute unknown air flow by (2) because Δp is measured and R is known from previous computation.

Table 2. Computed resistance R

$q_v [L \cdot s^{-1}]$	$\Delta p [Pa]$	$R [Pa \cdot s \cdot L^{-1}]$
0,30	0,0325	0,14
0,35	0,0490	0,1083
0,40	0,0653	0,1632
0,45	0,0817	0,1816
0,50	0,0980	0,1960
0,55	0,1143	0,2078
0,60	0,1307	0,2178
0,65	0,1470	0,2262
0,70	0,2251	0,3216

5 Testing

Test was made for all functions of the device and measurement software. Readings from the program were compared with values measured on the spirometer made by ZAN. The spirometer ZAN100 (the principle of differential pressure) was used for

testing. There have been several controlled test measurements. The next table (see Table 3) shows results from testing.

The difference of the reference level and minimum inspiration values is equal to the peak of inspiration flow (PIF) in liters per second. By integrating the volumetric flow across the interval of maximum inspiration is received a value of vital lung capacity in liters. See (4) where $t_{1,2}$ is start and end of inspiration, q_v is volume flow and V is vital lung capacity.[9][10]

$$V = \int_{t_1}^{t_2} q_v dt \tag{4}$$

Table 3. Compare between SDP610 and ZAN100

Parameters	SDP610	ZAN100
PEF	3,6 $L \cdot s^{-1}$	3,8 $L \cdot s^{-1}$
PIF	3,4 $L \cdot s^{-1}$	3,6 $L \cdot s^{-1}$
VC	5,2 L	5,5 L

Next figure (see Fig. 3.) shows the record along breath made from one inspiration and expiration, and an intensive inspiration and expiration.

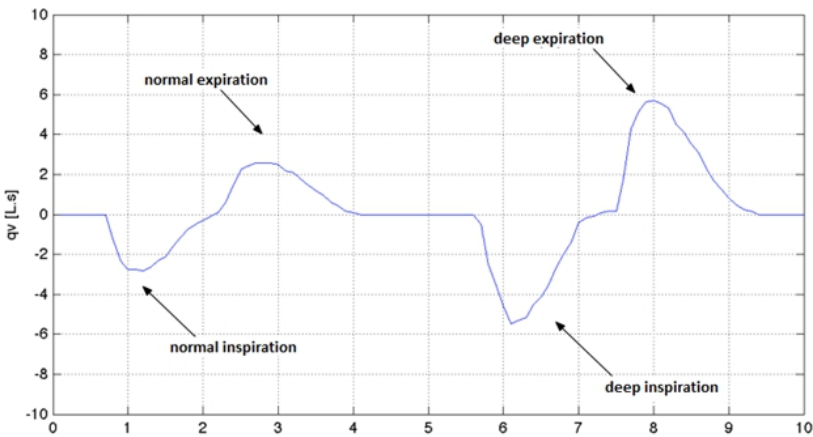


Fig. 3. Air flow along breath and intensive breath

6
Conclusions

Method of measurement using differential manometer was chosen. Durability and reliability of the spirometer are improving by no moving parts are needed during the measurement. For compiling of the devices were used materials from health care, or

materials wholesome. HEPA filter Light is used as the air resistance to the spirometer. It also serves as a barrier to the spread of bacteria. Spirometer is designed for easy using with the hygienic mouthpiece.

Electronic part of the device was selected for processing of signal on the computer. As differential sensor, the digital sensor SDP610 from the company Sensirion has been selected. With the compensation capabilities of this sensor is not necessary to further modify the signal. The signals from the sensor are transferred via serial communication line and then sent to a computer. In the computer was created software for processing.

Sensor SDP610 proved to be very suitable sensor with potential for use in medical applications.

Acknowledgements. The work and the contribution were supported by the project Student grant agency “Biomedical engineering systems VIII” and TACR TA01010632 “SCADA system for control and measurement of process in real time”.

Has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic.

References

1. Cerny, M., Penhaker, M.: Wireless Body Sensor Network in Health Maintenance system. Journal Electronics and Electrical Engineering IX(9), 113–116 (2011), doi:10.5755/j01.eee.115.9.762, ISSN 1392-1215
2. Spišák, J., Imramovský, M., Penhaker, M.: Senzory a snímače v biomedicíně, 117 p. VŠB-Technická univerzita Ostrava, Ostrava (2007) ISBN 978-80-248-1607-4
3. Augustynek, M., Penhaker, M., Vybíral, D.: Devices for position detection. Journal of Vibroengineering 13(3), 531–523, ISSN: 1392-8716
4. SDP610 [online]. Sensirion: The Sensor Company (2009), http://www.sensirion.com/en/pdf/product_information/Datasheet_SDP600series_differential_pressure_sensor.pdf (cit. November 03, 2010)
5. Penhaker, M., et al.: Lékařské diagnostické přístroje: Učební texty. 1. vydání, 332 p. VŠB - Technická univerzita Ostrava, Ostrava (2004) ISBN 80-248-0751-3
6. SirXpert [online]. Become an expert in spirometry, <http://spirxpert.com/> (cit. April 22, 2011)
7. Spirometrie.info [online]. World of spirometry, <http://www.spirometrie.info> (cit. April 20, 2011)

Hybrid Negative Selection Approach for Anomaly Detection

Andrzej Chmielewski¹ and Sławomir T. Wierchoń^{2,3}

¹ Faculty of Computer Science, Białystok University of Technology,
ul. Wiejska 45a, 15-331 Białystok, Poland

a.chmielewski@pb.edu.pl

² Institute of Informatics, Gdańsk University,
ul. Wita Stwosza 57, 80-952 Gdańsk, Poland

³ Institute of Computer Science, Polish Academy of Sciences,
ul. Jana Kazimierza 5, 01-248 Warszawa, Poland

stw@ipipan.waw.pl

Abstract. This paper describes a *b-v* model which is enhanced version of the negative selection algorithm (*NSA*). In contrast to formerly developed approaches, binary and real-valued detectors are simultaneously used. The reason behind developing this hybrid is our willingness to overcome the scalability problems occurring when only one type of detectors is used. High-dimensional datasets are a great challenge for *NSA*. But the quality of generated detectors, duration of learning stage as well as duration of classification stage need a careful treatment also. Thus, we discuss various versions of the *b-v* model developed to increase its efficiency. Versatility of proposed approach was intensively tested by using popular testbeds concerning domains like computer's security (intruders and spam detection) and recognition of handwritten words.

Keywords: Artificial immune system, anomaly detection, multi-dimensional data.

1 Introduction

Natural immune system (NIS) prevents living organism against intruders called *pathogens*. It consists of a number of cells, tissues, and organs that work together to protect the body. The main agents responsible for the adaptive and learning capabilities of the NIS are white blood cells called *lymphocytes*. These differentiate into two primary types: B- and T-lymphocytes called also B- and T-cells for brevity. T-lymphocytes are like the body's military intelligence system, seeking out their targets and sending defenses to lock onto them. Next, B-lymphocytes, destroys detected invaders to protected the body. It is only a very short description of NIS; an unacquainted reader is referred e.g. to [3] for further details.

The mechanisms and procedures developed within NIS were an inspiration for *Artificial Immune Systems* (AIS). Negative selection, clonal selection, idiotypic networks are prominent examples of such mechanisms oriented towards

fast and efficient discrimination between own cells (called *self*) and pathogens (called *nonself*). Only fast and effective response on intruders activity can protect organisms against damaging or even die. It is worth to emphasize that in Nature the total number of various types of pathogens is far greater than 10^{16} , whereas there are about 10^6 own cells types only. This discrepancy between the two magnitudes illustrates unusual efficiency of detection mechanisms developed by the NIS. It is expected that employing such ideas in computer algorithms will result in their efficiency in such domains like novelty detection, falsification detection, diagnosis systems, computer security (intruders and spam detection) and many others, where binary classification is sufficient and is required to process an huge amount of data. Exhaustive review of current state in domain of AIS was presented in [16].

In this paper, we focus on negative selection. This mechanism is employed to generate a set of detectors and help protect the body against self-reactive lymphocytes. The lymphocytes start out in the bone marrow and either stay there and mature into B-cells (this process is called *affinity maturation*), or they leave for the thymus gland, where they mature into T-cells in the process of *negative selection*. This process has inspired Forrest *et al.* [7] to formulate so-called *negative selection algorithm* (*NSA*). First, detectors (a counterpart of T-lymphocytes) are generated (usually, in random way). Next, freshly generated detector is added to the set of valid detectors only if does not recognize any *self* element. A nice feature of the *NSA* is that it does not need examples of *nonself* samples (counterpart of *pathogens*) to detect them.

A key problem when applying *NSA* in real-life application seems to be its scalability. For example, to detect spam or intruders at computer networks *NSA* should be able operate on high-dimensional data. Moreover, in such domains the classification process have to be performed online, without significant delays, what makes this task much more difficult to solve. Up to now, neither binary (called *b*-detectors) nor real-valued detectors (called *v*-detectors as they are generated by the *V-Detector* algorithm mentioned in subsection 2.2) were not capable to detect anomalies in satisfactory degree.

To overcome this problem, we propose to use both the types of detectors simultaneously. This hybrid, called *b-v* model, as showed performed experiments, provides much better results in comparing to single detection models as well as in comparing to traditional, statistical approaches, even though only positive *self* examples are required at learning stage. It makes this approach interesting alternative for well known classification algorithms, like SVM, *k*-nearest neighbors, etc.

2 Negative Selection Algorithm

The *NSA*, i.e. the negative selection algorithm, proposed by Forrest *et al.*, [7], is inspired by the process of thymocytes (i.e. young T-lymphocytes) maturation: only those lymphocytes survive which do not recognize any *self* molecules.

Formally, let \mathcal{U} be a universe, i.e. the set of all possible molecules. The subset \mathcal{S} of \mathcal{U} represents collection of all *self* molecules and its complement \mathcal{N} in \mathcal{U}

represents all *nonself* molecules. Let $\mathfrak{D} \subset \mathcal{U}$ stands for a set of detectors and let $match(d, u)$ be a function (or a procedure) specifying if a detector $d \in \mathfrak{D}$ recognizes the molecule $u \in \mathcal{U}$. Usually, $match(d, u)$ is modeled by a distance metric or a similarity measure, i.e. we say that $match(d, u) = \text{true}$ only if $dist(d, u) \leq \delta$, where $dist$ is a distance and δ is a pre-specified threshold. Various matching function are discussed in [8], [11].

The problem relies upon construction the set \mathfrak{D} in such a way that

$$match(d, u) = \begin{cases} \text{false} & \text{if } u \in \mathcal{S} \\ \text{true} & \text{if } u \in \mathcal{N} \end{cases} \quad (1)$$

for any detector $d \in \mathfrak{D}$.

A naive solution to this problem, implied by biological mechanism of negative selection, consists of five steps:

- (a) Initialize \mathfrak{D} as empty set, $\mathfrak{D} = \emptyset$.
- (b) Generate randomly a detector d .
- (c) If $match(d, s) = \text{false}$ for all $s \in \mathcal{S}$, add d to the set \mathfrak{D} .
- (d) Repeat steps (b) and (c) until sufficient number of detectors will be generated.

Below the binary and real-valued representations of the problem are described.

2.1 Binary Representation

This type of representation was applied by Forrest *et al.* [6] to capture anomalous sequences of system calls in UNIX systems and next to model the system for monitoring TCP SYN packets to detect network traffic anomalies (called LISYS) [9].

In case of binary encoding, the universe \mathcal{U} becomes l -dimensional Hamming space, $\mathbb{H}^l = \{0, 1\}^l$, consisting of all binary strings of fixed length l :

$$\mathbb{H}^l = \{\underbrace{000\dots000}_l, \underbrace{000\dots001}_l, \dots, \underbrace{111\dots111}_l\}$$

Hence the size of this space is 2^l . The most popular matching rules used in this case are:

- (a) r -contiguous bit rule [6], or
- (b) r -chunks [2].

Both the rules say that a detector bonds a sample (i.e. data) only when both the strings contain the same substring of length r . To detect a sample in case (a), a window of length r ($1 \leq r \leq l$) is shifted through censored samples of length l . In case (b) the detector $t_{i,\mathbf{s}}$ is specified by a substring \mathbf{s} of length r and its

position i in the string. Below an example of matching a sample by r -detector (left) and r -chunk for affinity threshold $r = 3$ is given

$$\begin{array}{ccc}
 \overbrace{1\ 0\ 0\ 0\ 1\ 1\ 1\ 0}^l & \leftarrow \text{sample} \rightarrow & \overbrace{1\ 0\ 0\ 0\ 1\ 1\ 1\ 0}^l \\
 0\ 1\ \underbrace{0\ 0\ 1}_r\ 0\ 0\ 1 & \leftarrow r\text{-detector; } r\text{-chunk} \rightarrow & **\ \underbrace{0\ 0\ 1}_r\ **
 \end{array}$$

Here it was assumed that irrelevant positions in a string of length l representing the r -chunk $t_{3,001}$ are filled in with the star (*) symbol. This way r -chunk can be identified with schemata used in genetic algorithms: its order equals r and its defining length is $r-1$. Although a single r -detector recognizes much more strings than a single r -chunk, this last type of detector allows more accurate coverage of the \mathcal{N} space [2].

Further, the notion of the ball of recognition allows to define “optimal” repertoire \mathfrak{D} . Namely it consists of the detectors located in \mathbb{H}^l in such a way that they cover the space \mathcal{N} and their balls of recognition overlap minimally. A solution to such stated problem was given in [17]. To construct the r -detectors we split all the *self* strings into the templates represented identically as the r -chunks and we construct the detectors by gluing these r -chunks that do not belong to the set \mathcal{S} . More formally, if $t_{i,\mathcal{S}}$ and $t_{j,\mathcal{W}}$ are two candidate r -chunks, we can glue them if both the substrings are identical on $r-1$ positions starting from position $i+1$.

Using such an optimality criterion we come to the conclusion that shortest detectors are more desirable as they are able to detect more samples. However, Stibor [14] showed the coherence between r and l values for various cardinalities of \mathcal{S} in terms of the probability of generating detectors, P_g . He distinguished three phases:

- Phase 1 (for lower r) – the probability P_g is near to 0,
- Phase 2 (for middle r) – the probability P_g rapidly grows from 0 to 1 (so called *Phase Transition Region*),
- Phase 3 (for higher r) – the probability is very near to 1.

Hence, we should be interested in generating detectors with medium length r (belonging to the second region) and eventually with larger values of r if the coverage of \mathcal{N} is not sufficient. It is worth to emphasize, that the detectors can not be too long, due to exponential increase in the duration of learning process, which should be finished in reasonable time.

2.2 Real-Valued Representation

To overcome scaling problems inherent in Hamming space, Ji and Dasgupta [10] proposed real-valued negative selection algorithm, termed *V-Detector*.

It operates on (normalized) vectors of real-valued attributes; each vector can be viewed as a point in the d -dimensional unit hypercube, $\mathcal{U} = [0, 1]^d$. Each *self* sample, $s_i \in \mathcal{S}$, is represented as a hypersphere $s_i = (c_i, r_s)$, $i = 1, \dots, l$, where

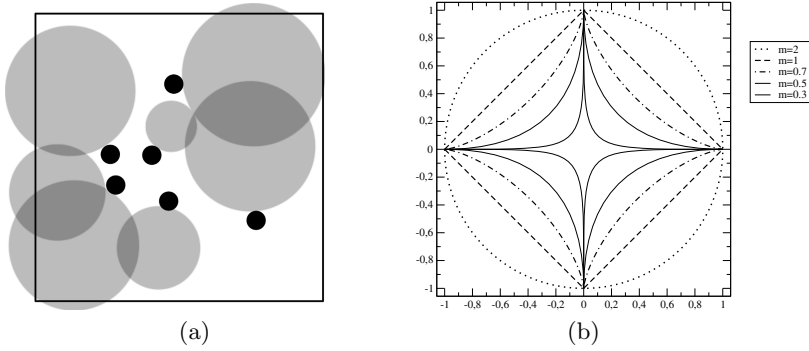


Fig. 1. (a) Example of performance V -Detector algorithm for 2-dimensional problem. Black and grey circles denotes *self* samples and *v*-detectors, respectively. (b) Unit spheres for selected L_m norms in 2D.

l is the number of *self* samples, $c_i \in \mathcal{U}$ is the center of s_i and r_s is its radius. It is assumed that r_s is identical for all s_i 's. Each point $u \in \mathcal{U}$ inside any *self* hypersphere s_i is considered as a *self* element.

The detectors d_j are represented as hyperspheres also: $d_j = (c_j, r_j)$, $j = 1, \dots, p$ where p is the number of detectors. In contrast to *self* elements, the radius r_j is not fixed but it is computed as the Euclidean distance from a randomly chosen center c_j to the nearest *self* element (this distance must be greater than r_s , otherwise detector is not created). Formally, we define r_j as

$$r_j = \min_{1 \leq i \leq l} \text{dist}(c_j, c_i) - r_s \quad (2)$$

The algorithm terminates if predefined number p_{max} of detectors is generated or the space $\mathcal{U} \setminus \mathcal{S}$ is sufficiently well covered by these detectors; the degree of coverage is measured by the parameter co – see [10] for the algorithm and its parameters description.

In its original version, the V -Detector algorithm employs Euclidean distance to measure proximity between a pair of samples. Therefore, *self* samples and the detectors are hyperspheres (see Figure 1(a)). Formally, Euclidean distance is a special case of Minkowski norm L_m , where $m \geq 1$, which is defined as:

$$L_m(x, y) = \left(\sum_{i=1}^d |x_i - y_i|^m \right)^{\frac{1}{m}}, \quad (3)$$

where $x = (x_1, x_2, \dots, x_d)$ and $y = (y_1, y_2, \dots, y_d)$ are points in \mathbb{R}^d .

Particularly, L_2 -norm is Euclidean distance, L_1 -norm is Manhattan distance, and L_∞ is Tchebyshev distance.

However, Aggarwal *et al.* [11] observed that L_m -norm loses its discrimination abilities when the dimension d and the values of m increase. Thus, for example,

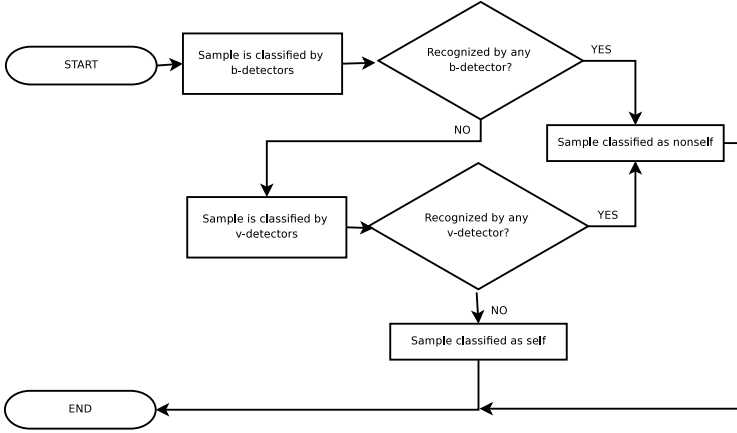


Fig. 2. Flow diagram of the classification process for b - v model

Euclidean distance is the best (among L_m -norms) metrics when $d \leq 5$. For higher dimensions, the metrics with lower m (i.e. Manhattan distance) should be used.

Based on this observation, Aggarwal introduced *fractional distance metrics* with $0 < m < 1$, arguing that such a choice is more appropriate for high-dimensional spaces. Experiments, reported in [4], partially confirmed efficiency of this proposition. For $0.5 < m < 1$, more samples were detected, in comparison to L_1 and L_2 norms. However, for $m < 0.5$ the efficiency rapidly decreased and for $m = 0.2$, none samples were detected. Moreover, these experiments confirmed also a trade-off between efficiency, time complexity and m . For fractional norms, the algorithm runs slower for lower m values; for $L_{0.5}$ the learning phase was even 2-3 times longer than for L_2 .

Another consequence of applying fractional metrics for V -Detector algorithm is modification of the shape of detectors. Figure 1(b) presents the unit spheres for selected L_m -norms in 2D with $m = 2$ (outer most), 1, 0.7, 0.5, 0.3 (inner most).

3 Model b - v

Unsatisfactory coverage of space \mathcal{N} is the main flaw of the v -detectors. To overcome this disadvantage as well as to improve the detection rate (DR for short) and to fasten the classification process, a mixed approach, i.e. b - v model was proposed. Its main idea is depicted in Figure 2.

Here the binary detectors, as those providing fast detection, are used for preliminary filtering of samples. The samples which did not activate any of b -detectors are censored by v -detectors next. It is important to note that we do not expect that b -detectors covers the space \mathcal{N} in sufficient degree, as it can consume too much time. More important aspect is their length. They should be relatively short (with high generalization degree) to detect as quickly as possible the significant part of *nonself* samples. The optimal length, r , of b -detectors can

be determined by studying the phase transition diagram mentioned in Section 2.1. Namely, we choose the r value guaranteeing reasonable value of the P_g probability what, in addition to ease of generating detectors, results in sufficiently high coverage of the \mathcal{N} space.

In the b - v model the overall DR ratio as well as the average time of detection depends mainly on the number of recognized *nonself* samples by “fast” b -detectors. Thus, in the experiments reported later, we focus mainly on these parameters.

3.1 Building b -Detectors in Space \Re^n

Usually, samples are represented as real-valued vectors. Thus, to construct b -detectors, the *self* samples should be converted into binary form first. This can be done in many ways, but probably the simplest one (at least from the computational point of view), is the uniform quantization, [12].

Generally, quantization (used e.g. in digital signal processing, or image processing) refers to the process of approximating a continuous range of values by relatively small set of discrete symbols or integer values. A quantizer can be specified by its input partitions and output levels (called also reproduction points). If the input range is divided into levels of equal spacing, then the quantizer is termed as the uniform quantizer; otherwise it is termed as a non-uniform quantizer, [12].

A uniform quantizer can be described by its lower bound and the number of output levels (or step size). However, in our case, the first of these value is always 0, as we operate only on values from the unit interval (required by the V -Detector algorithm). Moreover, for binary representation of output values, instead of the number of output levels, we should rather specify the parameter bpa , denoting the number of bits reserved for representing a single level.

The quantization function $Q(x)$ for a scalar real-valued observation x , can be expressed as follows:

$$Q(x) = \lfloor x * 2^{bpa} \rfloor, \quad (4)$$

The resulting integer from the range $\{0, 2^{bpa} - 1\}$ is converted to a bit string of length $l = n * bpa$, where n is the dimension of real-valued samples.

3.2 Sliding Window for Real-Valued Samples

Experiments conducted on the datasets involving 30-40 attributes have showed, that V -Detector algorithm with Euclidean or Manhattan distance metrics provides too low DR values (slightly exceeding 50%-60%, in the best cases). Even the use of fractional distance together with higher values of estimated coverage co do not lead to significant improvement of the DR. Note, that due to large number of the attributes involved in the data description it was not possible to construct efficient b -detectors.

In practice, e.g. in spam detection or recognition of handwritten letters, the data are characterized by 50 and more (even up to 250) attributes. In such

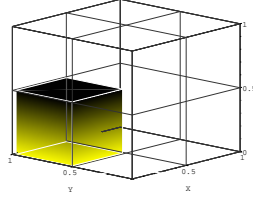


Fig. 3. Representation of b -detector 101 in space \mathbb{R}^3 for $bpa = 1$

cases, probably none of known metrics is able to measure properly the similarity between the data. Thus, the problem of detecting anomalies becomes very hard (or even impossible) to solve, when only “traditional” methods are used.

Our solution is to incorporate the sliding window idea for the v -detectors, to reduce the dimensionality of real-valued samples. This mechanism is already applied for b -detectors and is very popular in e.g. segmentation of time series, [5]. Moreover, it is consistent with one of the features of NIS, according to which the *pathogens* are detected by using only partial information [3].

By using sliding window with length w , the dimensionality of samples can be reduced from n to w . The value of w should be tuned, taking into account the two constraints: (a) it can not be too small as the probability of generating detectors can be too small (similarly to b -detectors), (b) it can not be too high as for higher dimensions, still one can meet problem with finding the suitable metric. Hence, the optimal w value seems to be near to the maximal dimensionality for which chosen metric is able to provide satisfactory proximity distance, i.e. for L_1 and L_2 the most appropriate seems to be w from the range [5, 20].

For example, for $n = 6$ and $w = 4$, each sample vector $\mathbf{x} = (x_1, \dots, x_6)$ ($n = 6$) will be divided on $n - w + 1 = 3$ following parts:

$$x_1, x_2, x_3, x_4; x_2, x_3, x_4, x_5; x_3, x_4, x_5, x_6.$$

3.3 Representation of b -Detectors in Space \mathbb{R}^n

V -Detector algorithm can take into consideration already generated b -detectors, only if they can be represented in space \mathbb{R}^n . In this case, two different shapes of b -detectors in real-valued space were investigated: hyperspheres and hypercubes.

The simplest way of converting b - to v -detector is when $w = r$. Then, the center of b -detector (c_{vb}) in space \mathbb{R}^w can be calculated as follow:

$$c_{vb}[k] = \frac{toInt(b_{k*bp a, (k+1)*bp a-1})}{2^{bp a}} + \frac{1}{2^{bp a+1}}, \quad \text{for } k = 0, \dots, w-1 \quad (5)$$

where $b_{i,j}$ denotes the substring of b -detector from position i , to j and $toInt$ is the function which returns the decimal value of the binary number. Depending

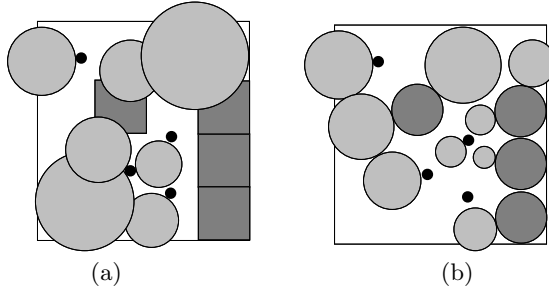


Fig. 4. Example of performing b -detectors in space \mathbb{R}^2 for $bpa = 2$, when (a) only the centers of both detectors do not cover each other, (b) the subspaces occupied by both detectors are disjoint. *self* samples, b - and v -detectors are represented as black, dark gray and light gray circles, respectively.

on used shape, the diameter (in case of hyperspheres) or edge (for hypercubes) is equal to 2^{-bpa} . An example of representation of single b -detector in space \mathbb{R}^3 is presented in Figure 3.

3.4 Minimizing of Overlapping Regions

When b - and v -detectors are generated independently, they could cover the same parts of \mathcal{U} . It means, some subset of generated v -detectors is superfluous and such detectors should be removed. This way we improve duration of classification which depends on the number of detectors.

Let us denote \mathfrak{D}_v and \mathfrak{D}_b the set of detectors built by V -Detector algorithm and those being the real-valued representation of b -detectors in space \mathbb{R}^n , respectively. To minimize the overlapping regions, two approaches were considered (see Figure 4). They differentiate according to the shape of \mathfrak{D}_b detectors and allowed overlapping regions.

In Figure 4a, a candidate for v -detector is build only when its center is located outside all the detectors from subsets \mathfrak{D}_v and \mathfrak{D}_b . In this case the hypercube shape is more appropriate as it is quite easy to check if the detector is covered by any detector \mathfrak{D}_b . Moreover, it is acceptable, that some parts of \mathcal{U} can be covered by 2 detectors: one from each set \mathfrak{D}_v and \mathfrak{D}_b (all items from each set are disjoint, by definition).

Other approach is presented in Figure 4b, where detectors from set \mathfrak{D}_b have the same shape as \mathfrak{D}_v detectors. Additionally, the radius of newly generated v -detectors were calculated as the distance either to the nearest *self* or b -detector. In this way, an overlapping region between \mathfrak{D}_v and \mathfrak{D}_b is empty ($\mathfrak{D}_v \cap \mathfrak{D}_b = \emptyset$).

In both the cases, V -Detector algorithm takes into account the \mathfrak{D}_b detectors already generated. As a result, the assumed coverage co can be achieved in shorter time. Hence, the overall duration of learning process is faster than in case when both types of detector were generated independently.

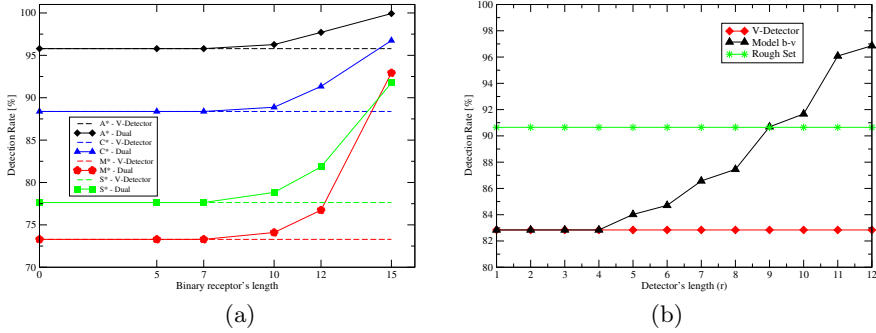


Fig. 5. Comparing DR for original V-Detector algorithm and b -v model: (a) *HWW* dataset, (b) *Diabetes* dataset

4 Experiments and Results

The main question we asked ourself, was: “Is it possible to cover a high-dimensional space \mathcal{S} by both types of detectors, in reasonable time, using various types of similarity metrics? If the answer would be positive, we can expect that proposed approach is suitable for anomaly detection in high-dimensional problems. Our main goal was to maximize the coverage of \mathcal{N} , which should be reflected in DR ratio. Moreover, we were interested also in reducing the classification time, as it is the one of the crucial parameters of on-line classification systems.

Our experiments with b -v model were performed on various multidimensional datasets from UCI Machine Learning Repository (*Spambase*, *Madelon*, *KDD Cup 1999*) as well as from other popular repositories, including *CBF* (Cylinder-Bell-Funnel, Keogh and Kasetty), *HWW* (*Handwritten Words* [15]) and *Diabetes* [13]. Here, we presents only small part our results.

HWW dataset consists of 40 words, each represented by 10 samples. Preliminary experiments has showed, that the recognition of particular words from this dataset is relatively easy task. We took 5 following words: *air*, *name*, *names*, *woman*, *women*. The first word was selected randomly, and next four words have created two pairs of very similar words, which should be rather hard to distinguish. In our experiments, the samples describing one of mentioned word was taken as *self* for which separate sets of detectors were created. Similar approach also was applied for other experiments described in this section.

Fig. 5(a) shows the DR, obtained for the b -detectors with the following parameters: $r \in \{3, 4, 5, 7, 10, 12, 15\}$ and $bpa = 1$. Generally, the results agree with the shape of *phase transition region* mentioned at Section 2.1, but here we achieved $DR \approx 100\%$ even for very short detectors (for $r = 12$ or even less, especially, in comparison to the sample length $l = 60$).

Similar experiments were performed for V -Detector algorithm with Euclidean and Manhattan distance. Also in this case, all the words were recognized with $DR > 98\%$. Thus we can suppose that this dataset contains easily separated groups of samples, representing particular words. We were not impelled to apply fractional distance metrics, as even Euclidean metric gave highly satisfactory results, although in high-dimensional datasets it should provide the less valuable proximity [1].

The similar DR ratios are showed in Figure 5(b). Our results for b - v model are even better than for rough set approach [13], where also *nonself* samples were used at the learning stage.

For all testing dataset, the overall duration of classification was decreased more than 10% in comparing to V -Detector algorithm. It is the result of using the very fast b -detectors which were able to recognize more than half of censored samples.

5 Conclusions

The b - v -model presented in this paper employs the negative selection mechanism, developed within the domain of Artificial Immune Systems. It is designed for anomaly detection in high-dimensional data, which are difficult to analyze due to the lack of appropriate similarity metrics which enable to cover space \mathcal{N} in sufficient degree and reasonable time. One of the important features of b - v model is its ability to minimize the overlapping regions between sets of b - and v -detectors. As a result the overall duration of classification could be significantly reduced as less v -detectors were needed to cover space \mathcal{N} . Hence, this model is more efficient for online classification systems in comparing to standard negative selection approaches which based only on one type of detectors.

Moreover, sliding window applied for both types of detectors can be viewed as a possibility to overcome the scaling problem, what makes this model can be applied to solve even the high-dimensional problems, which usually, were beyond the capabilities of *NSA*.

Acknowledgment. This work was supported by Bialystok University of Technology grant S/WI/5/08.

References

1. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the Surprising Behavior of Distance Metrics in High Dimensional Space. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, pp. 420–434. Springer, Heidelberg (2001)
2. Balthrop, J., Esponda, F., Forrest, S., Glickman, M.: Coverage and generalization in an artificial immune system. In: Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2002), New York, July 9–13, pp. 3–10 (2002)
3. de Castro, L., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer (2002)

4. Chmielewski, A., Wierzchoń, S.T.: On the distance norms for multidimensional dataset in the case of real-valued negative selection application. *Zeszyty Naukowe Politechniki Białostockiej* (2), 39–50 (2007)
5. Dasgupta, D., Forrest, S.: Novelty detection in time series data using ideas from immunology. In: *Fifth International Conf. on Intelligent Systems*, Reno, Nevada, June 19–21 (1996)
6. Forrest, S., Hofmeyr, S.A., Somayaji, A., Longstaff, T.A.: A sense of Self for Unix Processes. In: *Proc. of the 1996 IEEE Symposium on Research in Security and Privacy*, pp. 120–128. IEEE Computer Society Press (1996)
7. Forrest, S., Perelson, A., Allen, L., Cherukuri, R.: Self-nonsel self discrimination in a computer. In: *Proc. of the IEEE Symposium on Research in Security and Privacy*, Los Alamitos, pp. 202–212 (1994)
8. Harmer, P.K., Williams, P.D., Gunsch, G.H., Lamont, G.B.: Artificial immune system architecture for computer security applications. *IEEE Trans. on Evolutionary Computation* 6, 252–280 (2002)
9. Hofmeyr, S., Forrest, S.: Architecture for an Artificial Immune System. *Evolutionary Computation J.* 8(4), 443–473 (2000)
10. Ji, Z., Dasgupta, D.: Real-Valued Negative Selection Algorithm with Variable-Sized Detectors. In: Deb, K., Tari, Z. (eds.) *GECCO 2004, Part I. LNCS*, vol. 3102, pp. 287–298. Springer, Heidelberg (2004)
11. Ji, Z., Dasgupta, D.: Revisiting negative selection algorithms. *Evolutionary Computation* 15(2), 223–251 (2007)
12. Sayood, K.: *Introduction to Data Compression*. Elsevier (2005)
13. Stepaniuk, J.: Rough Set Data Mining of Diabetes Data. In: Raś, Z.W., Skowron, A. (eds.) *ISMIS 1999. LNCS*, vol. 1609, pp. 457–465. Springer, Heidelberg (1999)
14. Stibor, T.: Phase Transition and the Computational Complexity of Generating r -Contiguous Detectors. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) *ICARIS 2007. LNCS*, vol. 4628, pp. 142–155. Springer, Heidelberg (2007)
15. Tabedzki, M., Rybnik, M., Saaed, K.: Method for handwritten word recognition without segmentation. *Polish J. of Environmental Studies* 17, 47–52 (2008)
16. Timmis, J., Hone, A., Stibor, T., Clark, E.: Theoretical advances in artificial immune systems. *Theoretical Computer Science* 403(1), 11–32 (2008)
17. Wierzchoń, S.T.: Generating optimal repertoire of antibody strings in an artificial immune system. In: Kłopotek, M.A., Michalewicz, M., Wierzchoń, S.T. (eds.) *Proc. of the IIS 2000 Symposium on Intelligent Information Systems*, Bystra, Poland, June 12–16, pp. 119–133. Springer (2000)
18. Wierzchoń, S.T.: Deriving concise description of non-self patterns in an artificial immune system. In: Jain, L.C., Kacprzyk, J. (eds.) *New Learning Paradigm in Soft Comptuning*, pp. 438–458. Physica-Verlag (2001)

Spectral Clustering Based on k -Nearest Neighbor Graph

Małgorzata Lucińska¹ and Sławomir T. Wierzchoń^{2,3}

¹ Kielce University of Technology, Kielce, Poland

² Institute of Computer Science Polish Academy of Sciences, Warsaw, Poland

³ University of Gdańsk, Gdańsk, Poland

Abstract. Finding clusters in data is a challenging task when the clusters differ widely in shapes, sizes, and densities. We present a novel spectral algorithm **Specclus** with a similarity measure based on modified mutual nearest neighbor graph. The resulting affinity matrix reflex the true structure of data. Its eigenvectors, that do not change their sign, are used for clustering data. The algorithm requires only one parameter – a number of nearest neighbors, which can be quite easily established. Its performance on both artificial and real data sets is competitive to other solutions.

Keywords: Spectral clustering, nearest neighbor graph, signless Laplacian.

1 Introduction

Clustering is a common unsupervised learning technique; its aim is to divide objects into groups, such that members of the same group are more similar each to another (according to some similarity measure) than any two members from two different groups. Different applications of clustering in practical problems are reviewed e.g. in [7]. The technique is successfully used for e.g. in management for risk assessment [13] and [1] or in portfolio management [19]. Although many clustering methods have been proposed in the recent decades, see e.g. [6] or [17], there is no universal one that can deal with any clustering problem, since the real world clusters may be of arbitrary complicated shapes, varied densities and unbalanced sizes.

Spectral clustering techniques [15] belong to popular and efficient clustering methods. They allow to find clusters even of very irregular shapes, contrary to other algorithms, like k -means algorithm [8]. Spectral techniques use eigenvalues and eigenvectors of a suitably chosen matrix to partition the data. The matrix is the affinity matrix (or a matrix derived from it) built on the basis of pairwise similarity of objects to be grouped. The structure of the matrix plays a significant role in correct cluster separation. If it is clearly block diagonal, its eigenvectors will relate back to the structural properties of the set of the objects, [10].

One of the key tasks in spectral clustering is the choice of similarity measure. Most spectral algorithms adopt a Gaussian kernel function defined as:

$$S(i, j) = \exp \left(- \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right) \quad (1)$$

where $\|\mathbf{x}_i - \mathbf{x}_j\|$ denotes the Euclidean distance between points \mathbf{x}_i and \mathbf{x}_j . The kernel parameter σ influences the structure of an affinity matrix and generally it is difficult to find its optimal value. Some authors propose a global value of σ for the whole data set e.g. [12] and [14] while the others suggest using a local parameter e.g. [18]. However both the solutions fail to reveal the properties of real world data sets [16]. Another open issue of key importance in spectral clustering is that of choosing a proper number of groups. Usually this number is a user defined parameter [12], but sometimes it is estimated – with varying success rate [14] – in a heuristically motivated way.

In this paper we present a spectral clustering algorithm **Specclus** that can simultaneously address both of the above mentioned challenges for a variety of data sets. It adopts the idea derived by Shi *et al.* from their analysis of the relationship between a probability distribution and spectrum of the corresponding distribution-dependent convolution operator, [14]. Their DaSpec (i.e. Data Spectroscopic) algorithm estimates the group number by finding eigenvectors with no sign change and assigns labels to each point based on these eigenvectors. In our algorithm the similarity between pairs of points is deduced from their neighborhoods. The use of similarity based on nearest neighbors approach removes, at least partially, problems with cluster varying densities and the unreliability of distance measure. Resulting adjacency matrix reflects true relationships between data points. Also the σ parameter is replaced by the number of neighbors parameter, which can be chosen more simply since it is an integer and takes a small number of values. Apart from only one parameter another advantage of the presented approach is that it incorporates a variety of recent and established ideas in a complete algorithm which is competitive to current solutions.

In section 2 the notation and related work is presented, the next section explains the main concepts used in the **Specclus** algorithm, which is presented in details in section 4. Then, in section 5, we compare performance of our algorithm with other solutions. Finally, in section 6, the main conclusions are drawn.

2 Notation and Related Work

Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ be the set of data points to be clustered. For each pair of points i, j an adjacency $a_{ij} \in \{0, 1\}$ is attached (see Section 3 for details). The value $a_{ij} = 1$ implies the existence of undirected edge $i \sim j$ in the graph G spanned over the set of vertices \mathbf{X} . Let $A = [a_{ij}]$ be the adjacency matrix. Let $d_i = \sum_j a_{ij}$ denote the degree of node i and let D be the diagonal matrix with d_i 's on its diagonal. A clustering $\mathcal{C} = (C_1, C_2, \dots, C_l)$ is a partition of \mathbf{X} into l nonempty and mutually disjoint subsets. In the graph-theoretic language the clustering represents a multiway cut in G [3].

In the **Specclus** algorithm a signless Laplacian $M = D + A$, introduced by Cvetković [2], is used. Cvetković proves that the spectrum (i.e. the set of eigenvalues) of M can better distinguish different graphs than spectra of other commonly

used graph matrices. Graphs with the same spectrum of an associated matrix B are called cospectral graphs with respect to B , or B -cospectral graphs. A graph H cospectral with a graph F , but not isomorphic to F , is called a cospectral mate of H . Let \mathcal{G} be a finite set of graphs, and let \mathcal{G}' be the set of graphs in \mathcal{G} which have a cospectral mate in \mathcal{G} with respect to M . The ratio $|\mathcal{G}'|/|\mathcal{G}|$ is called the spectral uncertainty of (graphs from) \mathcal{G} with respect to B . Cvetković compares spectral uncertainties with respect to the adjacency matrix, the Laplacian ($L = D - A$), and the signless Laplacian of sets of all graphs on n vertices for $n \leq 11$. Spectral uncertainties in case of the signless Laplacian are smaller than for the other matrices. This indicates that the signless Laplacian seems to be very convenient for use in studying graph properties.

As already mentioned the **Specplus** algorithm utilizes the idea proposed by Shi *et al.* in [14]. They study the spectral properties of an adjacency matrix A and its connection to the data generating distribution P . The authors investigate the case when the distribution P is a mixture of several dense components and each mixing component has enough separation from the others. In such a case A and L are (close to) block-diagonal matrices. Eigenvectors of such block-diagonal matrices keep the same structure. For example, the few top (i.e. corresponding to highest eigenvalues) eigenvectors of L can be shown to be constant on each cluster, assuming infinite separation between clusters. This property allows to distinguish the clusters by looking for data points corresponding to the same or similar values of the eigenvectors. Shi *et al.* develop in [14] theoretical results based on a radial similarity function with a sufficiently fast tail decay. They prove that each of the top eigenvectors of A corresponds exactly to one of the separable mixture components. The eigenvectors of each component decay quickly to zero at the tail of its distribution if there is a good separation of components. At a given location \mathbf{x}_i in the high density area of a particular component, which is at the tails of other components, the eigenvectors from all other components should be close to zero.

Also Elon [4] attempts to characterize eigenvectors of the Laplacian on regular graphs. He suggests that the distribution of eigenvectors, except the first one, follows approximately a Gaussian distribution. There are also proofs that in general, top eigenvalues have associated eigenvectors which vary little between adjacent vertices. The two facts confirm the assumption that each cluster is reflected by at least one eigenvector with large components associated with the cluster vertices and almost zero values in the other case.

Another concept incorporated in the **Specplus** algorithm comes from Newman. It concerns a quality function called modularity, which is used for assessing a graph cut [11].

Another concept incorporated in the **Specplus** algorithm is so-called modularity, i.e. a quality function introduced by Newman [11] for assessing a graph cut. According to its inventor a good division of a graph into partitions is not merely one in which there are few edges between groups; it is one in which there are fewer than expected edges between groups. The modularity Q is, up to a multiplicative constant, the number of edges falling within groups minus the

expected number in an equivalent graph with edges placed at random, or in functional form

$$Q = \frac{1}{2m} \sum_{ij} \left[a_{ij} - \frac{d_i d_j}{2m} \right] \delta(g_i, g_j) \quad (2)$$

where $\delta(r, s) = 1$ if $r = s$ and 0 otherwise, and m is the number of edges in the graph. Newman suggests that a division on a graph makes sense if $Q > 0.3$.

3 Neighborhood Graph and Structure of Its Eigenvectors

The novel concept of the **Specclus** algorithm is the similarity measure based on nearest neighbors approach. Specifically the k mutual nearest neighbor graph is constructed with points as the vertices and edges as similarities. First for each of the points k symmetric nearest neighbors are found with Euclidean distance as the distance metric. Then for each two vertices \mathbf{x}_i and \mathbf{x}_j the connecting edge v_{ij} is created if vertex \mathbf{x}_i belongs to k -nearest neighbors of vertex \mathbf{x}_j and vice versa. Afterwards vertices with a small number of edges (less than half of an average number of edges connected to one point in the graph) are identified. Each of such vertices with low degree is additionally connected to a few nearest neighbors of vertices in its closest proximity. By “closest proximity” we understand approximately the first $k/2$ neighboring vertices and half of its neighbors create additional connections, but only in case their degree is less than $k/2$. The resulting graph is similar to a mutual nearest neighbor graph described in [9]. The difference lies in additional edges, which are created between vertices with low degrees. For each pair of nodes \mathbf{x}_i and \mathbf{x}_j in such constructed graph the value a_{ij} is set to one if and only if there is an edge joining the two vertices. Otherwise a_{ij} equals 0. Also all diagonal elements of the affinity matrix A are zero.

Such an approach guarantees a sparse affinity matrix, capturing the core structure of the data and achieved simply with only one parameter k . It can also handle data containing clusters of differing densities. To illustrate this statement let us consider two neighboring clusters: a dense cluster A and a sparse cluster B , as Figure 1 shows. The point P does not belong to the mutual nearest neighbors of the point A , as the last one has many other neighbors closer to it than P . In such a case lacking neighbors of the point P will be supplemented by the nearest neighbors of points Q and R .

In order to estimate the number of groups and divide data into clusters the **Specclus** algorithm utilizes structure of the top eigenvectors of signless Laplacian. According to works [4] and [14], and our extensive numerical observations, top eigenvectors of sparse matrices, related to points creating disjoint subsets, reflect the structure of the data set. Figure 2 shows an ideal example, when three clusters are completely separated and each of them can be presented in the form of the regular graph of the same degree. Top eigenvectors of signless Laplacian show clearly its structure. Each cluster is represented by an eigenvector, which assumes relatively high values (of one sign) for points belonging to the cluster



Fig. 1. Choice of mutual nearest neighbors in case of two clusters with different densities

and zero values for points from other clusters. An additional regularity can also be seen – if a point is close to a cluster center its value in the corresponding eigenvector is high. The points, which lay at the border of a cluster have relatively small values of the appropriate eigenvector.

In real situations, when subsets are close to each other, overlap or have different densities, the picture of data structure given by the top eigenvectors can be a little confusing. Shi *et al.* notice that smaller or less compact groups may not be identified using just the very top part of the spectrum. More eigenvectors need to be investigated to see these clusters. On the other hand, information in the top few eigenvectors may also be redundant for clustering, as some of these eigenvectors may represent the same group. In the **Speclus** algorithm the problems are solved with the help of modularity function. If two eigenvectors indicate two different divisions of the set, the modularity is calculated in order to choose a better cut in terms of modularity maximization.

4 The Speclus Algorithm

The steps of the **Speclus** algorithm are as follows:

The Speclus algorithm

Input: Data X , number of nearest neighbors k

Output: C clustering of X

Algorithm:

1. Compute, in the following order
 - k -nearest neighbors for each x
 - mutual nearest neighbors for each x
 - additional neighbors in case degree of $x < \text{half of average degree in } X$
2. Create affinity matrix A and signless Laplacian $M=D+A$
3. Compute top w eigenvectors of M

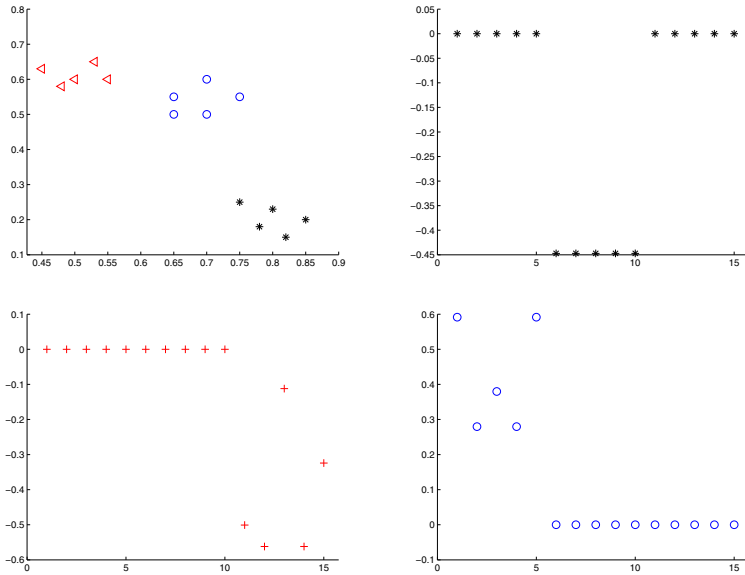


Fig. 2. Perfectly separated clusters (top right) and their eigenvectors (top left and bottom)

4. Find eigenvectors with no sign change (up to standard deviation of its values)
5. Determine overlapping eigenvectors (related to the same cluster)
6. Calculate modularity corresponding to the graph cut for each overlapping eigenvector and choose the best one
7. Create a set of eigenvectors A , each representing one cluster
8. Assign each point x to one eigenvector from the set A , having the biggest entry for x

The algorithm builds a graph, with points as vertices and similarities between points as edges. It starts by retrieving k nearest neighbors for each point and afterwards generates mutual nearest neighbors. If a degree of a vertex is smaller than half of the average degree of vertices in the graph, edges to nearest neighbors of the vertices from the closest proximity are added, as it is described in section 3. After determining the affinity matrix and signless Laplacian the eigenvectors and eigenvalues of the last one are calculated. The number of eigenvectors computed, $w = 20$ is estimated as twice the maximum expected number of clusters, that guarantees representation of each cluster by at least one eigenvector. Next, eigenvectors with no sign change are extracted. We assume that an eigenvector does not change a sign if all its positive entries are smaller than its standard deviation or absolute values of its all negative entries do not exceed the standard deviation. If the clusters are not perfectly separated or have varying densities one

cluster may be represented by a few eigenvectors. Such overlapping eigenvectors are recognized with a help of a point with the biggest entry for each eigenvector (eigenvector maximum). As it is mentioned in the section 3 of this paper, points located in the center of a cluster have big entries in appropriate eigenvectors. A point corresponding to the maximum of one eigenvector should have small entries in the other eigenvectors, unless they represent the same cluster. After establishing the maximum of an eigenvector \mathbf{v} we compare its values in the other eigenvectors. If the appropriate entry in an eigenvector \mathbf{w} is bigger than a small value ϵ , e.g. $\epsilon = 0.001$, it means that the two eigenvectors overlap. Such pairs of eigenvectors create a set B , while the eigenvectors, which do not overlap belong to a set A . First the data set is divided into clusters on the basis of the set A . If a point \mathbf{x} has the biggest entry in the eigenvector \mathbf{v} it receives a label v , etc. Afterwards similar divisions are made with a use of each overlapping eigenvector pair from the set B . Let us assume that eigenvectors $\mathbf{v1}$ and $\mathbf{v2}$ overlap with each other. A point, which has an entry in $\mathbf{v1}$ bigger than ϵ is labeled as a set $C1$, if the entry in $\mathbf{v2}$ is bigger the label corresponds to a set $C2$. For each of the two divisions a modularity function is calculated. The eigenvector, which leads to better division in terms of modularity function is added to the set A . Eigenvectors from this set are used for the final labeling of the data. The number of eigenvectors included in the set A indicates the number of groups. Each eigenvector represents one cluster. Each point is labeled according to the eigenvector with the highest entry for the point.

Computational complexity of the proposed algorithm is relatively small. First of all the affinity matrix is very sparse as we use the concept of mutual neighbors. Second the number of needed eigenvectors is relatively small, if we consider clusters of reasonable size only, i.e. if we require that the minimal cluster size exceeds 1 percent of the size of the whole data set. Moreover, in case of a signless Laplacian we seek for top eigenvectors, which are easier to find than eigenvectors corresponding to smallest eigenvalues. In such situation solving the eigen problem even for large data set is not very time consuming. The other steps of the algorithm take time $O(n)$ each. So the solution is scalable.

5 Experimental Results

We have compared the performance of the **Specclus** algorithm (implemented in MATLAB) to three other methods: the Ng *et al.* algorithm [12], the Fischer *et al.* algorithm [5], and the DaSpec algorithm. The first one is a standard spectral algorithm, which uses normalized Laplacian $L = D^{1/2}SD^{1/2}$ and k -means algorithm for final clustering. The second one aims at amplifying the block structure of affinity matrix by context-dependent affinity and conductivity methods. The DaSpec algorithm uses the same properties of eigenvectors as the **Specclus** algorithm and similarly does not need a cluster number to be given in advance. The first two algorithms need a number of clusters as an input parameter.

In the case of the three algorithms the σ parameter should be carefully established. Ng *et al.* and Shi *et al.* have proposed heuristics to calculate the value

of the σ parameter. For many data sets neither of the formulas can guarantee correct classification. In order to compare the best achievements of all the algorithms the values of the σ parameter were chosen manually, as described by Fischer *et al.* For each data set they systematically scanned a wide range of σ 's and ran the clustering algorithms. We use their results in case of the first two algorithms.

All the algorithms are evaluated on a number of benchmark data sets identical as in [5]. Six of the sets are artificial and three are real-world problems. They cover a wide range of difficulties, which can be met during data segmentation. The first data set 2R3D.2 is obtained by dispersing points around two interlocked rings in 3D. The dispersion is Gaussian with standard deviation equal 0.2. The 2RG data set consists of two rather high density rings and a Gaussian cluster with very low density. The 2S set is created by two S-shaped clusters with varying densities within each group. Sets 4G and 5G have four and five Gaussian clusters each of different density in 3D and 4D respectively. 2Spi is a standard set used for evaluation of spectral clustering algorithms consisting of two spirals with double point density. The last three sets are common benchmark sets with real-world data: the iris, the wine and the breast cancer. The first one consists of three clusters, two of which are hardly separated. The wine is a very sparse set with only 178 points in 13 dimensions.

Table 1. Number of incorrectly clustered points for Ng *et al.*, Fischer *et al.*, DaSpec, and **Specclus** algorithms. n denotes number of points, l – number of clusters, and D – data dimension.

Data	n	l	D	Ng <i>et al.</i>	Fischer <i>et al.</i>	DaSpec	Specclus
2R3D.2	600	2	3	4	93	195	11
2RG	290	3	2	101	0	180	0
2S	220	2	2	0	0	70	0
4G	200	4	3	18	1	41	2
5G	250	5	4	33	11	53	11
2Spi	386	2	2	0	193	191	0
Iris	150	3	4	14	7	35	14
Wine	178	3	13	3	65	89	9
BC	683	2	9	22	20	239	21

As can be seen from Table 1 the **Specclus** algorithm is the most flexible one and performs well independently on data set structure. Although both the **Specclus** and the DaSpec algorithms use the same concept of eigenvector properties the second one often fails on real-world data or clusters with different densities. For sets presented in Table 1 it usually is not able to detect all the clusters. The dramatic differences in the performance between the two algorithms can be explained as a result of the use of signless Laplacian and special similarity

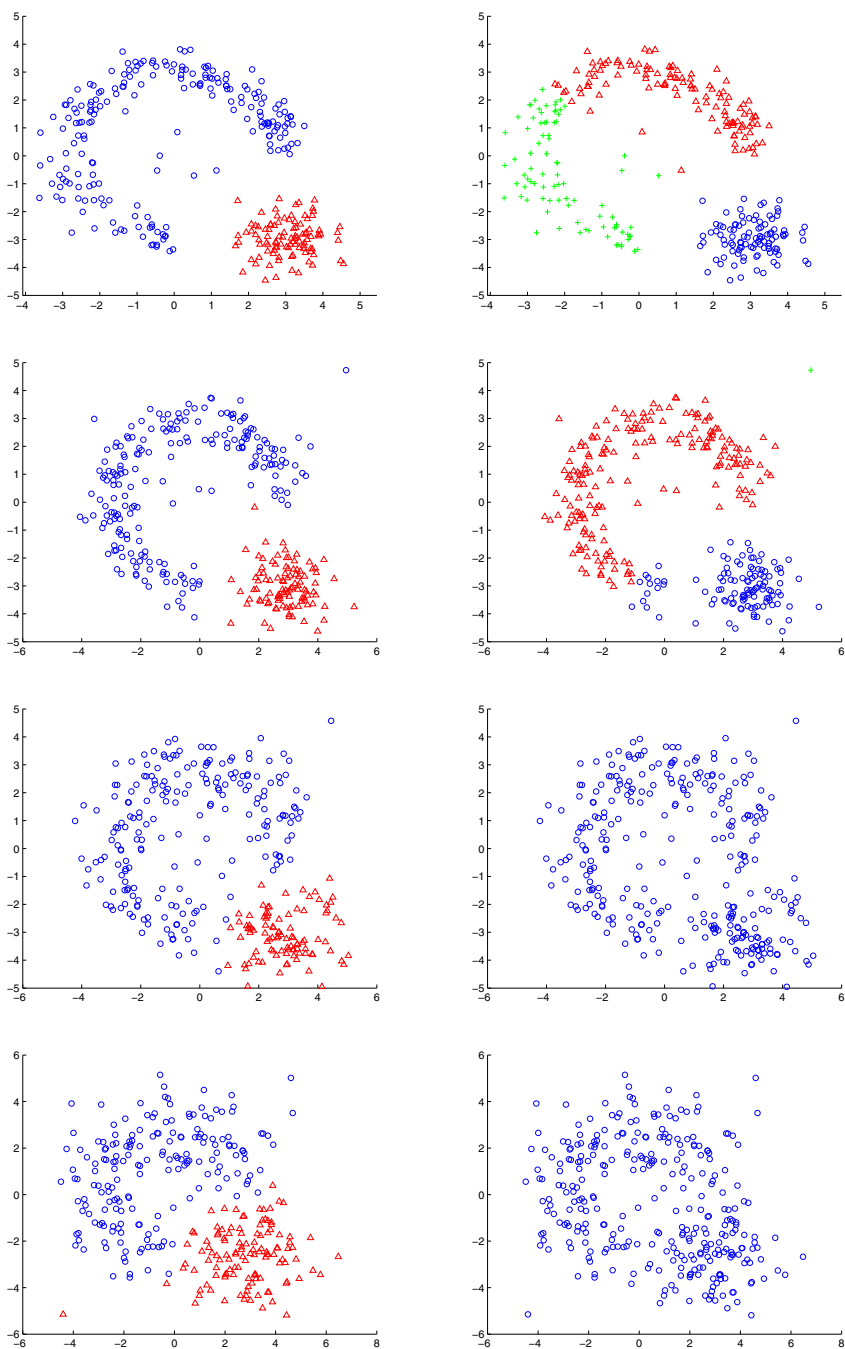


Fig. 3. Performance of Specus (left) and DaSpec (right) for artificial data sets DS1, DS2, DS3, and DS4

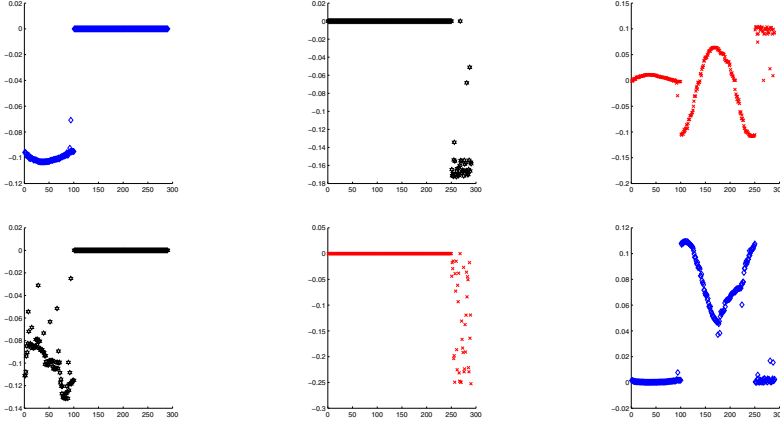


Fig. 4. Comparison of ordinary Laplacian (top) and signless Laplacian (bottom) eigenvectors for 2RG dataset

measure in the **Speclus** algorithm. The signless Laplacian spectrum pictures better data structure than spectrum of any other graph matrix. The similarity measure based on the mutual neighbors concept caused the affinity matrix to be more clearly block diagonal. Whether density of points varies meaningfully with or within clusters, our method of constructing affinity matrix gives good results. Even in case of very sparse data or high Gaussian noise the labeling of data by **Speclus** is correct. If data are sparse, adding edges between vertices with few connections and neighbors of vertices from their closest proximity avoids lost of graph connectivity and isolation of some points. On the other hand creating connections between graph vertices on the basis of mutual nearest neighbors eliminates influence of any noises in the data.

We also compare the performance of the **Speclus** algorithm with the DaSpec algorithm on the basis of sets suggested by Shi *et al.* in [14], that contain non-Gaussian groups and various levels of noise. We use the set *DS1*, that consists of three well-separable groups in \mathbb{R}^2 . The first group of data is generated by adding independent Gaussian noise $N((0,0)^T, 0.15^2\mathbb{I})$ to 200 uniform samples from three fourth of a ring with radius 3. The second group includes 100 data points sampled from a bivariate Gaussian $N((3, -3)^T, 0.5^2\mathbb{I})$ and the last group has only five data points sampled from a bivariate Gaussian $N((0,0)^T, 0.3^2\mathbb{I})$. Here \mathbb{I} stands for the unit matrix. Given *DS1*, three more data sets (*DS2*, *DS3*, and *DS4*) are created by gradually adding independent Gaussian noise (with standard deviations 0.3, 0.6, 0.9 respectively). The results obtained for the four data sets with the **Speclus** algorithm are shown in the left column and with the DaSpec algorithm in the right column of Figure 3. It is clear that the degree of separation decreases from top to bottom. The divisions resulting from our algorithm are more correct than in the case of the other algorithm. However, neither of them is able to separate the five points inside the part of the ring. But

even for the highest level of noise the **Speclus** algorithm finds the right number of groups.

At last we show how performance of the **Speclus** algorithm changes if we use ordinary Laplacian $L = D - A$ instead of signless Laplacian $M = D + A$. In Figure 4 there are eigenvectors of Laplacian L and Laplacian M , which are used for partitioning of the set 2RG. The data structure is perfectly illustrated by signless Laplacian eigenvectors, indicating three separate clusters. Ordinary Laplacian eigenvectors indicate only two clusters, whereas the third one does not have any clear representation. This result constitutes an experimental proof, that signless Laplacian is more suitable for partitioning sets with varying densities than Laplacian L .

6 Conclusions and Future Work

We have presented a new spectral clustering algorithm, which uses signless Laplacian eigenvectors and a novel affinity matrix.

The matrix is created on the basis of a mutual nearest neighbor graph with additional edges connecting points from sparse density areas. Experiments confirm that a good similarity measure is crucial to the performance of spectral clustering algorithms. Our solution correctly separates different types of clusters with varying densities with and within groups, being simultaneously noise-resistant. It has only one parameter, which is quite easy to establish. The **Speclus** algorithm does not require a group number as an input parameter and estimates it correctly using eigenvectors structure and a modularity function.

These observations show that our algorithm is a good candidate to apply it to image segmentation, that will be our next task.

References

1. Chen, Y., Jensen, C.D., Gray, E., Seigneur, J.M.: Risk Probability Estimating Based on Clustering, Technical Report No. TCD-CS-2003-17, Trinity College Dublin (2003)
2. Cvetković, D.: Signless Laplacians and line graphs. *Bull. Acad. Serbe Sci. Arts, Cl. Sci. Math. Natur., Sci. Math.* 131(30), 85–92 (2005)
3. Deepak, V., Meila, M.: Comparison of Spectral Clustering Methods. UW TR CSE-03-05-01 (2003)
4. Elon, Y.: Eigenvectors of the discrete Laplacian on regular graphs a statistical approach. *J. Phys. A: Math. Theor.* 41 (2008)
5. Fischer, I., Poland, J.: Amplifying the Block Matrix Structure for Spectral Clustering. Technical Report No. IDSIA-03-05, Telecommunications Lab (2005)
6. Jain, A., Murty, M., Flynn, P.: Data clustering: A review. *ACM Computing Surveys* 31, 264–323 (1999)
7. Jain, A.: Data clustering: 50 years beyond K-means. *Pattern Recognition Letters* 31, 651–666 (2010)
8. MacQueen, L.: Some methods for classification and analysis of multivariate observations. In: LeCam, L., Neyman, J. (eds.) 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)

9. Maier, M., Hein, M., von Luxburg, U.: Cluster Identification in Nearest-Neighbor Graphs. In: Hutter, M., Servedio, R.A., Takimoto, E. (eds.) ALT 2007. LNCS (LNAI), vol. 4754, pp. 196–210. Springer, Heidelberg (2007)
10. Meila, M., Shi, J.: A random walks view of spectral segmentation. In: Proc. of 10th International Workshop on Artificial Intelligence and Statistics (AISTATS), pp. 8–11 (2001)
11. Newman, M.E.J.: Detecting community structure in networks. *European Physics J. B* 38, 321–330 (2004)
12. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: *Advances in Neural Information Processing Systems* 14, pp. 849–856 (2001)
13. Sanchez-Silva, M.: Applicability of Network Clustering Methods for Risk Analysis. In: Topping, B.H.V., Tsompanakis, Y. (eds.) *Soft Computing in Civil and Structural Engineering*, pp. 283–306. Saxe-Coburg Publications, Stirlingshire (2009)
14. Shi, T., Belkin, M., Yu, B.: Data spectroscopy: eigenspace of convolution operators and clustering. *The Annals of Statistics* 37(6B), 3960–3984 (2009)
15. von Luxburg, U.: A tutorial on spectral clustering. *J. Statistics and Computing* 17(4), 395–416 (2007)
16. Xia, T., Cao, J., Zhang, Y., Li, J.: On defining affinity graph for spectral clustering through ranking on manifolds. *Neurocomputing* 72(13–15), 3203–3211 (2008)
17. Xu, R., Wunsch II, D.: Survey on clustering algorithms. *IEEE Trans. on Neural Networks* 16(3), 645–678 (2005)
18. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: *Proc. of NIPS 2004*, pp. 1601–1608 (2004)
19. Zhang, J.: A Clustering Application in Portfolio Management. *Lecture Notes in Electrical Engineering*, vol. 60, pp. 309–321 (2010)

A New Scale for Attribute Dependency in Large Database Systems

Soumya Sen¹, Anjan Dutta¹, Agostino Cortesi¹, and Nabendu Chaki²

¹ University of Calcutta, Kolkata, India

² Universita Ca Foscari, Venice, Italy

{iamsoumyasen, anjanshines}@gmail.com,
cortesi@unive.it, nabendu@ieee.org

Abstract. Large, data centric applications are characterized by its different attributes. In modern day, a huge majority of the large data centric applications are based on relational model. The databases are collection of tables and every table consists of numbers of attributes. The data is accessed typically through SQL queries. The queries that are being executed could be analyzed for different types of optimizations. Analysis based on different attributes used in a set of query would guide the database administrators to enhance the speed of query execution. A better model in this context would help in predicting the nature of upcoming query set. An effective prediction model would guide in different applications of database, data warehouse, data mining etc. In this paper, a numeric scale has been proposed to enumerate the strength of associations between independent data attributes. The proposed scale is built based on some probabilistic analysis of the usage of the attributes in different queries. Thus this methodology aims to predict future usage of attributes based on the current usage.

Keywords: Materialized view, Query Processing, Attribute dependency, Numeric scale, Query Optimization.

1 Introduction

Success of any large database application depends on the efficiency of storing the data and retrieving the same from the database. Contemporary database applications are expected to support fast response and low turn-around time irrespective of the mediums and applications. Speeding up the query processing in distributed environment is even more challenging. Users demand high speed execution over internet, mobile phone or any other modern electrical gadgets. Fetching of closely related data attributes together would help to reduce the latency. This would be particularly significant to reduce communication cost for query processing in a distributed database. The core technology proposed in this paper could further be extended in cloud computing environment where data is distributed in different data centers. Faster query processing in cloud computing environment result in quick service processing to the users. Each of these diverse application platforms have

specific, and distinct features and could be differentiated based on their nature. Thus not only the data stored in the database is subject of interest. Proper analysis of different run time parameters of the execution environment could excel the performance.

In this paper, a numeric scale has been proposed to measure the degree of association among attributes based on their usage in recent queries. This forms the foundation for several optimization aspects that could improve different database perspectives such as building materialized view, maintain indexes, formation of database clusters etc.

Attribute is the most granular form of representing data in database applications. Thus this analysis based on attributes give a deep insight of the system. Hence the deployment of optimization techniques based on this scale would help to improve the performance of the application from the granular level. Once developed, this numeric scale could be rebuilt dynamically depending on the changing nature of the queries over time. The proposed scale takes into consideration the independent characteristics of diverse applications or execution environment. Hence, by incorporating the assumptions and constraints of the specific system, this scale could be used in heterogeneous applications.

The rest of this paper is organized in several sections after this brief introduction. Section 2 describes different existing work on optimization of query execution. Section 3 contains the proposed methodology of constructing the numeric scale. In section 4, the selection of parameters is discussed along with the complexity analysis of this method. In section 5, the entire process is illustrated through an example. The concluding remarks in section 6 summarize the work and mention the future extensions and applications of the proposed methodology.

2 Related Work

Optimizing the query processing in large data centric application has traditionally been studied under the name of query optimization. Several works has been reported on this area. Initially the focus was on simple database. Over the time, multiple aspects including diverse performance criteria and constraints as well as the requirements for specific applications are taken into consideration. In the rest of this section, a brief survey work has been presented focused on query optimization.

A method of query processing and standardization is proposed in [1] where query graphs are used to represent the queries. These are converted to query trees which in turn are represented in canonical vector form. These trees are optimized to enhance the performance. Another graph based model used in this context help in further optimization by considering the parameters like relation size, tuple size, join selectivity factors [2]. An algorithm has also been proposed in [2] to find a near optimal execution plan in polynomial time. The time complexity of analyzing the graph is often quite high. In [2], instead of considering the whole execution path, selective paths are processed. Genetic algorithm and heuristic approaches are also used for query optimization [3, 4]. A genetic algorithm [3] to minimize the data

transmission cost required for a distributed set query processing is presented. This work is also a contribution in the distributed database application. On the other hand, a heuristic approach [4] is proposed to efficiently derive execution plans for complex queries. These works take into account presence of index and goes beyond simple join reordering. Mathematical model is also helpful in this context. Tarski Algebra [5] along with graphical representation of query is used to achieve efficient query optimization. Another graph based approach is shown to optimize the linear recursive queries [6] in SQL. This approach computes transitive closure of a graph and computes the power matrix of its adjacency matrix. Using this [6], optimization plan is evaluated for four types of graphs: binary tree, list, cyclic graph and complete graph. In recent past, the distributed query optimization gets serious research attention as many of the current applications run in distributed environment. A multi-query optimization aspect for distributed similarity query processing [7] attempts to exploit the dependencies in the derivation of a query evaluation plan. A four-step algorithm [7] is proposed to minimize the response time and towards increasing the parallelism of I/O and CPU.

Creation of materialized view and its archival in fast cache also helps in reducing the query access time by prior assessment of the data that is frequently accessed. The view management process would be more effective if such data can be included in the view that is likely to be accessed in near future. An algorithm was proposed by Yang, et. al. that utilizes a Multiple View Processing Plan (MVPP) [8] to obtain an optimal materialized view selection. The objective had been to achieve the combination of good performance and low maintenance cost. However, Yang's approach did not consider the system storage constraints. Gupta proposed a greedy algorithm [9] to incorporate the maintenance cost and storage constraint in the selection of data warehouse materialized views. The AND-OR view graphs were used [9] to represent all possible ways to generate warehouse views such that the best query path can be utilized to optimize query response time.

Materialized views are built to minimize the total query response time while there is an associated overhead towards the creation and maintenance of these views. Thus, an effort has always been to balance a strike between optimizing the processing cost [10] and time for view selection vis-à-vis increasing the efficiency of query processing [11] by utilizing well-organized materialized view.

Use of index also helps to achieve faster data processing. The application of bitmap index [12] helps in query processing for data warehouse and decision-making systems. The work proposed in [13], couples the materialized view and index selection to take view-index interactions into account and achieve efficient storage space sharing.

However, none of these existing works of query optimization and query processing is based on the outcome of a quantitative analysis on the intensity of association between the attributes. The work proposed in this paper, therefore, aims towards finding a numeric measure to assess such inter-attribute associations. This numeric scale provides the relationship between attributes in terms of both present and future usage. Hence this scale could be applicable in any optimization methods that involve the association of a set of attributes. The knowledge of this scale could be used in building and maintenance of materialized views or indexes.

3 Attribute Scaling

The motivation behind this work is to create a numeric scale to represent the degree of associations between different attributes based on a set of queries. This scale would help to generate different materialized views based on the requirement of the users. The proposed methodology of constructing this numeric scale is explained using a seven step algorithm named *Numeric_Scale* (described in section 3.1). In the pre-processing phase, a set of queries (say m number of queries) are picked from the recent queries evoked in an application. Say, a total of n numbers of attributes participate in the query set. The proposed scale is based on these n attributes.

The 1st step of the proposed algorithm builds the Query Attribute Usage Matrix (QAUM), which shows what attributes are used by which queries (described in section 3.2). In the next step, mutual dependencies among every pair of attributes are computed, yielding to the Attribute Dependency Matrix (ADM) (described in 3.3). This is a symmetric matrix. Based on the result of ADM a new matrix, called Probability Distribution Matrix (PDM), is computed. PDM shows the dependencies among every pair of attributes based on a probabilistic function (described in 3.4). This is followed by the computation of standard deviation of each attribute (described in section 3.5). Then for every attribute, scaling is calculated using a function of standard deviation and frequency of attribute occurrences (described in section 3.6). This result is stored in the Numeric Scale Matrix (NSM). Now this result is normalized in a scale of 10 for every attribute and stored in Normalized Numeric Scale Matrix (NNSM) (described in section 3.7). The NNSM Matrix shows the dependency among all pair of attributes in the query set based on a numeric scale. Higher the value in each cell of NNSM lower the dependency among the pair of attributes corresponding to the particular cell. Thus the entry of 10 in some cell say, $[i, j]$ means that for i th attribute it has lowest dependency on j th attribute.

3.1 Algorithm *Numeric_Scale*

Begin

Step 1. The association between the queries and attributes is computed in Query Attribute Usage Matrix (QAUM).

Call method *QAUM_Computation*;

Step 2. Mutual dependencies of the attributes are stored in Attribute Dependency Matrix (ADM). The sum of 1 to n th columns (except the diagonal cell) for a given tuple is stored in the newly inserted $(m+1)^{th}$ column of ADM known as Total Measure.

Call method *ADM_Computation*;

Step 3. The probability that an attribute is dependent on another attribute is calculated and stored in a Probability Distribution Matrix (PDM).

Call method *Probability_Calculation*;

Step 4. Standard Deviation (SD) of each attribute is calculated.

Call method *StandardDeviation_Computation*;

Step 5. A particular attribute (PIVOT attribute) is selected and scaling of each attribute is done using the methodology Scaling_Calculation and the result is stored in Numeric Scale Matrix [NSM].

Call method Scaling_Calculation;

Step 6. Normalize the computed value of NSM in the closed interval of [1, 10] and stored in Normalized Numeric Scale Matrix[NNSM].

Call method Normalized_Scale;

End Numeric_Scale.

3.2 Method QAUM_Computation

In this stage, a $m \times n$ binary valued matrix is constructed named as Query Attribute Usage Matrix (QAUM). Here, m is the numbers of queries in the query set and n is the total numbers of attributes used in this query set. If query h uses k^{th} attribute, $QAUM[h, k]$ would be 1 else 0.

Begin QAUM_Computation

/ Procedure to build Query Attribute Usage Matrix (QAUM) */*

$\forall h \in [1..m], \forall k \in [1..n], \text{if } k \text{ is used in } h,$

$QAUM_{h,k} = 1;$

else

$QAUM_{h,k} = 0;$

End QAUM_Computation

3.3 Method ADM_Computation

In this stage, a $n \times n$ symmetric matrix named Attribute Dependency Matrix (ADM) is built. Each cell say $[h, k]$ of this matrix keeps a count on the number of times that both h^{th} and k^{th} attributes are used simultaneously in the set of m queries. As this is a symmetric matrix at this stage $ADM[h, k] = ADM[k, h]$. The diagonal of this matrix is marked as '#'. The diagonal cells contain trivial information that the dependency of an attribute is with itself only. After this new column is inserted into ADM named Total Measure, which stores the sum of every row. So, finally ADM is a $n \times (n+1)$ matrix.

Begin ADM_Computation

/ Procedure to count number of times two attributes a, b occur simultaneously and store it in matrix ADM and finally adding the values of each row to store in column Total Measure.*/*

$\forall h, k \in [1..n], \text{if } h = k$

$ADM_{h,k} = \#;$

else

*$ADM_{h,k} = \text{total count of occurrences of}$
*attributes h and k together in the set of N queries ;**

$\forall h, k \in [1..n],$

$ADM_{h,k+1} = \sum_{k=1}^n ADM_{h,k} \forall h \neq k.$

End ADM_Computation

3.4 Method Probability_Calculation

In this stage an $n \times n$ Probability Distribution Matrix (PDM) is constructed. This matrix is build to estimate a probabilistic measure of dependencies of every h^{th} attribute with other attributes. Every value of $PDM[h, k]$ is computed by dividing the value of $ADM[h, k]$ by the value of Total Measure ($ADM[h, n+1]$) corresponding to the h^{th} row of ADM. However, computing the measures of two types of cells are not required. These are diagonal cells and the cells for which ADM entry is 0. These types of cells are marked as '#' in PDM

Begin Probability_Calculation

/* Procedure to build Probability Distribution Matrix (PDM) on the basis of use of attributes */

$\forall h, k \in [1..n], \text{if } (h = k) \vee (ADM_{h,k} = 0),$

$PDM_{h,k} = \#;$

else

$PDM_{h,k} = \frac{ADM_{h,k}}{ADM_{h,n+1}};$

End Probability_Calculation

3.5 Method StandardDeviation _Computation

In this stage the mean, variance and standard deviation of attributes are computed as function of ADM and PDM. This is computed to measure the deviation of mean of other attribute from a given attribute.

$$\begin{aligned} \text{Mean}(\mu) &= \sum_{h=1}^n p_h \cdot x_h \\ \text{Variance}(X) &= \sum_{h=1}^n p_h \cdot (x_h - \mu)^2 \\ \text{Standard Deviation}(SD) &= \sqrt{\text{Variance}(X)} \end{aligned}$$

Fig. 1. Formulas for Mean, Variance and Standard Deviation

If the random variable X is discrete with probability mass function $x_1 \rightarrow p_1, \dots, x_n \rightarrow p_n$ then Mean, Variance and Standard Deviation(SD) are calculated using the three formulas shown in Figure. 1. Here, p_h and x_h are the entries of PDM and ADM respectively. However those entries which are marked as '#' in PDM they are not considered in this computation. The results of Mean, Variance and Standard Deviation of every attribute are stored in MVSD table. The 1st row contains the mean, 2nd row contains the variance and 3rd row contains the standard deviation.

Begin StandardDeviation_Computation

/* Procedure to compute and store mean, variance and standard deviation for attributes in MVSD matrix*/

$\forall h \in [1..n]$

$S = 0;$

$\forall k \in [1..n]$ if ($PDM_{h,k} \neq \#$)

$S = S + ADM_{h,k} \times PDM_{h,k};$ /*Value stored in ADM [h, k] is multiplied with the value stored in PDM[h,k] */

$MVSD_{1,h} = S;$ /* Stores mean(μ) */

$\forall h \in [1..n]$

$SD = 0;$

$\forall k \in [1..n]$ if ($PDM_{h,k} \neq \#$)

$SD = SD + PDM_{h,k} \times (ADM_{h,k} - MVSD_{1,h})^2;$

$MVSD_{2,h} = SD;$ /* Stores Variance(X) */

$MVSD_{3,h} = \sqrt{SD};$ /* Stores Standards Deviation */

End StandardDeviation_Computation**3.6 Method Scaling _Calculation**

In this stage an $n \times n$ matrix is constructed and named as Numeric Scale Matrix (NSM). The values of this matrix are computed as the function of standard deviation in MVSD and ADM. The result of every $MVSD[h, k]$ is computed as : modulus difference of standard deviation of h^{th} and k^{th} attribute, which is divided by the $ADM[h,k]$. However, if the $PDM[h, k]$ is #, it is not considered for computation. This matrix is constructed taking the help of both the probabilistic estimate of attribute usage as well as the current context of attribute usage. Thus this matrix identifies the degree of interdependence among every pair of attribute. Lower the value in every cell of NSM, higher the degree of dependence among the attributes corresponds to the row and column.

Begin Scaling _calculation

/* Procedure to compute and store degree of interdependence among attributes to build Numeric Scale Matrix (NSM)*/

$\forall h \in [1..m], \forall k \in [1..n], \text{ if } PDM_{h,k} = \#,$

$NSM_{h,k} = \#;$

else

$D = ADM_{h,k};$

$NSM_{h,k} = \frac{|MVSD_{3,h} - MVSD_{3,k}|}{D};$

End Scaling _Calculation**3.7 Method Normalized_Scale**

In this stage another $n \times n$ matrix named Normalized Numeric Scale Matrix (NNSM) is constructed by normalizing every row of NSM in a scale of 10. For every row the highest value is mapped to 10, similarly all other values of the row are mapped to the

new value with the same mapping function. For a row (say, for attribute h) if the attribute k has the value 10, that means h has the weakest relationship with attribute k where as the lowest entry (say, for attribute p) in some column signifies that h has the strongest relationship with attribute p.

Begin Normalized_Scale

/* Procedure to compute normalized numeric scale and result stored in NNSM*/

$\forall h \in [1..n]$

$Max = 0;$

$\forall k \in [1..n] \text{ if } (NSM_{h,k} \neq \#) \wedge (NSM_{h,k} > Max)$

$Max = NSM_{h,k};$

$\forall k \in [1..n] \text{ if } (NSM_{h,k} \neq \#)$

$NNSM_{h,k} = \#;$

 else

$NNSM_{h,k} = (NSM_{h,k}/Max) \times 10;$

End Normalized_Scale

4 Selection of Parameters and Complexity Analysis

The proposed model is based on set of queries and the attributes belonging to this set. Thus some selection criterions are important for successful execution of it. The different performance issues for the proposed numeric scale include scalability, dynamicity, and generalization aspect. The roles of the identified parameters are discussed below:

1) Query Selection: This algorithm starts with a set of queries. The entire analysis process is based on this query set. Hence, identification of query set is an important parameter for this process. It could be done in different ways. Two of the widely used methods are random selection, and interval based Selection. The first method extracts some of the executed queries from a given set randomly. In the second approach, certain time interval is chosen and the queries that have been executed during this are taken for analysis purpose.

2) Attribute Selection: Once the queries have been selected a set of attributes belonging to this query set is clearly identified. However, all of the attributes may not be subject of interest. As for example, if an attribute is used rarely in a query set, discarding that attribute would reduce the size of ADM and hence result in a faster execution of this method.

3) Threshold Selection: In the preceding step the requirement of attribute selection is defined. This is to be supported by some proper usage ratio. Thus the selection of threshold value of usage is also need to be defined.

The overall asymptotic run-time complexity of this algorithm is $O(n^2)$, where n is the number of attributes selected for analysis. Therefore, the effectiveness of our approach relies on one hand on the ratio n/M , where M is the overall number of attributes in the database, and on the other hand variance degree of attributes appearing in the query sequence. Measuring the actual computational advantage of our algorithm is the main subject of our ongoing work.

5 Illustrative Example

Let's consider a small example set of queries. This is only for the sake of a lucid explanation of the steps to be followed in the proposed algorithm. There are ten queries (q1, q2, ..., q10) in the set which use ten different attributes namely a1, a2, ..., a10. The queries are not given here due to space constraint, the example is shown starting from QAUM. The results are shown up to 2 decimal places.

Step 1: The use of these 10 attributes, by these 10 queries is shown in the QAUM (Table 1) using the method QAUM_Computation. If we consider query q1 we can say this query uses attributes a1, a2, a3, a4, a5 and a9.

Table 1. QAUM

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
q1	1	1	1	1	1	0	0	0	1	0
q2	1	0	0	1	0	1	1	0	0	0
q3	0	0	1	0	1	1	1	1	0	0
q4	1	0	0	0	1	1	0	0	1	1
q5	0	1	0	0	0	0	1	1	1	1
q6	0	0	1	1	0	1	0	0	0	1
q7	1	1	1	0	0	1	0	1	1	0
q8	1	1	1	0	1	0	0	0	0	1
q9	0	1	1	0	1	0	0	1	1	1
q10	1	0	0	1	0	1	1	0	1	1

Step 2: The mutual dependencies among all the attributes are stored in Attribute Dependency Matrix (ADM). For example the attributes a1 and a2 are used simultaneously in three queries, namely q1, q7 and q8. Thus, the entry in ADM for (a1, a2) is 3. The Total Measure is computed in ADM by adding the attribute dependency in every row. For instance, the Total Measure for a1 is 24. (Table 2)

Table 2. ADM

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	Total Measure
a1	#	3	3	3	2	3	2	1	4	3	24
a2	3	#	4	1	3	2	1	3	4	3	24
a3	3	4	#	2	4	3	2	3	3	2	26
a4	3	1	2	#	1	3	2	0	2	2	16
a5	3	2	4	1	#	2	1	2	3	3	21
a6	3	2	3	3	2	#	3	2	3	3	24
a7	2	1	2	2	1	3	#	2	2	2	17
a8	1	3	3	0	2	2	2	#	3	2	18
a9	4	4	3	2	3	3	2	3	#	4	28
a10	3	3	2	2	3	3	2	2	4	#	24

Step 3: PDM is built (Table 3) using the method Probability_Calculation to define the probabilistic estimate of attribute occurrence; e.g., (a1, a2) in PDM is computed by dividing ADM(a1,a2) with the Total Measure of a1 from ADM.

Table 3. PDM

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
a1	#	0.13	0.13	0.13	0.08	0.13	0.08	0.04	0.17	0.13
a2	0.13	#	0.17	0.04	0.13	0.08	0.04	0.13	0.17	0.13
a3	0.12	0.15	#	0.08	0.15	0.12	0.08	0.12	0.12	0.08
a4	0.19	0.06	0.13	#	0.06	0.19	0.13	#	0.13	0.13
a5	0.14	0.10	0.19	0.05	#	0.10	0.05	0.10	0.14	0.14
a6	0.13	0.08	0.13	0.13	0.08	#	0.13	0.08	0.13	0.13
a7	0.12	0.06	0.12	0.12	0.06	0.18	#	0.12	0.12	0.12
a8	0.06	0.17	0.17	0.00	0.11	0.11	0.11	#	0.17	0.11
a9	0.14	0.14	0.11	0.07	0.11	0.11	0.07	0.11	#	0.14
a10	0.13	0.13	0.08	0.08	0.13	0.13	0.08	0.08	0.17	#

Step 4: Using the method StandardDeviation_Computation Mean(μ), Variance, standard deviation for all the attributes are computed and stored in table MVSD (Table 4). It has three data rows and n columns for the attributes. The first row of the table contains the mean, the second row contains variance and the third row contains the standard deviation (SD). The formulations for these three counts are specified in Fig. 1.

Table 4. MVSD

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
Mean	2.92	3.08	3.08	2.25	2.71	2.75	2.06	2.44	3.29	2.83
Variance	0.30	2.09	2.61	5.54	3.95	3.82	3.46	3.41	6.98	7.38
SD	0.55	1.45	1.62	2.35	1.99	1.95	1.86	1.85	2.64	2.72

Table 5. NSM

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
a1	#	0.30	0.36	0.60	0.72	0.47	0.66	1.30	0.52	0.72
a2	0.30	#	0.04	0.90	0.18	0.25	0.41	0.13	0.30	0.42
a3	0.36	0.04	#	0.37	0.09	0.11	0.12	0.08	0.34	0.55
a4	0.60	0.90	0.37	#	0.36	0.13	0.25	#	0.15	0.19
a5	0.48	0.27	0.09	0.36	#	0.02	0.13	0.07	0.22	0.24
a6	0.47	0.25	0.11	0.13	0.02	#	0.03	0.05	0.23	0.26
a7	0.66	0.41	0.12	0.25	0.13	0.03	#	0.01	0.39	0.43
a8	1.30	0.13	0.08	0.00	0.07	0.05	0.01	#	0.26	0.44
a9	0.52	0.30	0.34	0.15	0.22	0.23	0.39	0.26	#	0.02
a10	0.72	0.42	0.55	0.19	0.24	0.26	0.43	0.44	0.02	#

Step 5: Using the method Scaling_Calculation, the NSM table (Table 5) is constructed. The entries in NSM are derived from the corresponding entries in ADM and MVSD. As for example $NSM(a1, a2)$ is computed at first by taking the modulo subtraction result of standard deviation of $a1$ and $a2$. Then this result is divided by $ADM(a1, a2)$. In this case, the difference from the modulo subtraction is 0.90 and $ADM(a1, a2)$ is 3. Thus the $NSM(a1, a2)$ is 0.30.

Table 6. NNSM

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
a1	#	2.31	2.74	4.62	5.54	3.59	5.04	10	4.02	5.56
a2	3.33	#	0.47	10	2.00	2.78	4.56	1.48	3.31	4.70
a3	6.48	0.77	#	6.64	1.68	2.00	2.18	1.39	6.18	10
a4	6.67	10	4.06	#	4.00	1.48	2.72	#	1.61	2.06
a5	10	5.63	1.93	7.50	#	0.42	2.71	1.46	4.51	5.07
a6	10	5.36	2.36	2.86	0.43	#	0.64	1.07	4.93	5.50
a7	10	6.26	1.83	3.74	1.98	0.46	#	0.08	5.95	6.56
a8	10	1.03	0.59	#	0.54	0.38	0.04	#	2.03	3.35
a9	10	5.69	6.51	2.78	4.15	4.40	7.46	5.04	#	0.38
a10	10	5.85	7.60	2.56	3.36	3.55	5.94	6.01	0.28	#

Step 6: Using the method Normalized_Scale, every row of NSM is scaled in a factor of 10. For every row the highest value of NSM is scaled to 10 and similarly all other attributes are mapped to new values in NNSM (Table 6). As for example in the 1st row of NSM $a8$ has the maximum value thus it is scaled to 10 using the algorithm described in section 3.7. Similarly all other values of 1st row are mapped.

6 Conclusion

The novelty of this paper is in proposing a methodology to build a numeric scale based on quantitative analysis on the set of attributes used in recent queries. Use of the standard deviation in this methodology helps to build a predictive model on future usage of attributes. Thus this method combines the actual usage with the probabilistic assumptions.

The proposed scale would find significant usage in diverse aspects of database management. This would improve the performance towards creation and maintenance of the materialized views. This in turn would enhance the query execution in both database and data warehouse applications. As the proposed scale is independent of any external parameters, materialized views could be formed for heterogeneous applications. Other database functionalities like indexing, cluster formation, etc. could also be done on the basis of quantitative measures using the proposed scale as compared to intuitive approaches. The proposed scale is also useful in any rank based analysis of attributes. The future research work of this scale includes several aspects. Firstly, the types of

queries to be selected to initiate this process for different applications are interesting and depend on the business logic. Experimental findings on diverse database applications by using the proposed scale could unearth interesting associations. Secondly, incorporating value based analysis over the attributes based analysis could be one using the scale. As all the values of the attributes are not accessed during query processing filtering could be used on the values as well. Combining the value based analysis with the existing numeric scale would help to achieve high speed query processing. Besides, the proposed scale could be combined with the concept of abstraction of attributes using concept hierarchy. This would help to reduce the amount of data to be accessed and to reduce size of materialized views.

References

1. Mukkamala, R.: Improving database performance through query standardization. In: IEEE Proceedings of Energy and Information Technologies in the Southeast, Southeastcon 1989 (1989)
2. Chiang, L., Chi Sheng, S., Chen Huei, Y.: Optimizing large join queries using a graph-based approach. IEEE Transactions on Knowledge and Data Engineering (March/April 2001)
3. Chin Wang, J., Tzong Horng, J., Ming Hsu, Y., Jhinue Liu, B.: A genetic algorithm for set query optimization in distributed database systems. In: IEEE International Conferences on Systems, Man, and Cybernetic (1996)
4. Bruno, N., Galindo-Legaria, C., Joshi, M.: Polynomial heuristics for query optimization. In: 26th IEEE International Conferences on Data Engineering (ICDE 2010) (2010)
5. Sarathy, V.M., Saxton, L.V., Van Gucht, D.: Algebraic foundation and optimization for object based query languages. In: Proceedings of 9th International Conference on Data Engineering (1993)
6. Ordonez, C.: Optimization of Linear Recursive Queries in SQL. IEEE Transactions on Knowledge and Data Engineering (2010)
7. Zhuang, Y., Qing, L., Chen, L.: Multi-query Optimization for Distributed Similarity Query Processing. In: 28th International Conference on Distributed Computing Systems, ICDCS 2008 (2008)
8. Yang, J., Karlapalem, K., Li, Q.: A framework for designing materialized views in data warehousing environment. In: Proceedings of 17th IEEE International Conference on Distributed Computing Systems, Maryland, U.S.A. (May 1997)
9. Gupta, H.: Selection of Views to Materialize in a DataWarehouse. In: Afrati, F.N., Kolaitis, P.G. (eds.) ICDT 1997. LNCS, vol. 1186, pp. 98–112. Springer, Heidelberg (1996)
10. Ashadevi, B., Subramanian, R.: Optimized Cost Effective Approach for Selection of Materialized views in Data Warehousing. International Journal of Computer Science and Technology 9(1) (April 2009)
11. Bhagat, P.A., Harle, R.B.: Materialized view management in peer to peer environment. In: International Conference and Workshop on Emerging Trends in Technology, ICWET 2011 (2011)
12. Goyal, N., Zaveri, K.S., Sharma, Y.: Improved Bitmap Indexing Strategy for Data Warehouses. In: Proceedings of 9th International Conference on Information Technology, ICIT 2006 (2006)
13. Aouiche, K., Darmont, J.: Data Mining Based Materialized View and Index Selection in Data Warehouses. Proceedings of J. Intell. Inf. Syst. (2009)

Left-Right Oscillate Algorithm for Community Detection Used in E-Learning System

Jan Martinovič², Pavla Dráždilová¹, Kateřina Slaninová¹,
Tomáš Kocyan², and Václav Snášel²

¹ VŠB - Technical University of Ostrava,
Faculty of Electrical Engineering and Computer Science,
17. Listopadu 15/2172, 708 33 Ostrava, Czech Republic
{katerina.slaninova,pavla.drazdilova}@vsb.cz

² VŠB - Technical University of Ostrava,
IT4Innovations,
17. Listopadu 15/2172, 708 33 Ostrava, Czech Republic
{jan.martinovic,tomas.kocyan,vaclav.snasel}@vsb.cz

Abstract. Learning management systems are widely used as a support of distance learning. Recently, these systems successfully help in present education as well. Learning management systems store large amount of data based on the history of users' interactions with the system. Obtained information is commonly used for further course optimization, finding e-tutors in collaboration learning, analysis of students' behavior, or for other purposes. The partial goal of the paper is an analysis of students' behavior in a learning management system. Students' behavior is defined using selected methods from sequential and process mining with the focus to the reduction of large amount of extracted sequences. The main goal of the paper is description of our Left-Right Oscillate algorithm for community detection. The usage of this algorithm is presented on the extracted sequences from the learning management system. The core of this work is based on spectral ordering. Spectral ordering is the first part of an algorithm used to seek out communities within selected, evaluated networks. More precise designations for communities are then monitored using modularity.

1 Introduction

E-learning is a method of education which utilizes a wide spectrum of technologies, mainly internet or computer-based, in the learning process. It is naturally related to distance learning, but nowadays is commonly used to support face-to-face learning as well. *Learning management systems* (LMS) provide effective maintenance of particular courses and facilitate communication within the student community and between educators and students [9]. Such systems usually support the distribution of study materials to students, content building of courses, preparation of quizzes and assignments, discussions, or distance management of classes. In addition, these systems provide a number of collaborative learning tools such as forums, chats, news, file storage etc.

Regardless of LMS benefits, huge amount of recorded data in large collections makes often too difficult to manage them and to extract useful information from them. To

overcome this problem, some LMS offer basic reporting tools. However, in such large amount of information the outputs become quite obscure and unclear. In addition, they do not provide specific information of student activities while evaluating the structure and content of the courses and its effectiveness for the learning process [26]. The most effective solution to this problem is to use data mining techniques [1].

The main goal of the paper is the description of our Left-Right Oscillate algorithm for community detection. The usage of this algorithm is presented on the extracted sequences from a learning management system. The core of this work is based on spectral ordering. Spectral ordering is the first part of an algorithm used to seek out communities within selected, evaluated networks. More precise designations for communities are then monitored using modularity.

The discovery and analysis of community structure in networks is a topic of considerable recent interest in sociology, physics, biology and other fields. Networks are very useful as a foundation for the mathematical representation of a variety of complex systems such as biological and social systems, the Internet, the world wide web, and many others [8][17]. A common feature of many networks is community structure, the tendency for vertices to divide into groups, with dense connections within groups and only sparser connections between them [12][18].

2 Analysis of Students' Behavior

Several authors published contributions with relation to mining data from e-learning systems to extract knowledge that describe students' behavior. Among others we can mention for example [14], where authors investigated learning process of students by the analysis of web log files. A 'learnograms' were used to visualize students' behavior in this publication. Chen et al. [3] used fuzzy clustering to analyze e-learning behavior of students. El-Hales [11] used association rule mining, classification using decision trees, E-M clustering and outlier detection to describe students' behavior. Yang et al. [25] presented a framework for visualization of learning historical data, learning patterns and learning status of students using association rules mining. The agent technology and statistical analysis methods were applied on student e-learning behavior to evaluate findings within the context of behavior theory and behavioral science in [2].

Our subject of interest in this paper is student behavior in LMS, which is recorded in form of events and stored in the logs. Thus, we can define the student behavior with the terms of process mining which are used commonly in business sphere. Aalst et al. [23][22] defines event log as follows:

Let A be a set of activities (also referred as tasks) and U as set of performers (resources, persons). $E = A \times U$ is the set of (possible) events (combinations of an activity and performer). For a given set A , A^* is the set of all finite sequences over A . A finite sequence over A of length n is mapping $\sigma = \langle a_1, a_2, \dots, a_n \rangle$, where $a_i = \sigma(i)$ for $1 \leq i \leq n$. $C = E^*$ is the set of possible event sequences. A simple event log is a multiset of traces over A .

The behavioral patterns are discovered using similarity of extracted sequences of activities performed in the system.

A sequence is an ordered list of elements, denoted $\langle e_1, e_2, \dots, e_l \rangle$. Given two sequences $\alpha = \langle a_1, a_2, \dots, a_n \rangle$ and $\beta = \langle b_1, b_2, \dots, b_m \rangle$. α is called a subsequence of β , denoted as $\alpha \subseteq \beta$, if there exist integers $1 \leq j_1 < j_2 < \dots < j_n \leq m$ such that $a_1 = b_{j_1}, a_2 = b_{j_2}, \dots, a_n = b_{j_n}$. β is then a super sequence of α .

For finding the behavioral patterns, we need to use the methods for the sequence comparison. There are generally known several methods for the comparison of two or more categorical sequences. On the basis of our previous work [21] we have selected the algorithm, which deals with the different lengths of sequences and with the possible error or distortion inside the sequence.

Time-warped longest common subsequence (T-WLCS) [13] is the method, which combines the advantages of two methods from pattern mining - The longest common subsequence (LCSS) [15] and Dynamic time warping (DTW) [16]. LCSS allows us to find the longest common subsequence of two compared sequences. DTW is used for finding the optimal visualization of elements in two sequences to match them as much as possible. Then, selected T-WLCS method is able to compare sequences of various lengths, it takes into consideration the order of elements in the sequences, and it is immune to minor distortions inside of one of the compared sequences. Moreover, the method emphasizes recurrence of the elements in one of the compared sequences.

To obtain behavioral patterns of similar sequences in LMS, we have used our proposed new Left-Right Oscillate algorithm for community detection. The algorithm is based on spectral ordering (see Section 3).

3 Spectral Clustering and Ordering

Spectral clustering has become one of the most popular modern clustering algorithms in recent years. It is one of the graph theoretical clustering techniques and is simple to implement, can be solved efficiently by standard linear algebra methods, and very often outperforms traditional clustering algorithms such as the k-means or single linkage (hierarchical clustering). A comprehensive introduction to the mathematics involved in spectral graph theory is the textbook of Chung [5]. Spectral clustering algorithm uses eigenvalues and eigenvectors of Laplacian of similarity matrix derived from the data set to find the clusters. A practical implementation of the clustering algorithm is presented in [4]. Recursive spectral clustering algorithm is used in [6]. There Dasgupta et al. analyzed the second eigenvector technique of spectral partitioning on the planted partition random graph model, by constructing a recursive algorithm. A spectral clustering approach to finding communities in graphs was applied in [24].

Ding and He showed in [7] that a linear ordering based on a distance sensitive objective has a continuous solution which is the eigenvector of the Laplacian. Their solution demonstrates close relationship between clustering and ordering. They proposed direct K-way cluster assignment method which transforms the problem to linearization of the clustering assignment problem. The linearized assignment algorithm depends crucially on an algorithm for ordering objects based on pairwise similarity metric. The ordering is such that adjacent objects are similar while objects far away along the ordering are dissimilar. They showed that for such an ordering objective function the inverse index

permutation has a continuous (relaxed) solution which is the eigenvector of the Laplacian of the similarity matrix.

3.1 Modularity-Quality of Detected Communities

To quantify the quality of the subdivisions we can use modularity [20], defined as the fraction of links between the nodes in the same community minus their expected value in a corresponding random graph [20]. Networks with the high modularity have dense connections between the nodes within community, but sparse connections between the nodes in the different communities. Modularity is often used in optimization methods for detecting community structure in the networks [19]. The value of the modularity lies in the range $\langle -0.5, 1 \rangle$. It is positive, if the number of edges within groups exceeds the number expected on the basis of chance.

For a *weighted graph* G we have a weight function $w : E \rightarrow R$. It is for example function of the similarity between the nodes v_i and v_j . The weighted adjacency matrix of the graph is the matrix $W = (w_{ij})$ $i, j = 1, \dots, n$. Than the degree of a vertex $v_i \in V$ in weighted graph is defined as

$$d_i = \sum_{j=1}^n w_{ij}.$$

The weighted degree matrix D is defined as the diagonal matrix with the weighted degrees d_1, \dots, d_n on the diagonal.

In terms of the edge weights, modularity $Q(C_1, \dots, C_k)$ is defined over a specific clustering into k known clusters C_1, \dots, C_k as

$$Q(C_1, \dots, C_k) = \sum_{i=1}^k (e_{ii} - \sum_{j=1, i \neq j}^k e_{ij})$$

where $e_{ij} = \sum_{(u,v) \in E, u \in C_i, v \in C_j} w(u, v)$ with each edge $(u, v) \in E$ included at most once in the computation.

4 Left-Right Oscillate Algorithm for Community Detection

Upon completing our study of various modifications of algorithms for spectral clustering, we designed our own algorithm for detecting communities within complex networks. This algorithm utilizes spectral ordering where similar vertices are closer to indexes and less similar vertices are further from indexes. When determining the ordering, it is necessary to calculate the eigenvector of the second smallest eigenvalue of the matrix $L = D - W$. Since we have designed our algorithm for large amounts of data in a complex network, we used Lanczos method to calculate the Fiedler vector. Once the Fiedler vector was calculated, we detected appropriate gaps that divide the vertices of a graph into communities. As observed in the experiment, this type of separation into gaps leads to several badly-assigned subgraphs. This is due to the fact that the Fiedler vector is only linear ordered, as is revealed in our data collection. The Left-Right algorithm (see Algorithm 3.1) we have designed for incorporating small subgraphs into larger communities, gradually increases modularity in a given calculation.

Algorithm 1. Left-Right Algorithm for Community Detection

Input: similarity matrix $W = w_{i,j}$ for $i = 1, \dots, n$ and S_c size of smallest communities.

Output: communities C_k , modularity of detected communities

1. Create Laplacian $L = D - W$ using a matrix of similarity W of a connected graph $G = (V, E, W)$.
2. Calculate the Fiedler vector (the second eigenvector of Laplacian).
3. Reorder vertices according to Fiedler vector.
4. Calculate the sums of antidiagonals $Asum_i = \sum_j w_{i-j, i+j}$ a $Asum_{(i+1/2)} = \sum_j w_{i-j, i+j+1}$ for all $i = 1, \dots, n$ and determine $sum_i = Asum(i - 1/2)/4 + Asum(i)/2 + Asum(i + 1/2)/4$.
5. Approximate the discrete function sum_i by its spline and determine its first and second derivation. Then find all local minimums and maximums.
6. Assign maximum gaps that lie between two local maximums. Divide the set of vertices according to its gaps. Obtain subsets $SS_k \subset V$, where $k = 1, \dots, K$ is the amount of subsets.
7. Detect a community using the Left-Right Oscillate assigning algorithm (see Algorithm 2).

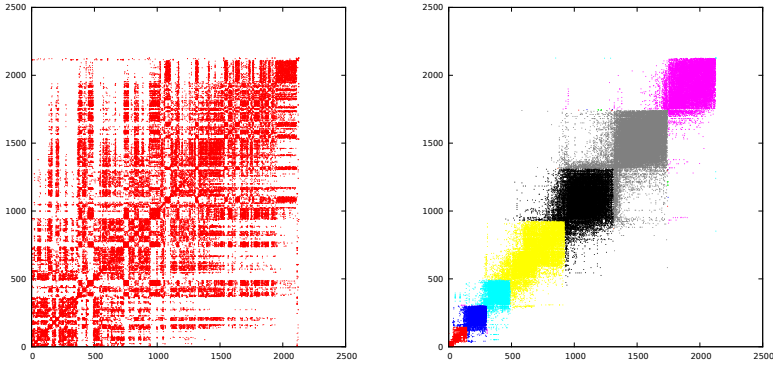


Fig. 1. Similarity Matrix and Permuted Similarity Matrix (Natural Number of Communities is 7)


Spectral ordering minimizes the sum of weighted edges multiplied to the power of the difference in index nodes with the edge incidence. The calculation used for this equation is the given eigenvector of the second smallest eigenvalue (Fiedler vector) matrix $L = D - W$. A visualized Fiedler vector and ordered matrix similarity (in agreement with the Fiedler vector) reveals the creation of several natural clusters which is assigned by our algorithm.

For finding the Fiedler vector of Laplacian above a large, sparse and symmetric matrix representative of the evaluated network, we used Lanczos method to partially solve the eigenvalue problem. To determine the dimension of Krylov subspaces (for a more precisely calculated Lanczos method), we used modularity for determining the quality of a detected community. In [7], there is an example of a symmetric Laplacian $L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$. Our experiment revealed, that this solution is only appropriate for some networks.

Algorithm 2. Left-Right-Oscillate Assigned

Input: subsets $SS_k \subset V$, where $k = 1, \dots, K$, S_c size of smallest communities.

Output: communities C_k , modularity of detected partitioning.

1. Find connected components C_j for subset SS_1 , which are greater than the selected size $|C_j| \geq S_c$. These components create communities. We add the rest of the vertices $v_i \in SS_1 - \bigcup C_j$ to the next subset of vertices SS_2 .
 2. Find next connected components C_j for every subset SS_k $k = 2, \dots, K$, which are greater than the selected size $|C_j| \geq S_c$. These components create communities. Attempt to assign other vertices to the previous community, which was established in the previous step. If the vertex has no edge leading to the previous community then we add the vertex to the next subset of vertices SS_{k+1} . Continue repeating this method  until reach the end of ordered vertices.
 3. Going through through all subsets of vertices, connected components are assigned to C_j for $j = 1, \dots, J - 1$ and C_J contain a set of connected components smaller than the selected size.
 4. Employ the same approach going right-left without "oscillation". Begin with $C_{J-1} = C_{J-1} \cup C_J$.
-

The next step for the algorithm is to order indexes of vertices $v_i \in V$ for all $i = 1, \dots, n$ in compliance with ordering using Fiedler vector values. Because we want to find communities that are easily detected in a visual representation when ordered by a similarity matrix, we must determine where one community in a linear order ends and the next begins (find two nodes that belong to various communities). For this reason, we have calculated the value of antidiagonal sums above an ordered the set of vertices that capture a cluster overlap in neighboring vertices v_i . We define cluster crossing as the sum of a small fraction of the pairwise similarities. This is aided by linear ordering data points. The goal is to find the nodes that lay in the areas with fewer edges. These vertices lie close to locales with minimum function that are attached by approximation of a cluster overlap discrete function Sum_i . We assigned this approximation using the spline function, allowing for easy calculations of both the first and second derivation, which are used to assign local extremes. Between the two local maximum extremes of this function, there lie two vertices. In this area, these vertices represent a maximum gap (the difference in their Fiedler vector value). This gap determines the border between two potential communities.

Using this method, we have found the natural amount of 'communities' above a given evaluated network. Since the precision with which the Fiedler vector is calculated is a determining factor, and since in some cases vertices are incorrectly assigned, the result is an irrelevant component. The benefit of using our algorithm lies within its ability to assign isolated nodes (or very small subgraphs with selected sizes) to the nearest, most suitable, connected component that creates the nucleus of a future community. Within our assignments, we gradually arrive at a set of vertices V separated by gaps in individual subsets V_k . If the found set V_k does not create a connected subgraph ($G_k = (V_k, E)$), we determine all connected components in this subgraph. The maximum connected subgraph then creates the nucleus of this community and all subgraphs smaller than the selected size are moved to the right. We then attempt to reassign the subgraph to the

next subset of vertices V_{k+1} . Due to the linear nature of spectral ordering, it is presumable that subgraphs not yet assigned are reordered to the next subset of vertices. This means that we add the vertices of these subgraphs to the vertices of the next subset (that came into existence along gaps and creates a subgraph of the original graph with a set of vertices V_{k+1}). Then we test the connectivity of subgraph G_{k+1} , which was expanded by the nodes from the previous, unassigned subgraph. We go through the entire, spectrally ordered set of graph vertices employing this method. At the end of this process, we have created the most relevant of components within which we assign small subgraphs that are not yet assigned. Then, we repeat this approach in the opposite direction - going from right to left - and we try to add vertices for inspection in a subgraph. We may then assign the vertices to a connected subgraph with adjacency to a vertex of a given subgraph.

Once we assign a subset of vertices using gaps, and once we have detected connected components from left to right and vice versa, we always calculate the modularity for the obtained separation of graphs into subgraphs. Our results have revealed that our Left-right method increases modularity. The resulting connected subgraphs then create the structure of communities in the graph, which is demonstrated on well known data collection Zachary karate club in Table 1.

Table 1. Modularity Before and After Left-Right Algorithm for Zachary Karate Club

	Before Left-right	Commun.	After Left	Commun.	After Left-right	Commun.
Laplacian	0.272	11	0.361	4	0.361	4
Normalized-cut	0.342	7	0.311	4	0.311	4

5 Sequence Extraction in LMS Moodle

In this section is presented the extraction of students' behavioral patterns performed in the e-learning educational process. The analyzed data collections were stored in the Learning Management System (LMS) Moodle logs used to support e-learning education at Silesian University, Czech Republic.

The logs consist of records of all events performed by Moodle users, such as communication in forums and chats, reading study materials or blogs, taking tests or quizzes etc. The users of this system are students, tutors, and administrators; the experiment was limited to the events performed only by students.

Let us define a set of students (users) U , set of courses C and term *Activity* $a_k \in A$, where $A = P \times B$ is a combination of activity prefix $p_m \in P$ (e.g. course view, resource view, blog view, quiz attempt) and an action $b_n \in B$, which describes detailed information of an activity prefix (concrete downloaded or viewed material, concrete test etc.). *Event* $e_j \in E$ then represents the activity performed by certain student $u_i \in U$ in LMS. On the basis of this definition, we have created a set S_i of sequences s_{ij} for the user u_i , which represents the students' (users') paths (sessions) on the LMS website. *Sequence* s_{ij} is defined as a sequence of activities, for example $s_{ij} = \langle a_{1j}, a_{2j}, \dots, a_{qj} \rangle$, which is j -th sequence of the user u_i .

The sequences were extracted likewise the user sessions on the web; the end of the sequences was identified by at least 30 minutes of inactivity, which is based on our previous experiments [10]. Similar conclusion was presented by Zorrilla et al. in [26].

Using this method, we have obtained a set of all sequences $S = \cup_{\forall i} S_i$, which consisted of large amount of different sequences s_i performed in LMS Moodle. We have selected the course Microeconomy A as an example for the demonstration of proposed method. In Table 2 is presented detailed information about the selected course.

Table 2. Description of Log File for Course Microeconomy A

Records	Students	Prefixes	Actions	Sequences
65 012	807	67	951	8 854

Table 3. Description of Sequence Graphs for T-WLCS Method

T-WLCS				
θ	Isolated Nodes	Edges	Avg. Degree	Avg. Weighted Degree
0.1	31	5577366	944.036	179.431
0.2	143	1739042	294.354	88.883
0.3	606	534648	90.496	38.735
0.4	1200	271826	46.010	22.998
0.5	2465	103028	17.439	10.430
0.6	3781	29298	4.959	3.596
0.7	5038	8080	1.368	1.269
0.8	5517	5914	1.001	0.997
0.9	5568	5788	0.980	0.980

The obtained set S of sequences consisted of large amount of different sequences, often very similar. Such large amount of information is hard to clearly visualize and to present in well arranged way. Moreover, the comparison of users based on their behavior is computationally expensive with such dimension. Therefore, we present in the article [21] the identification of significant behavioral patterns based on the sequence similarity, which allows us to reduce amount of extracted sequences.

We have used T-WLCS methods for the similarity measurement of sequences. The T-WLCS find the longest common subsequence α of compared sequences β_x and β_y , where $\alpha \subseteq \beta_x \wedge \alpha \subseteq \beta_y$, with relation to T-WLCS. Similarity was counted by the Equation 1

$$Sim(\beta_x, \beta_y) = \frac{(l(\alpha) * h)^2}{l(\beta_x) * l(\beta_y)}, \quad (1)$$

where $l(\alpha)$ is a length of the longest common subsequence α for sequences β_x and β_y ; $l(\beta_x)$ and $l(\beta_y)$ are analogically lengths of compared sequences β_x and β_y , and

$$h = \frac{Min(l(\beta_x), l(\beta_y))}{Max(l(\beta_x), l(\beta_y))} \quad (2)$$

On the basis of selected T-WLCS method for finding the similarity of sequences, we have constructed the similarity matrix for sequences ($|S| \times |S|$) which can be represented using tools of graph theory. For the visualization of network was constructed weighted graph $G(V, E)$, where weight w is defined as function $w : E(G) \rightarrow R$, when $w(e) > 0$. Set V is represented by set of sequences S , weights w are evaluated by the similarity of sequences, see Equation 1 depending on selected method. In Table 3 is more detailed description of weighted graphs of sequences, where weight is defined by T-WLCS method for selected threshold θ (threshold for edges filtering - edges with smaller weights are removed). The number of nodes for each graph is 5908.

5.1 Reduction of Large Amount of Sequences by Left-Right Oscillate Algorithm

We described the procedure for extraction of sequences from the LMS system in previous parts of the paper. We created the graphs of sequences by T-WLCS method. The examples in this section show how the graphs of sequences are divided to clusters by Left-Right Oscillate algorithm. We will use the concept "clusters" instead of "communities" in this part of the paper because we used Left-Right Oscillate algorithm for finding clusters of sequences.

Table 4. Description of Selected Biggest Clusters of Sequence Graphs for T-WLCS Method

T-WLCS		
θ	Nodes	Edges
0.2	5763	1739040
0.4	4639	271732
0.7	142	1030

Table 5. Partitioning of Sequence Graph for $\theta \leq 0.2$

$S_c = 1$					
Modularity Type	Modularity	Clusters	Size of 1th	Size of 2nd	Size of 3rd
Original	0.23471	85	560	365	104
Left-Right	0.23471	85	560	365	104
$S_c = 3$					
Modularity Type	Modularity	Clusters	Size of 1th	Size of 2nd	Size of 3rd
Original	0.23471	85	560	365	104
Left-Right	0.23721	7	581	421	104
$S_c = 6$					
Modularity Type	Modularity	Clusters	Size of 1th	Size of 2nd	Size of 3rd
Original	0.23471	85	560	365	104
Left-Right	0.23717	5	580	424	104

Table 6. Partitioning of Sequence Graph for $\theta \leq 0.4$

$S_c = 1$					
Modularity Type	Modularity	Clusters	Size of 1th	Size of 2nd	Size of 3rd
Original	0.52153	656	829	648	497
Left-Right	0.52153	656	829	648	497
$S_c = 3$					
Modularity Type	Modularity	Clusters	Size of 1th	Size of 2nd	Size of 3rd
Original	0.52153	656	829	648	497
Left-Right	0.52820	101	956	720	579
$S_c = 6$					
Modularity Type	Modularity	Clusters	Size of 1th	Size of 2nd	Size of 3rd
Original	0.52153	656	829	648	497
Left-Right	0.52955	59	962	739	594
$S_c = 10$					
Modularity Type	Modularity	Clusters	Size of 1th	Size of 2nd	Size of 3rd
Original	0.52153	656	829	648	497
Left-Right	0.52890	40	952	794	586

Table 7. Partitioning of Sequence Graph for $\theta \leq 0.7$

$S_c = 1$					
Modularity Type	Modularity	Clusters	Size of 1th	Size of 2nd	Size of 3rd
Original	0.26457	63	15	15	14
Left-Right	0.26457	63	15	15	14
$S_c = 3$					
Modularity Type	Modularity	Clusters	Size of 1th	Size of 2nd	Size of 3rd
Original	0.26457	63	15	15	14
Left-Right	0.38731	12	23	22	21
$S_c = 6$					
Modularity Type	Modularity	Clusters	Size of 1th	Size of 2nd	Size of 3rd
Original	0.26457	63	15	15	14
Left-Right	0.46304	7	45	15	14
$S_c = 10$					
Modularity Type	Modularity	Clusters	Size of 1th	Size of 2nd	Size of 3rd
Original	0.26457	63	15	15	14
Left-Right	0.45416	4	58	42	27

For the illustration, we have selected three different graphs from Table 3. These are the graphs created with parameter θ is greater than 0.2, 0.4 and 0.7. In each graph, there was identified the largest connected component, and on the basis on this component was created the new graph. This graph was partitioned by our new Left-Right Oscillate algorithm (maximum cycles of Lanczos algorithm inside Left-Right Oscillate algorithm was set to 1500). The sizes of these newly generated graphs are presented in Table 4.

Individual outputs and quality cuts of graphs after Left-Right Oscillate algorithm can be seen in Table 5, Table 6 and Table 7. In these tables we have column "Modularity Type" where row "Original" is without applied Left-Right Algorithm and "Left-Right" is row with information after Left-Right algorithm. Other columns in these tables are "Modularity" (see section 3.1), "Clusters" with amount of clusters after partitioning and columns with sizes of the top three communities (columns "Size of 1th", "Size of 2th", "Size of 3th").

It is apparent that the modularity is improved, if the parameter S_c (size of smallest clusters - see Algorithm 3.1) of the Left-Right Oscillate algorithm is greater than 1.

6 Conclusion

In the paper we introduced the Left-Right Oscillate algorithm, which allows us to improve the results of community detection based on spectral ordering. We showed effect of parameter S_c on the quality of clustering of sequences, which were extracted from the Moodle e-learning system. This allows us to better identify the same behavior of students in the online e-learning system. Modularity was used for measuring the quality of the distribution of sequences within clusters. In the future work we want to consider using the Left-Right Oscillate algorithm for hierarchical graph of sequences partitioning. Thanks to this we want to aim a more appropriate division of student's behavioral patterns.

Acknowledgment. This work was partially supported by SGS, VSB – Technical University of Ostrava, Czech Republic, under the grant No. SP2012/151 Large graph analysis and processing and by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

References

1. Castro, F., Vellido, A., Nebot, A., Mugica, F.: Applying Data Mining Techniques to e-Learning Problems. In: Jain, L., Tedman, R., Tedman, D. (eds.) *Evolution of Teaching and Learning Paradigms in Intelligent Environment*. SCI, vol. 62, pp. 183–221. Springer, Heidelberg (2007)
2. Chen, B., Shen, C., Ma, G., Zhang, Y., Zhou, Y.: The evaluation and analysis of student e-learning behaviour. In: *IEEE/ACIS 10th International Conference on Computer and Information Science (ICIS)*, pp. 244–248 (2011)
3. Chen, J., Huang, K., Wang, F., Wang, H.: E-learning behavior analysis based on fuzzy clustering. In: *Proceedings of International Conference on Genetic and Evolutionary Computing* (2009)
4. Cheng, D., Kannan, R., Vempala, S., Wang, G.: On a recursive spectral algorithm for clustering from pairwise similarities. Technical report, MIT (2003)
5. Chung, F.R.K.: *Spectral Graph Theory*, vol. 92. American Mathematical Society (1997)
6. Dasgupta, A., Hopcroft, J., Kannan, R., Mitra, P.: Spectral Clustering by Recursive Partitioning. In: Azar, Y., Erlebach, T. (eds.) *ESA 2006*. LNCS, vol. 4168, pp. 256–267. Springer, Heidelberg (2006)
7. Ding, C., He, X.: Linearized cluster assignment via spectral ordering. In: *Twentyfirst International Conference on Machine Learning, ICML 2004*, vol. 21, p. 30 (2004)

8. Dorogovtsev, S.N., Mendes, J.F.F.: *Evolution of Networks: From Biological Nets to the Internet and WWW*, vol. 57. Oxford University Press (2003)
9. Dráždilová, P., Obadi, G., Slaninová, K., Al-Dubaei, S., Martinovič, J., Snášel, V.: Computational Intelligence Methods for Data Analysis and Mining of eLearning Activities. In: Khafa, F., Caballé, S., Abraham, A., Daradoumis, T., Juan Perez, A.A. (eds.) *Computational Intelligence for Tech. Enhanced Learning. SCI*, vol. 273, pp. 195–224. Springer, Heidelberg (2010)
10. Dráždilová, P., Slaninová, K., Martinovič, J., Obadi, G., Snášel, V.: Creation of students' activities from learning management system and their analysis. In: Abraham, A., Snášel, V., Wegrzyn-Wolska, K. (eds.) *IEEE Proceedings of International Conference on Computational Aspects of Social Networks, CASON 2009*, pp. 155–160 (2009)
11. El-halees, A.: Mining students data to analyze learning behavior: a case study (2008)
12. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America* 99(12), 7821–7826 (2002)
13. Guo, A., Siegelmann, H.: Time-Warped Longest Common Subsequence Algorithm for Music Retrieval, pp. 258–261. Universitat Pompeu Fabra (2004)
14. Hershkovitz, A., Nachmias, R.: Learning about online learning processes and students' motivation through web usage mining. *Interdisciplinary Journal of E-Learning and Learning Objects* 5, 197–214 (2009)
15. Hirschberg, D.S.: Algorithms for the longest common subsequence problem. *J. ACM* 24, 664–675 (1977)
16. Müller, M.: *Information Retrieval for Music and Motion*. Springer (2007)
17. Newman, M.E.J., Barabási, A.-L., Watts, D.J.: *The structure and dynamics of networks*, vol. 107. Princeton University Press (2006)
18. Newman, M.E.J.: Detecting community structure in networks. *The European Physical Journal B Condensed Matter* 38(2), 321–330 (2004)
19. Newman, M.E.J.: Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America* 103(23), 8577–8582 (2006)
20. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Physical Review E - Statistical, Nonlinear and Soft Matter Physics* 69(2 Pt 2), 16 (2004)
21. Slaninová, K., Kocyan, T., Martinovič, J., Dráždilová, P., Snášel, V.: Dynamic time warping in analysis of student behavioral patterns. In: *Proceedings of the DATESO 2012, Annual International Workshop on DAtabases, TExts, Specifications and Objects*, pp. 49–59 (2012)
22. van der Aalst, W.M.P.: *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, 1st edn. Springer, Heidelberg (2011)
23. van der Aalst, W.M.P., Reijers, H.A., Song, M.: Discovering social networks from event logs. *Comput. Supported Coop. Work* 14(6), 549–593 (2005)
24. White, S., Smyth, P.: A spectral clustering approach to finding communities in graphs. In: *Proceedings of the Fifth SIAM International Conference on Data Mining*, vol. 119, p. 274 (2005)
25. Yang, F., Shen, R., Han, P.: Construction and application of the learning behavior analysis center based on open e-learning platform (2002)
26. Zorrilla, M.E., Menasalvas, E., Marín, D., Mora, E., Segovia, J.: Web Usage Mining Project for Improving Web-Based Learning Sites. In: Moreno Díaz, R., Pichler, F., Quesada Arençibia, A. (eds.) *EUROCAST 2005. LNCS*, vol. 3643, pp. 205–210. Springer, Heidelberg (2005)

Plan and Goal Structure Reconstruction: An Automated and Incremental Method Based on Observation of a Single Agent

Bartłomiej Józef Dzieńkowski and Urszula Markowska-Kaczmar

Wrocław University of Technology, Poland

{bartlomiej.dzienkowski, urszula.markowska-kaczmar}@pwr.wroc.pl

Abstract. Plan reconstruction is a task that appears in plan recognition and learning by practice. The document introduces a method of building a goal graph, which holds a reconstructed plan structure, based on analysis of data provided by observation of an agent. The approach is STRIPS-free and uses only a little knowledge about an environment. The process is automatic, thus, it does not require manual annotation of observations. The paper provides details of the developed algorithm. In the experimental study properties of a goal graph were evaluated. Possible application areas of the method are described.

Keywords: data analysis, hierarchical plan, sub-goals, reconstruction, recognition, machine learning, agent system.

1 Introduction

Building a plan in complex environments is a difficult task. Usually a system designer simplifies a world model, thus, it can be rigidly constrained by a designer's knowledge. In practice this approach gives acceptable results. The process can be improved by analysis and use of existing expert plans. There are sources from which we can mine new plans from the perspective of an observer. However, we do not have an access to some internal parts of the model, especially to a communication channel between planning and execution modules. In fact, we can analyze events in the environment to guess sub-goals of observed agents. Here the task is similar to plan recognition. However, often plan recognition is limited to predefined goals and tasks [3]. Thus, we would like to provide a method that automatically extracts reusable plan elements from a set of recorded observations. We believe that avoiding predefined STRIPS-like conditions and operators increases model flexibility [1].

The following section of the paper motivates the undertaken subject and indicates possible fields of application. Next, references to other works and the state of art is briefly introduced. The next part of the document provides details of the developed method. Subsequently, study results are presented. The final sections are conclusions and further work.

2 Motivation

The primary objective of the study is to provide a method that supports planning by utilizing existing data containing effective solution templates. Moreover, the crucial assumption is to deliver a solution which requires a minimal amount of predefined knowledge, so that the model can become more flexible and domain independent.

In order to give a better understanding of the target application domain, we refer to a general example. Let's assume there is an abstract environment in which agents continuously fulfill their goals. The agents are black-boxes characterized by a high level of intelligence. We are interested in their work and solutions they generate because we can use them for solving our own problems related to the same or an analogical environment. However, we do not have enough resources to build a complete agent model since it is too complex.

The agent's actions are the result of its decisions. Subsequently, the decisions are the product of a plan. A nontrivial task is to reconstruct a plan on the basis of observed actions and events that occur in an environment. Furthermore, a structure of sub-goals inside a plan can be updated and extended when new observations are provided.

The developed method collects plans from observed expert agents and merges them into a map of goals connected by observed solutions. It supports transferring knowledge from experts and learning by practice approaches. However, the algorithm can be applied in many fields. It can be used in network infrastructures for predicting an attacker's goals and sub-goals based on historical intrusions [1][2]. Another example is building a rescue robot or other a human supporting machine by utilizing recorded human experts' actions [5][4][6].

3 State of the Art

Mining data from an expert's actions have been previously done in several papers [7][9]. However, those approaches are based on STRIPS-like operators. In this case the method is based on a general environment state representation, which is an alternative approach. In fact, STRIPS operators (or other planning concept) can be added later, but for the purpose of recorded data analysis our algorithm focuses on an encoded sequence of environment states.

Defining a set of STRIPS operators is an additional task to accomplish during development of a system and it can be difficult for complex environment models. Not only operators, but also preconditions cause problems. In paper [7] authors reported a problem with overgrowth of negative preconditions that detracted quality of solutions. Finally, every predefined data greatly constrains system flexibility. In fact, STRIPS provides human-readable planning rules, but it is not perfectly suitable for automatically extracted state transition relations and state transition operators.

Comparing to [7] our approach does not require a set of expert annotations for each observation, or any other effort related to manual marking of recorded data. All the provided input data is processed automatically.

4 Method Description

This section provides details and a pseudo-code of the developed method. However, before we proceed to the detailed description of the algorithm a general overview on a model is required.

4.1 General Overview

The model consists of an environment and modules that share data flow (Fig. 1). The environment is a well-defined part of an agent system exposed to the observer module, which tracks and records events. The module holds a repository of recorded state sequences. Our algorithm is located in the plan reconstruction module. The output of the module is passed to the planner. An agent controlled by the planner can be placed in the same or a similar environment.

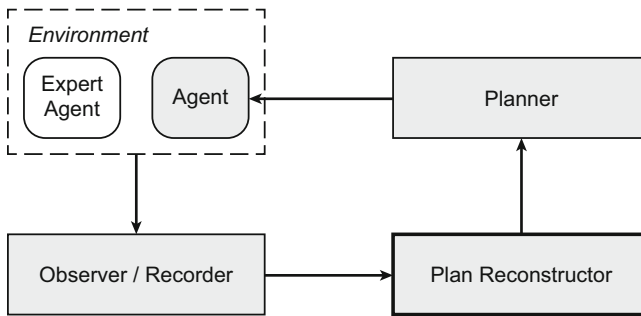


Fig. 1. Module organization and data flow of the discussed model

The method accepts a set of recorded state sequences on input and returns a goal graph on output. Each input state sequence is an ordered list of encoded environment states collected by a recorder component during a single simulation run. A sequence covers the environment state from the initial to the final state in which agent accomplishes its final task.

A returned goal graph represents a structure of directed connections between environment goal states. It enables us to find the shortest path from the initial to the final state. At the same time it provides a hierarchy of goals and information about method of their accomplishment. In fact, two types of nodes can be found. The first one is a Key Node that stands for a goal state. The second, a Transitive Node is used for connecting Key Nodes. A sequence of adjacent Transitive Nodes between two Key Nodes is a path that stores simple state-change information. Subsequently – a goal graph is a data that can be successfully applied in hierarchical planning methods [10]. However, in this paper we focus on the description of building a goal graph on the basis of an input set of state sequences.

Table 1. Symbol description

<i>state</i>	–	encoded state of the environment
<i>StateSequence</i>	–	sequence of encoded states
<i>SetOfStateSequences</i>	–	set of state sequences
KEY	–	label for a Key State
TRANSITIVE	–	label for a Transitive State
<i>GoalGraph</i>	–	directed graph in which each node stores a value and a list of references to child nodes

4.2 Algorithm Details

The algorithm consists of several steps. Its general framework is presented by Alg. 1 (look for the symbol description in Table 1). The first step is to initialize an empty graph data structure. Next, each state sequence is preprocessed by marking all elements as a Key or Transitive State (in analogy to a Key and Transitive Node). If the goal graph is still empty, the first element from the first state sequence is inserted as the root node. In the same loop the state sequence is broken into a set of paths each of which is a sub-sequence that starts and ends with a Key State – the algorithm is straightforward and specified by Alg. 3. Each sub-sequence (path) is attached to the goal graph (Fig. 2).

Alg. 1. Build Goal Graph from a Set of State Sequences

```

function buildGoalGraph(SetOfStateSequences)
1: GoalGraph := ⟨value = nil, children = {}⟩
2: for all StateSequence in SetOfStateSequences do
3:   StateSequence := markStates(StateSequence)
4:   if GoalGraph.value = nil then
5:     GoalGraph.value := StateSequence.getFirstElem()
6:   end if
7:   Paths := breakApart(StateSequence)
8:   for all path in Paths do
9:     GoalGraph := attachPath(GoalGraph, path)
10:  end for
11: end for
12: return GoalGraph

```

The method of marking states as a Key or Transitive State showed by Alg. 2 depends on environment characteristics. In general, when we consider some dynamic environment, we should focus on environment state properties that an agent has modified. We can assume that these properties are more important, because they are somehow related to tasks of an agent. Thus, a modification applied to the environment state can be recognized as fulfillment of a goal. A specific approach for marking states problem is described in the next section.

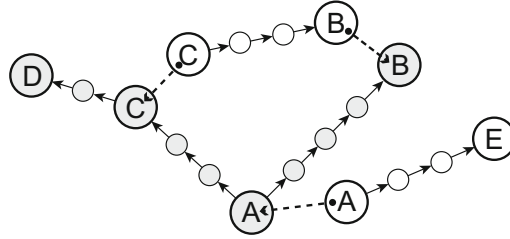


Fig. 2. An example of a goal graph. The larger circles are Key Nodes and the smaller are Transitive Nodes. Dashed arrows show places where new paths are attached to the goal graph. In this case A is the initial node, $\{B, D, E\}$ are the final nodes and C can be considered as a sub-goal.

Alg. 2. Mark States

function markStates(*StateSequence*)

```

1: for all state in StateSequence do
2:   find state properties modified by agent and mark them as important, ignore the
   rest
3:   if modification has permanent or long term effect
        $\vee$  state = StateSequence.getFirstElem() then
4:     mark state as KEY
5:   else
6:     mark state as TRANSITIVE
7:   end if
8: end for
9: return StateSequence
```

Alg. 3. Break Apart State Sequence

function breakApart(*StateSequence*)

```

1: Paths := {}
2: path := {}
3: for all state in StateSequence do
4:   path.add(state)
5:   if state  $\neq$  StateSequence.getFirstElem() then
6:     if state is KEY then
7:       Paths.add(path)
8:       path := {}
9:       path.add(state)
10:    end if
11:  end if
12: end for
13: return Paths
```

The last part of the method, attaching a path to a goal graph is presented by Alg. 4. Since a node in a goal graph holds a state representation, we require a special method of node comparison that provides some level of generality. In other words, we are more interested in checking whether two nodes are equivalent and represent a very similar situation in the environment rather than finding out whether they are identical. This can be done referring to the previously mentioned state property importance concept. The comparison method is used for searching nodes in a goal graph. It is used for the first time for collecting all nodes equivalent to the first element of the input path, we call them a set of start nodes. In the next step, we use the same method of finding a set of end nodes each of which is equivalent to the last element of the input path. These two node sets enables us to connect gaps between each of two start and end nodes using the input path. In case a connection between two nodes already exists, it is replaced by a shorter path. If the set of end nodes is empty, the input path is simply attached to the start node.

Alg. 4. Attach Path to Goal Graph

```

function attachPath(GoalGraph, path)
1: // value comparison is made according to the importance of state properties
2: StartNodes := GoalGraph.findNodesWithValue(path.getFirstElem())
3: EndNodes := GoalGraph.findNodesWithValue(path.getLastElem())
4: for all startNode in StartNodes do
5:   for all endNode in EndNodes do
6:     GoalGraph.connectNodes(startNode, endNode, path)
7:   end for
8:   if EndNodes.getCount() = 0 then
9:     GoalGraph.attachPath(startNode, path)
10:  end if
11: end for
12: return GoalGraph

```

The described algorithm is a set of general steps that can be adjusted according to some special cases. However, there is an additional feature of the method that is worth mentioning. The algorithm was showed to process all of the state sequences in a single block. It is possible to organize the algorithm flow to work incrementally. Thus, new state sequences can be attached to a goal graph online. A set of sequences will produce the same result no matter the order of attaching to a graph. Additionally, because duplicate paths in a goal graph can be replaced, the graph is not expanding infinitely.

5 Test Environment

For the purpose of the research we have developed an agent environment, which is described in this section. The system was designed to cover a general case

of a possible application, but maintaining its simplicity. The environment space is a discrete grid. Each space cell is a resource, but some of them are blocked or their acquisition causes a special effect in the environment. In fact, there are five kinds of resources: an empty space, a permanently blocked space, a gate (a temporarily blocked space), a trigger (locks and unlocks gate), and an objective. A configuration of the environment used in the experiment is presented in Fig. 3. An agent starts a simulation run always in the same position cell. It is able to acquire and move to an unblocked space resource provided that it is in an adjacent cell. If an agent enters a trigger resource, the trigger's internal state $\in \{0, 1\}$ is switched. Simultaneously, a gate linked to the trigger changes its blocking state. A simulation run reaches the end when an agent acquires the objective resource.

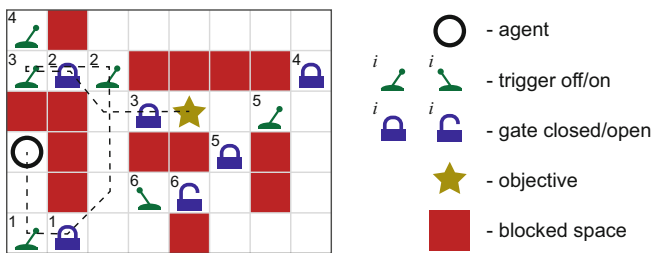


Fig. 3. A resource distribution of an example environment. The dashed line follows an agent’s path from the initial position to the objective in 13 steps. A trigger opens (or closes) a gate with the same index number.

An environment state is encoded as a two-dimensional array of state properties. Each state property in the array corresponds to a cell in the space grid. A state property stores a type of a resource and its internal state. Additionally, it shows a position of an agent. Despite the fact that the resource acquisition action can be invoked at any time and it has assigned an execution time, the observer module records only state changes.

Regarding the state marking method from the previous section, in this specific environment it is reasonable to mark state as a Key State whenever an agent uses a trigger or reaches an objective. However, there is still an issue of comparing equivalent states. The problem can be solved by assigning an importance flag to a state property. When an agent modifies a state property for the first time, the property is set to important. Fig. 4 explains the procedure by an example. Hereby, each important state property must match between two states to be considered as equivalent (Fig. 5). The comparison function also checks a stored position of an agent to maintain consistency of a goal graph.

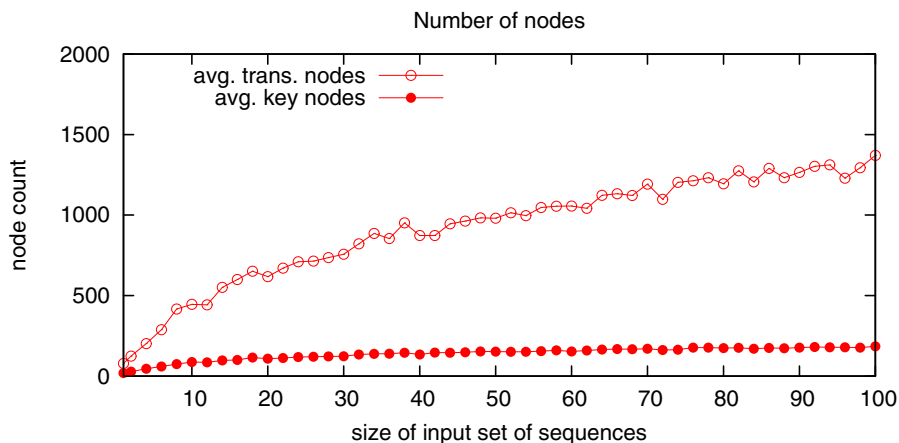


Fig. 6. An average number of transitive and key nodes with respect to a size of an input set of sequences

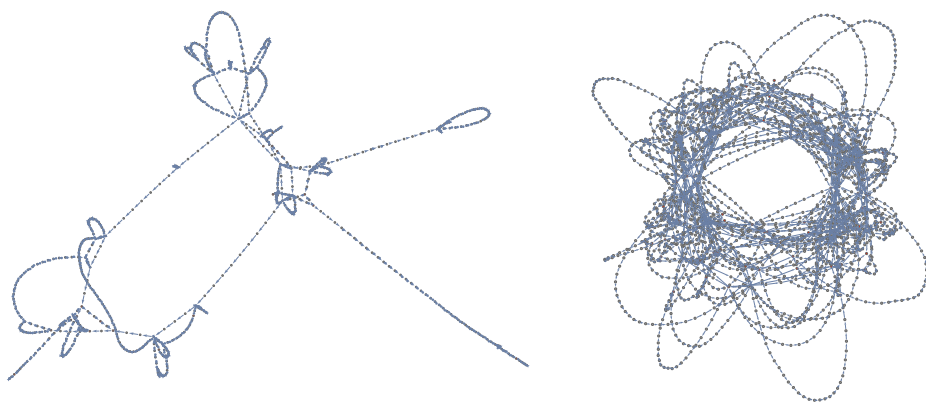


Fig. 7. On the left a goal graph with 546 nodes built from 10 state sequences and on the right a goal graph with 2953 nodes built from 7000 state sequences

in the future while shorter paths between Key Nodes are provided. However, in an early phase new connections are still revealed and added. The experiment showed that even for our simple environment a goal graph can look very complex – selected graphs are presented in Fig. 7.

Anyhow, Transitive Nodes are insignificant. In fact, they can be pruned since a transition between two connected Key Nodes can be easily found with the aid of a heuristic. More importantly the number of Key Nodes is stable. Even though a number of Key States is proportional to a number of permutation of switchable state properties, complexity of a graph depends on actions of an observed agent.

In the next part of the experiment we have studied how the number of input sequences and graph expansion affects the length of the shortest path between the initial and the final node. The results are visualized in Fig. 8. Clearly an increase of input sequences positively influences a solution quality. Finally, a long-run test showed that a much larger size of input gives no real improvement of the shortest path length (Table 3).

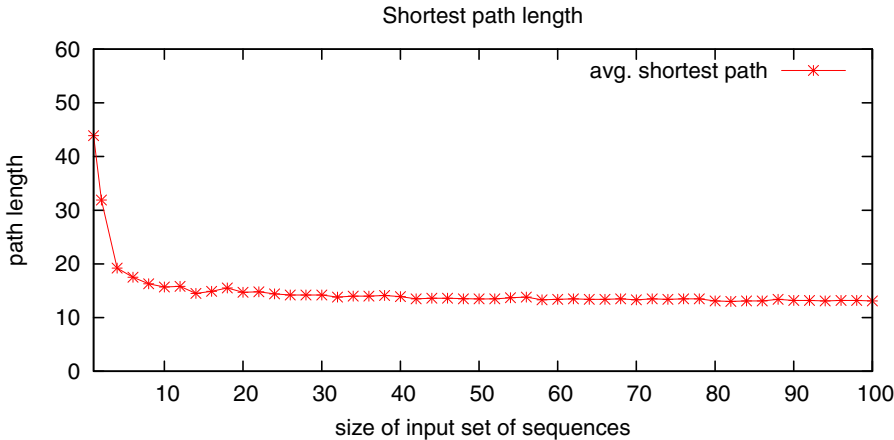


Fig. 8. An average shortest path from the initial to the nearest final node with respect to a size of an input set of sequences

Table 3. Statistics of the biggest tested input sequences set. The numbers stands for: size of an input set of sequences, average number of key nodes, average number of transitive nodes and shortest path from the initial to the nearest final node.

input set size	avg. key nodes	avg. trans. nodes	avg. shortest path
7000	213	2737.3	13

7 Conclusions

The described algorithm differs from other STRIPS-based methods. It requires only basic information about types of elements in an environment. Expert knowledge about relations between objects and their use is compiled into a goal graph. Thus, workload of a system designer is reduced.

Despite the fact that the method in some cases can have a limited application, its main advantage is flexibility and automatic data acquisition support. A planner based on the approach is not required to manipulate STRIPS conditions and operators. In fact, it can use a set of general transition functions to move between graph nodes. The set of transitions can be automatically extended based on observed expert solutions.

However, the method is not free from flaws. It requires a large amount of input data. Thus, it can be applied for analysis of existing models which continuously generate data (e.g., multiplayer game matches). Additionally, the algorithm in its original form can lead to performance problems. It is required to utilize some graph search optimizations before applying in practice. Finally, the approach can evolve to a hybrid model.

8 Further Work

The described algorithm is dedicated to environments with a single agent, or many agents that work independently. This assumption limits practical application. However, the paper refers to an early stage of the research that aims on an analogical approach but dedicated to a group of cooperating agents. In the new case, the algorithm can no more entirely rely on environment state comparison. A new approach based on cooperation patterns should be proposed. Hereby, not a state change event but an execution of cooperation pattern will link nodes in the goal graph [8]. Finally, in order to provide system flexibility, the cooperation patterns should be automatically recognized and mined.

References

1. Gu, W., Ren, H., Li, B., Liu, Y., Liu, S.: Adversarial Plan Recognition and Opposition Based Tactical Plan Recognition. In: International Conference on Machine Learning and Cybernetics, pp. 499–504 (2006)
2. Gu, W., Yin, J.: The Recognition and Opposition to Multiagent Adversarial Planning. In: International Conference on Machine Learning and Cybernetics, pp. 2759–2764 (2006)
3. Camilleri, G.: A generic formal plan recognition theory. In: International Conference on Information Intelligence and Systems, pp. 540–547 (1999)
4. Takahashi, T., Takeuchi, I., Matsuno, F., Tadokoro, S.: Rescue simulation project and comprehensive disaster simulator architecture. In: International Conference on Intelligent Robots and Systems, vol. 3, pp. 1894–1899 (2000)
5. Takahashi, T., Tadokoro, S.: Working with robots in disasters. *IEEE Robotics and Automation Magazine* 9(3), 34–39 (2002)
6. Chernova, S., Orkin, J., Breazel, C.: Crowdsourcing HRI through Online Multiplayer Games. In: Proceedings of AAAI Fall Symposium on Dialog with Robots, pp. 14–19 (2010)
7. Wang, X.: Learning by Observation and Practice: An Incremental Approach for Planning Operator Acquisition. In: Proceedings of the 12th International Conference on Machine Learning, pp. 549–557 (1995)
8. Ehsaei, M., Heydarzadeh, Y., Aslani, S., Haghighat, A.: Pattern-Based Planning System (PBPS): A novel approach for uncertain dynamic multi-agent environments. In: 3rd International Symposium on Wireless Pervasive Computing, pp. 524–528 (2008)

9. Nejati, N., Langley, P., Konik, T.: Learning Hierarchical Task Networks by Observation. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 665–672 (2006)
10. Sacerdoti, E.D.: A Structure For Plans and Behavior. In: Artificial Intelligence Center. Elsevier North-Holland, Technical Note 109 (1977)
11. Fikes, R., Nilsson, N.: STRIPS: a new approach to the application of theorem proving to problem solving. In: IJCAI, pp. 608–620 (1971)

On Spectral Partitioning of Co-authorship Networks

Václav Snášel¹, Pavel Krömer¹, Jan Platoš¹,
Miloš Kudělka¹, and Zdeněk Horák¹

Department of Computer Science, VŠB-Technical University of Ostrava,
17.Listopadu 15/2172, 708 33 Ostrava-Poruba, Czech Republic
{vaclav.snasel,pavel.kromer,jan.platos,
milos.kudelka,zdenek.horak}@vsb.cz

Abstract. Spectral partitioning is a well known method in the area of graph and matrix analysis. Several approaches based on spectral partitioning and spectral clustering were used to detect structures in real world networks and databases. In this paper, we explore two community detection approaches based on the spectral partitioning to analyze a co-authorship network. The partitioning exploits the concepts of algebraic connectivity and characteristic valuation to form components useful for the analysis of relations and communities in real world social networks.

Keywords: spectral partitioning, algebraic connectivity, co-authorship, DBLP.

1 Introduction

Spectral clustering (or spectral partitioning) is a useful method for partitioning and clustering of graphs and networks with solid mathematical background and clear interpretation. The ubiquity of social and communication networks in today's information society hand in hand with the increasing power of computers makes the usage of algebraic techniques such as spectral clustering very practical. In this work, we use the spectral partitioning to analyze selected parts of the DBLP¹, a large database of computer science publications. The DBLP can be seen as a vast, dynamic and constantly updated social network that captures several years of author co-operations in the form of joint publications. It is very interesting for social network (SN) researcher because the authors can be easily grouped based on their affiliations, areas of interest, and advisor-advisee relationship. Moreover, we can trace in the DBLP the development of each author's activities, types of activities, areas of interest and so on.

In this paper, we present two spectral partitioning based algorithms to iteratively detect communities in the DBLP (and social networks in general).

¹ <http://www.informatik.uni-trier.de/~ley/db/>

2 Spectral Graph Clustering

The basics of the spectral clustering (SC) were introduced in 1975 by M. Fiedler [4]. Fiedler's work defined spectral clustering for both, unweighted and weighted graphs. The following definitions apply to weighted graphs because the edges in a co-authorship network intuitively have different weights. An edge between two authors that have published one joint paper has different quality (i.e. weight) than an edge between two authors that have published a large number of joint papers through the years. The frequency, regularity, and age of such co-operations can be a hint for an edge weighting scheme.

Definition 1 (Generalized Laplacian of weighted graph G). For a graph $G = (V, E)$, the generalized Laplacian is the matrix of the quadratic form

$$(A_C(G)x, x) = \sum_{(i,k) \in E} c_{ik}(x_i - x_k)^2 \quad (1)$$

The $A_C(G)$ can be easily computed:

$$a_{ik} = \begin{cases} 0 & \text{if } i \neq k \text{ and } (i, k) \notin E \\ -c_{ik} & \text{if } i \neq k \text{ and } (i, k) \in E \end{cases} \quad (2)$$

$$a_{ii} = -\sum_{k \neq i} c_{ik} \quad i, k \in N \quad (3)$$

Definition 2 (Algebraic connectivity of weighted graph G). The algebraic connectivity of graph G denoted $a_C(G)$ is the second smallest (first non-zero) eigenvalue of $A_C(G)$. Let $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ be the eigenvalues of $A_C(G)$. Then $a_C(G) = \lambda_2$.

The algebraic connectivity $a_C(G)$ is also known as the Fiedler value [17].

Definition 3 (Characteristic valuation of G). The characteristic valuation of G (also known as the Fiedler vector of G) denoted $\mathbf{a}(G) = (a_1, \dots, a_n)$ is defined by the values of the eigenvector corresponding to $a_C(G)$.

The characteristic valuation assigns a non-zero (positive or negative) value to each vertex in the graph in a natural way. There is a number of interesting properties of $a_C(G)$ and \mathbf{a} , for example [4, 6, 17]:

- $a_C(G)$ is positive iff G is connected.
- if $a_C(G)$ is small, then a graph cut according to the values of vertices in $\mathbf{a}(G)$ will generate a cut with good ratio of cut edges to separated vertices.
- $\mathbf{a}(G)$ represents an ordering (Fiedler ordering) which can be used for spectral partitioning of connected graphs (for the rationale see theorem [1]).

Theorem 1. For a finite connected graph G with n vertices that has a positive weight c_{ik} assigned to each edge (i, k) , characteristic valuation $\mathbf{a}(G)$, and any $r \geq 0$ let

$$M(r) = \{i \in N | y_i + r \geq 0\} \quad (4)$$

The subgraph $G(r)$ induced by G on $M(r)$ is connected.

Via theorem 1 can be defined iterative (stepwise) partitioning of connected graph G into connected subgraph $G(r)$ and general subgraph $G \setminus G(r)$. Via theorem 1 can be also defined iterative elimination of vertices with lowest significance to the graph so that the remainder of the graph is connected. The proof of theorem 1 can be found in [4].

2.1 Graph Partitioning

A graph $G = (V, E)$ can be partitioned into two disjoint sets A, B such that $A \cup B = V$ and $A \cap B = \emptyset$. The *cut* value, which describes the dissimilarity between the two partitions, can be defined as the sum of weights of the edges removed by the cut [16]:

$$\text{cut}(A, B) = \sum_{i \in A, j \in B} c_{ij} \quad (5)$$

It can be shown that the Fiedler vector represents solution for finding partitions A and B such that the following cost function (the *average cut*) is minimized [15, 16]:

$$\text{Acut}(A, B) = \frac{\text{cut}(A, B)}{|A|} + \frac{\text{cut}(B, A)}{|B|} \quad (6)$$

The average cut is a measure with known imperfections [16]. However, its usage is simple and its computation is fast.

3 Related Work

As the need for efficient analysis of graph-like structures including social networks is growing, there was much attention given to spectral partitioning and spectral clustering of graphs. In this section, we provide brief state of the art of graph partitioning methods based on spectral clustering.

The use of spectral partitioning for graph analysis was advocated by Spielman and Teng [17]. They have shown that spectral partitioning works well for bounded-degree planar graphs and well-shaped d-dimensional meshes. Today, methods based on spectral clustering are being used to analyze the structure of a number of networks.

An influential study on spectral clustering and its application to image segmentation was published in 2000 by Shi and Malik [16]. The authors approached the graph partitioning task from the graph cuts point of view. They described the graph cut defined by the Fiedler vector and called it *average cut*. The average cut was shown to be good at finding graph splits whereas the newly defined *normalized cut* was designed to compute the cut costs as a ratio of cut edge weights to all edge weights in the segments. The normalized cut was shown to be useful when seeking partitions that are both, balanced and tight. On the other hand,

a study by Sarkar and Soundararajan showed that the increased computational cost of the normalized cut does not result in statistically better partitions [14].

Ding et al. [3] have proposed in 2001 another graph cut algorithm, the *min-max cut*, and showed its usefulness for partitioning real world graphs into balanced parts. Bach and Jordan [1] proposed an algorithm based on a new cost function evaluating the error between given partition and a minimum normalized graph cut. The partitions can be learned from given similarity matrix and vice-versa - the similarity matrix can be learned from given clusters. Similarity of nodes i and j in this context means large weight of the edge (i, j) , i.e. large c_{ij} . The method leads to clusters with large in-cluster similarity and small inter-cluster similarity of nodes.

The algebraic connectivity has been used to define a new method for construction of well-connected graphs by Gosh and Boyd in 2006 [5]. The algorithm uses the properties of algebraic connectivity and defines an edge perturbation heuristic based on the Fiedler vector to choose from the set of candidate edges such edges that would improve the value of $a_C(G)$.

The work of Ruan and Zhang [13] presents an application of spectral partitioning in the area of social networks. The authors developed an efficient and scalable algorithm *Kcut* to partition the network to k components so that the modularity Q of community structures is maximized. For more details on Q see [13]. The usefulness and effectiveness of *Kcut* was demonstrated on several artificial and real world networks.

Mishra et al. [12] have used spectral clustering for social network analysis in 2007. They aimed at finding good cuts on the basis of conductance, i.e. the ratio of edges crossing the cut to the minimum volume of both partitions. Volume in this context means the number of edges incident with vertices in the sub-graph. Moreover, the proposed algorithm was able to find overlapping clusters with maximum internal density and external sparsity of the edges.

Kurucz et al. [8,9] have applied spectral clustering to telephone call graphs and to social networks in general. In their studies, the authors discussed various types of Laplacians, edge weighting strategies, component size balancing heuristics, and the number of eigenvectors to be utilized. The work proposed a *k-way* hierarchical spectral clustering algorithm with heuristic to balance clusters and showed its superiority over the Divide-and-Merge clustering algorithm.

In 2008, Leskovec et al. [10] investigated the statistical properties of communities in social and information networks. They used the *network community profile plot* to define communities according to the conductance measure. Their work demonstrated that the largest communities in many real world data sets blend with the rest of the graph with increasing size, i.e. their conductance score is decreasing.

Xu et al. [18] have analyzed social networks of spammers by spectral clustering. They have used the normalized cut diassociation measure that is known to minimize the normalized cut between clusters and simultaneously maximize the normalized association within clusters.

A recent work on generalized spectral clustering based on the graph p -Laplacian is due to Bühler and Hein [2]. It was shown that for $p \rightarrow 1$ the cut defined by Fiedler vector converges to the Cheeger cut. The p -Spectral Clustering using the p -Laplacian, a nonlinear generalization of the graph Laplacian, was in this paper evaluated on several data sets.

An overview of spectral partitioning with different Laplacians was given by Luxburg in [11]. The study contained a detailed description of the algorithm, properties of different Laplacians and a discussion on suitability of selected Laplacians for given task.

In general, many variants of the basic spectral clustering algorithm were used to partition graphs and detect network structure in multiple application areas with good results. Real world networks and social networks constituted by the natural phenomena of communication, interaction, and cooperation are especially interesting application field for the spectral partitioning.

4 Spectral Partitioning of Co-author Communities in the DBLP

We have defined two iterative partitioning algorithms based on spectral clustering and algebraic connectivity to find co-author communities in the graph. In the algorithm 1 (*simple iterative spectral partitioning*, SimpleISP) was the initial connected graph divided into two subgraphs, each containing vertices with positive valuation (and incident edges) and vertices with negative valuation (and incident edges) respectively. For the next iteration was used as an input the subgraph that contained the *author vertex*. If the *author vertex* belonged to the negative subgraph (that was not guaranteed to be connected), all vertices that were not connected to the *author vertex* were removed. The partitioning ended when the subgraph contained only single vertex (*author vertex*). This variant of the algorithm creates in every iteration a smaller (narrower) community centered around the author.

In the algorithm 2 (*iterative spectral partitioning*, ISP), the graph for next iteration was created differently. In each iteration, we removed all vertices that had lower characteristic valuation than the *author vertex*. It is guaranteed that the resulting subgraph is connected. The algorithm ended when the *author vertex* had the lowest valuation among all vertices in the graph (i.e. it was not possible to remove loosely connected vertices). This variant of the algorithm centers on the community to which the author belongs rather than on the author herself.

4.1 Experiments

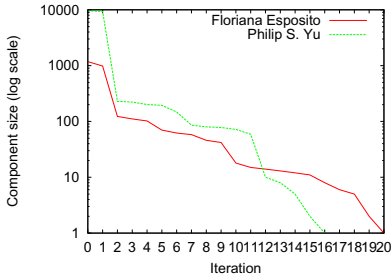
To observe the communities generated by proposed algorithms, we have conducted a series of experiments with the DBLP data. We have downloaded the DBLP dataset from April 2010 in XML and preprocessed it for further usage. We have selected all conferences held by IEEE, ACM or Springer, which gave us 9,768 conferences. For every conference we identified the month and year of

Algorithm 1. Simple iterative spectral partitioning (SimpleISP)

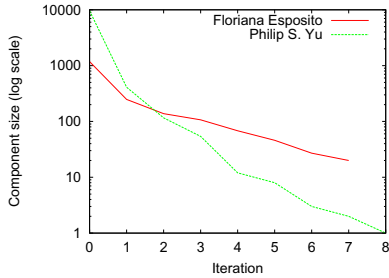
- 1: Find a connected subgraph S containing the vertex of selected author (*author vertex*), vertices of all his or her co-authors, vertices of all their co-authors, and edges among them.
 - 2: **while** $|S| > 1$ **do**
 - 3: Compute $a(S)$
 - 4: Cut S according to $a(S)$
 - 5: Let S^+ contain all vertices and incident edges for which the value of $a(S)_i \geq 0$ and S^- contain all vertices and incident edges for which $a(S)_i < 0$.
 - 6: Remove all edges between vertices in S^+ and S^- .
 - 7: **if** *author vertex* $\in S^+$ **then**
 - 8: $S = S^+$
 - 9: **else**
 - 10: $S = S^-$
 - 11: **end if**
 - 12: Remove from S all vertices that are not connected to *author vertex*
 - 13: **end while**
-

Algorithm 2. Iterative spectral partitioning (ISP)

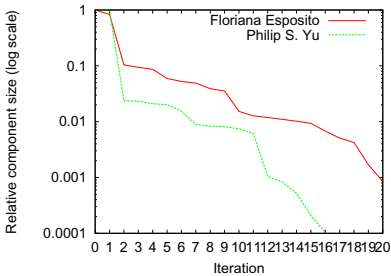
- 1: Find a connected subgraph S containing the vertex of selected author (*author vertex*), vertices of all his or her co-authors, vertices of all their co-authors, and edges among them.
 - 2: **repeat**
 - 3: Compute $a(S)$
 - 4: Get the valuation of author vertex $a_{AV} = a(S)_{\text{author vertex}}$
 - 5: Remove from S all vertices with valuation lower than a_{AV} . The rest is connected.
 - 6: **until** $\min a(S) < a_{AV}$
-



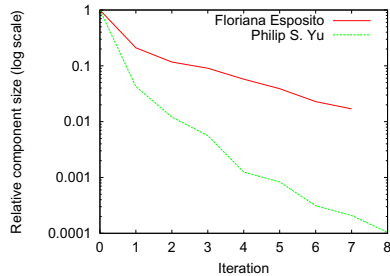
(a) Component size (SimpleISP).



(b) Component (ISP).



(c) Relative component size (SimpleISP).



(d) Relative component size (ISP).

Fig. 1. Size of author components during the partitioning

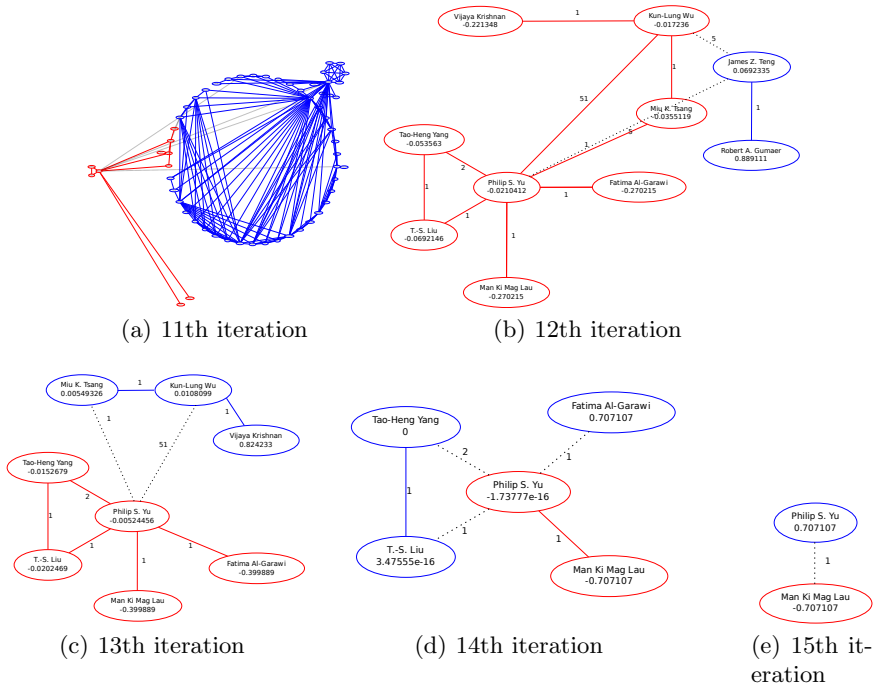


Fig. 2. Philip S. Yus network in selected iterations of SimpleISP

the conference. In the next step we extracted all authors having at least one published paper in the mentioned conferences (as authors or co-authors). This gave us 443,838 authors. Using the information about authors and their papers we were able to create a set of cooperations between these authors consisting of 2,054,403 items. Finally, the cooperations were represented as a graph. A vertex in the graph represented one author and an edge represented a co-operation between the authors (joint publication). The edges were weighted according to the number of joint publications between the two authors, i.e. if two authors published one joint work, the weight of the edge between their vertices was 1. If they co-operated on n papers, the weight of the edge between their vertices was n . We note that this weighting scheme is quite naïve and much more sophisticated approaches can be used, but such a research is out of the scope of this paper.

We have selected two authors and investigated spectral partitions of the connected graph consisting of their co-authors and their co-authors' co-authors. We investigated only two levels of co-authors to obtain components that could be manually inspected. Floriana Esposito and Philip S. Yu were investigated in a recent work on co-authorship network analysis [7]. Floriana Esposito is an author who has been active since 1990 and who has a lot of strong ties whereas Philip S. Yu is an author with the greatest number of records in the data set and with a number of strong co-authors. We have applied both, simple iterative spectral partitioning and strict iterative spectral partitioning to the subgraphs around selected authors.

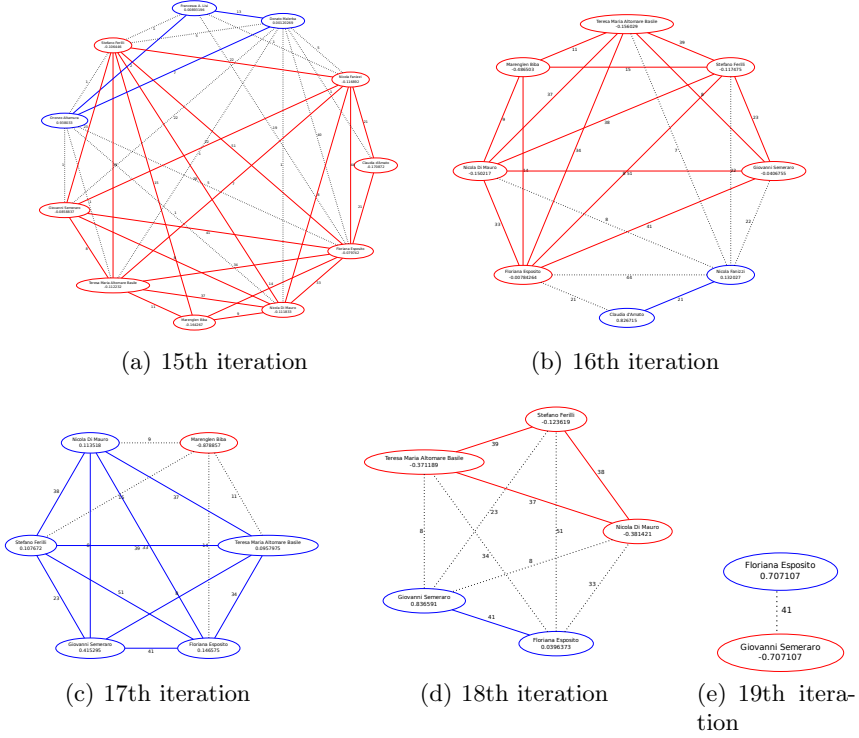


Fig. 3. Floriana Esposito network in selected iterations of SimpleISP

4.2 Results

The process of iterative spectral partitioning of subgraphs for Philip S. Yu and Floriana Esposito is captured in Fig. 1. The figures illustrate the sizes of components (communities) of both authors in each iteration of SimpleISP and ISP. Initial size of P. S. Yus component was 9607 and initial size of F. Esposito component was 1180, so for a better comparison, the relative component sizes are compared in Fig. 1(c) and Fig. 1(d). Figures Fig. 1(a) and Fig. 1(c) show the process of SimpleISP. We can see that both authors loose the majority of their collaborators in the second iteration. However, Floriana Esposito network keeps larger fraction of the original nodes during the whole process and it becomes larger than Philip S. Yus network after 12th iteration. The SimpleISP ended for Floriana Esposito after 20 iterations and for Philip S. Yu after 16 iterations.

The ISP process is shown in Fig. 1(b) and Fig. 1(d). In this case, the most significant reduction of the communities was done in the first iteration. Floriana Esposito network lost 932 out of 1180 nodes and Philip S. Yus network reduced from 9607 to 410 nodes. Again, the relative component size of Floriana Esposito

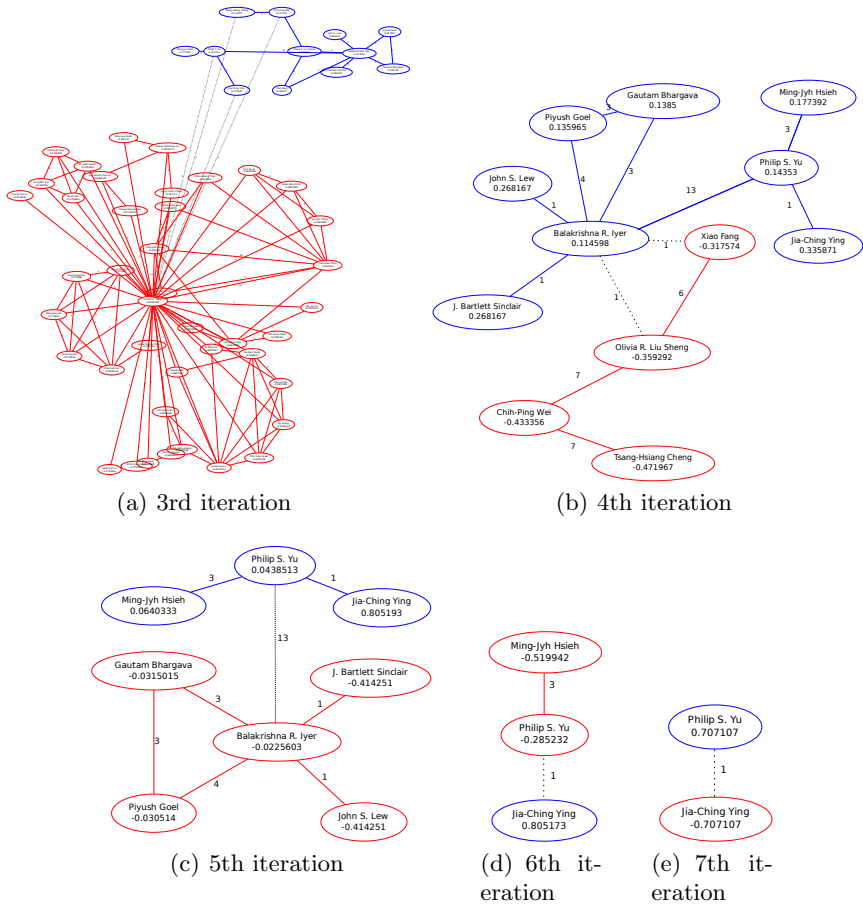


Fig. 4. Philip S. Yus network in selected iterations of ISP

was greater than the relative component size of Philip S. Yu during the whole ISP and it becomes larger than P. S. Yus community after second iteration. The ISP ended for F. Esposito after 7 iterations and the final network contained 20 nodes. In contrast, the ISP for P. S. Yu ended after 8 iterations and the final network contained only one node - the *author node*.

Examples of the partitions in selected iterations of the SimpleISP and ISP for P. S. Yu and F. Esposito are shown in Fig. 2, Fig. 3, Fig. 4, and Fig. 5 respectively. Blue and red vertices and edges represent the components and dotted edges represent the cut. The number on each vertex corresponds to characteristic valuation of the vertex and the number on each edge represents the weight of the edge, i.e. the multiplicity of author co-operation in this experiment. We note that larger graphs are shown to illustrate the structure of the community and cut rather than to provide the names of the co-authors which is printed using very small font.

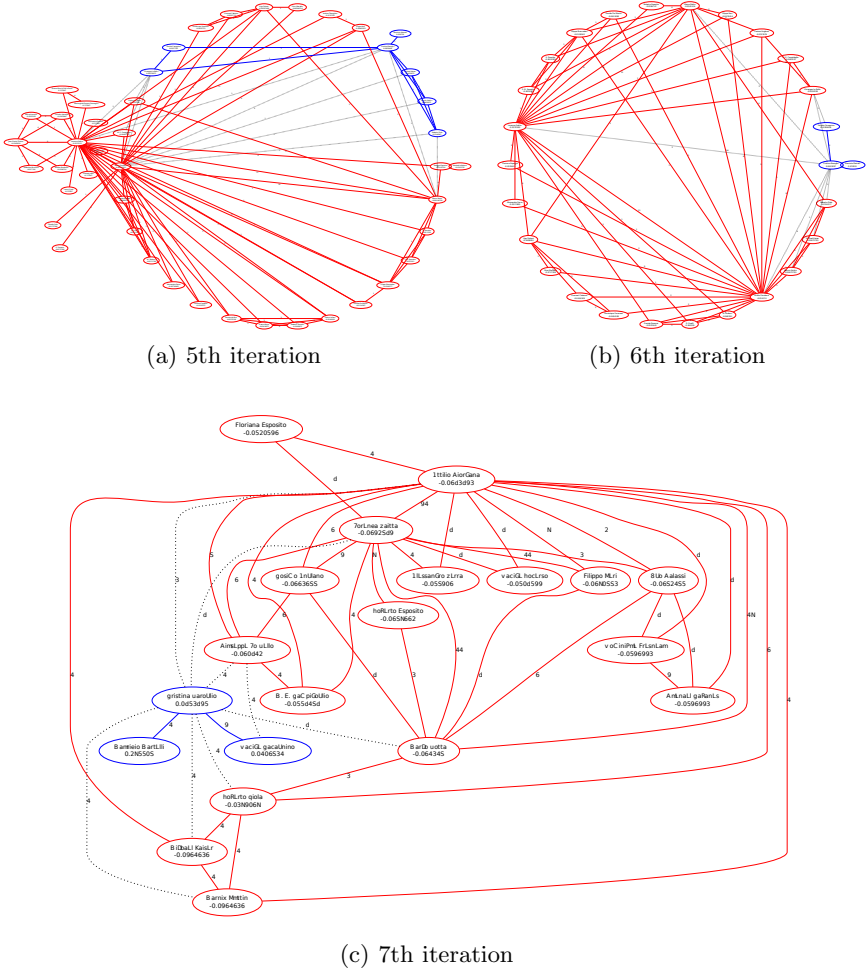


Fig. 5. Floriana Esposito's network in selected iterations of ISP

5 Conclusions and Future Work

In this paper we present two algorithms for iterative spectral partitioning of social networks. The goal of the algorithms is to find meaningful communities in networked data. We demonstrate the application of the algorithms on a co-authorship network, namely the DBLP, in which we sought for communities of selected authors. The first algorithm focused on a central node around which it iteratively created connected subgraphs, i.e. possible communities. It was searching for communities around an author. The second algorithm, following more closely the idea of algebraic connectivity and spectral clustering, focused on a community rather than on the author. It was highlighting the community to which the author belonged. We have selected two authors with

different statistical properties and searched for their communities using both approaches.

The results of the experiment show that both, the partitioning process and generated partitions, were quite different for the two authors, no matter which algorithm was used. An author with strong ties to other authors retained connection to a large number of co-author nodes during most of the partitioning process. On the other hand, a highly co-operative author lost the links to majority of his/her co-authors very early. The results support the intuition that the partitioning of such a different authors will be different. We have also observed, that the author with strong relationships to others was placed to a community of twenty collaborators whereas the highly collaborative author ended alone.

There are many directions in which this work can continue. First, the observations presented in this paper should be confirmed on a large number of authors. Second, the weighting scheme used in this study was rather simple - a different edge weighting schemes should be applied and their influence on the partitioning should be investigated. Third, in this work we have used the simple *average cut* in which we have split the network according to negative and positive values of vertex characteristic valuation. Many different cuts were proposed and their effect on co-authorship network partitioning should be investigated. Also the effect of different Laplacians should be investigated. Finally, the results of the spectral clustering of the co-authorship network should be compared to other non-spectral network and graph analytical methods.

Acknowledgments. This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the Bio-Inspired Methods: research, development and knowledge transfer project, reg. no. CZ.1.07/2.3.00/20.0073 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic.

References

1. Bach, F.R., Jordan, M.I.: Learning spectral clustering. In: Thrun, S., Saul, L.K., Schölkopf, B. (eds.) NIPS. MIT Press (2003)
2. Bühler, T., Hein, M.: Spectral clustering based on the graph p -laplacian. In: Danyluk, A.P., Bottou, L., Littman, M.L. (eds.) ICML. ACM Int. Conf. Proceeding Series, vol. 382, p. 11. ACM (2009)
3. Ding, C., He, X., Zha, H., Gu, M., Simon, H.: A min-max cut algorithm for graph partitioning and data clustering. In: Proceedings IEEE Int. Conf. on Data Mining, ICDM 2001, pp. 107–114 (2001)
4. Fiedler, M.: A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. Czechoslovak Mathematical Journal 25 (1975)
5. Ghosh, A., Boyd, S.: Growing well-connected graphs. In: 2006 45th IEEE Conference on Decision and Control, pp. 6605–6611. IEEE (2006)
6. Grady, L., Polimeni, J.R.: Discrete Calculus - Applied Analysis on Graphs for Computational Science. Springer (2010)

7. Kudělka, M., Horák, Z., Snášel, V., Krömer, P., Platoš, J., Abraham, A.: Social and swarm aspects of co-authorship network. *Logic Journal of IGPL Special Issue: HAIS 2010* (2011)
8. Kurucz, M., Benczur, A., Csalogany, K., Lukacs, L.: Spectral clustering in telephone call graphs. In: *Proc. of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis, WebKDD/SNA-KDD 2007*, pp. 82–91. ACM, New York (2007)
9. Kurucz, M., Benczúr, A.A., Csalogány, K., Lukács, L.: Spectral Clustering in Social Networks. In: Zhang, H., Spiliopoulou, M., Mobasher, B., Giles, C.L., McCallum, A., Nasraoui, O., Srivastava, J., Yen, J. (eds.) *WebKDD 2007*. LNCS, vol. 5439, pp. 1–20. Springer, Heidelberg (2009)
10. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Statistical properties of community structure in large social and information networks. In: *Proceedings of the 17th Int. Conf. on World Wide Web, WWW 2008*, pp. 695–704. ACM, New York (2008)
11. Luxburg, U.: A tutorial on spectral clustering. *Statistics and Computing* 17(4), 395–416 (2007)
12. Mishra, N., Schreiber, R., Stanton, I., Tarjan, R.E.: Clustering Social Networks. In: Bonato, A., Chung, F.R.K. (eds.) *WAW 2007*. LNCS, vol. 4863, pp. 56–67. Springer, Heidelberg (2007)
13. Ruan, J., Zhang, W.: An efficient spectral algorithm for network community discovery and its applications to biological and social networks. In: *Proceedings of the 2007 Seventh IEEE Int. Conf. on Data Mining*, pp. 643–648. IEEE Computer Society, Washington, DC (2007)
14. Sarkar, S., Soundararajan, P.: Supervised learning of large perceptual organization: graph spectral partitioning and learning automata. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(5), 504–525 (2000)
15. Shen, X., Papademetris, X., Constable, R.T.: Graph-theory based parcellation of functional subunits in the brain from resting-state fmri data. *NeuroImage* 50(3), 1027–1035 (2010)
16. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8), 888–905 (2000)
17. Spielman, D.A., Teng, S.H.: Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications* 421(23), 284–305 (2007)
18. Xu, K.S., Kliger, M., Chen, Y., Woolf, P.J., Hero III, A.O.: Revealing social networks of spammers through spectral clustering. In: *Proceedings of the 2009 IEEE International Conference on Communications, ICC 2009*, pp. 735–740. IEEE Press, Piscataway (2009)

An Efficient Binary Playfair Algorithm Using a 4×4 Playfair Key Matrix

Saswati Mukherjee¹, Matangini Chattopadhyay¹, Ayan Lahiri¹,
and Samiran Chattopadhyay²

¹ School of Education Technology, Jadavpur University, Kolkata, India
sash_cal@rediffmail.com, ayanlahiri007@gmail.com,
chttpdhy@yahoo.com

² Department of Information Technology, Jadavpur University, Kolkata, India
samiranc@it.jusl.ac.in

Abstract. Playfair cipher is a digraph cipher which is not preferred now a day for two main reasons. Firstly, it can be easily cracked if there is enough text and secondly, frequency analysis of digraph is anyway possible. This paper proposes a new solution, which encrypts / decrypts each byte by applying the Playfair on its nibbles with the help of a reduced 4×4 Key matrix. This byte by byte encryption supports any character (even multilingual character), number (of any base), symbol and any type of media file and thereby ensures flexibility. Randomness of the algorithm is achieved by rotating the key matrix randomly after encryption / decryption of each byte. Several operations are performed to support the mechanism of lightweight cryptography. The proposed method is implemented and compared with other popular ciphers on the basis of certain parameters, like Avalanche Effect, Time Complexity, and Space Requirement. The result obtained demonstrates efficiency of the proposed algorithm.

Keywords: Playfair cipher, Digraph, Encryption, Decryption, Avalanche effect.

1 Introduction

Playfair cipher was invented by Charless Wheatstone [1] in 1854 but was named after Lord Playfair who prompted the use of the cipher. In Playfair cipher, the alphabets are arranged in a 5×5 key matrix based on secret key. Though there are 26 alphabets in English language, Playfair cipher can handle only 25 alphabets. So either Q is to be discarded or any one of i/j can be used. Despite its proven efficiency the algorithm lacks on several areas. Over the years several attempts have been made to modernize the algorithm [2], [3], [4], [5], [6], [7], [8], [9], [10] to increase its acceptances by eliminating its limitations. This paper provides a new solution approach to overcome the shortcomings of the Playfair algorithm.

1.1 Algorithm

Playfair cipher is a symmetric encryption technique which uses digraph substitution [1]. This cipher encrypts alphabets based on a reference 5×5 key matrix which is formed from the given key- PASSWORD.

Table 1. Key Matrix

P	A	S	W	O
R	D	B	C	E
F	G	H	I/J	K
L	M	N	Q	T
U	V	X	Y	Z

Now two alphabets are taken at a time from the source plaintext file and then with the reference of this key matrix a new pair of cipher text are obtained. Despite of its proven efficiency, the main drawback of Playfair Cipher lies in its limitations. The limitations of the cipher are as follows.

- Only 25 alphabets of English language are supported.
- No support for numeric characters.
- Only either upper cases or lower cases are supported.
- No special characters (viz. blank space, new line, punctuations etc.) can be used.
- Unable to deal with languages other than English.
- Any type of media files cannot be encrypted.

Beside these limitations playfair is a poly alphabetic cipher. So by testing the frequency of occurrence the plain text can easily be tracked.

1.2 Related Work

The main limitation of playfair algorithm lies in its support strictly limited to 25 alphabets of English language. Over the years several attempts have been made to increase the character limit of its dataset. Some modifications increase the matrix size to enhance the character set. A 6×6 matrix supports 36 characters, which include 26 alphabets of English language and all 10 decimal numbers (0-9) [2]. But it needs more character support in order to be able to work over a large range of text file. To increase the character set the matrix size is further increased to 8×8 to allow 64 characters, which includes 26 English alphabets, 10 decimal numbers (0-9) and a selected set of 28 symbols [3]. Though these modifications increases the character set, but still they were limited. Especially 64characters are not at all sufficient in modern day encryptions. Keeping this in mind some modifications focus on the use of ASCII values in playfair. The use of 7-bit ASCII values increased the character support to 128 characters. To deal with the matrix manipulation of playfair the concept of interweaving is introduced. It actually jumbles the binary values of the ASCII codes of a set of characters. Use of multiple iteration and character substitution further increases the security. These interweaving and iteration actually leads to lots of

confusion and diffusion [4] [5]. These modifications are useful for text encryption, but for modern day encryption of multilingual character or media files, the algorithm must be modified to support binary file encryption. To achieve the desired output some adaptations are made on traditional playfair cipher. The binary values of 7-bit ASCII codes corresponds colors of ARGB color model. To utilize the maximum color limit some further calculations are made before choosing the next color of the cipher based on the key/password [6]. But still a 7-bit ASCII support limits the scope to English language only. Multilingual characters are not supported. Keeping this in mind another modification focuses on the incorporation of DNA coding in the playfair cipher. Binary 8-bit ASCII values are initially replaced by DNA bases, and then converted into amino acid groups before applying normal playfair algorithm [7]. This actually solves the problem of limited character support but due to its complicated and lengthy process it takes more time to encrypt / decrypt. These modifications actually treat the plaintext file as binary data stream and thus enrich the character set. Another major problem of the traditional playfair is the predictability of the cipher by using frequency testing of character occurrences. To overcome this a new approach was introduced which keeps track of the frequency of occurrences of each and every character in English language and replaces the every next occurrence of the character with a character of least frequency of use [10]. Though this provides a variation in case of repetitive characters but it also takes extra time as it searches for the frequency table for each and every replacement. Another efficient way to deal with the problem is to use Random Numbers [3] [8] is a popular technique. Unpredictable different random sequences are produced from Linear Feedback Shift Register by varying logic functions and taps based on key. But the use of random numbers dose not supersedes probability of breaking it on the basis of the frequency test. Some research is also done on integration of several encryption algorithms. To be precisely if suggests use of a hybrid technique blending of both classical encryption technique as well as modern techniques to provide better security [9].

1.3 Contribution

This paper provides a new solution approach to overcome the shortcomings of the Playfair algorithm. Enhanced Binary Playfair Algorithm uses a reduced 4×4 key matrix to encrypt each byte of the plaintext file. Two nibbles of each byte are applied on the reference key matrix and a new pair of nibbles are obtained, which form the cipher byte. This algorithm also uses a dynamic matrix rearrangement to incorporate randomness in playfair instead of conventional poly alphabetic block cipher technique. The proposed Binary Playfair Algorithm is a stream cipher which uses dynamic byte substitution. This algorithm also is also efficient in respect to time complexity and space complexity and power consumption as it uses simple XOR and assignment operations. For these features this lightweight encryption algorithm can be used, where resources are limited in terms of- memory, computing time, computing power, battery supply, especially in the case of encryption/decryption in mobile device.

2 The Modified Binary Playfair Cipher

The new approach to overcome the limitations of the playfair cipher, which is discussed in this paper, treats each file as a binary file and applies the playfair algorithm on each byte of the file. The nibbles of each byte are used to encrypt / decrypt with the help of a reduced 4×4 reference key matrix. A nibble consists of 4 bits having a value of range 0-15. On the other hand, the 4×4 reference key matrix also contains 16 values, in the range 0-15. So, each pair of the nibbles of a byte is replaced with a new pair of nibbles and thus new byte is obtained. This algorithm encrypts / decrypts the file byte wise and hence the reminder offset or odd length word problem doesn't arise.

2.1 Algorithm

This algorithm consists of two phases: Key Matrix Formation and the main Encryption / Decryption phase.

Password / Key Matrix formation

- Step 1: Read the key file K_f .
- Step 2: XOR both nibbles of the byte and put the result value in a key-buffer.
- Step 3: If the XOR value already exists in the buffer, put the next value.
- Step 4: After putting the values if the buffer-size is less than 16 put the remaining values in the buffer, so that no repetition occurs.
- Step 5: Put the values of the key-buffer in the key-matrix in any specific arrangement order, based on the key value.

Encryption / Decryption

- Step 1: Read the plaintext file P_f .
- Step 2: Take a byte of the plaintext file to encrypt/decrypt.
- Step 3: Take two nibbles of each byte as reference and apply Playfair to get two resultant nibbles.
- Step 4: Combine these two nibbles to get the encrypted byte.
- Step 5: Write the byte in cipher text file C_f .
- Step 6: Rearrange the key matrix on the basis of plaintext value and present key matrix arrangement.
- Step 7: Repeat the steps 2-6 for next byte value until the plaintext is empty.

2.2 Step by Step Illustration for an Example String

Now to understand the algorithm more clearly, let us take a more detailed illustration of the algorithm.

- 1. Create Key-Buffer:** Let us assume that the key file contains text “IT(cwe)”.

Table 2. Creating unique key buffer

Key	ASCII	HEX-Code	Nibble 1	Nibble 2	XOR	Buffer
I	73	49	4	9	D (13)	13
T	84	54	5	4	1	1
(40	29	2	9	B (11)	11
c	99	63	6	3	5	5
w	101	65	6	5	3	3
e	119	77	7	7	0	0
)	41	30	3	0	3	4

Now, the remaining values (that are not already placed) are placed serially starting from 0. So the total key buffer is given below.

13 1 11 5 3 0 4 2 6 7 8 9 10 12 14 15

- 2. Key Matrix Arrangement:** Initially key matrix can be arranged in different variations. Some of the possible options are as follows.

Table 3. Some possible key arrangements

8	0	9	1
4	12	5	13
10	2	11	3
6	14	7	15

12	11	4	3
13	10	5	2
14	9	6	1
15	8	7	0

0	1	5	6
2	4	7	12
3	8	11	13
9	10	14	15

0	1	2	3
11	12	13	4
10	15	14	5
9	8	7	6

- 3. Encrypt / Decrypt:** After the arrangement of key-matrix the main encryption/ decryption is performed. Let us assume the first byte of the plaintext is K (HEX value: 4B). And the present state of Key Matrix is as follows.

Table 4. Key Matrix

8	0	9	1
5	12	5	13
10	2	11	3
6	14	7	15

The nibble values 4 & 11 of K are applied on the key matrix and the output nibbles are 5 & 10. So the output cipher is 5A which resembles Z.

4. **Key Matrix Rotation:** The size of the reference key matrix is reduced in this algorithm. So to prevent frequency based tracking this paper suggests a compulsory rotation / rearrangement of the reference key matrix.

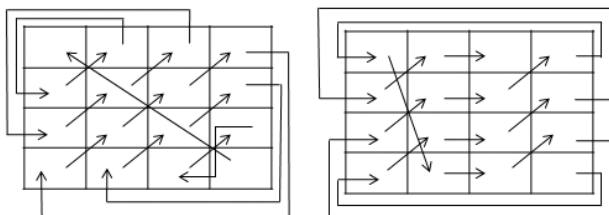


Fig. 1. Some possible key re-arrangements

2.3 Illustration with an Example File

To understand the algorithm more clearly, let us take an example and see its illustration. Let us assume the key / password is ➔ IT(cwe).

And the plaintext is ➔ *Switch to prudential home insurance and you could get a "25%" introductory discount for the first year of your policy.*

1. **Create Key-Buffer:** According to Table 2 of the previous example, following key buffer is formed from the password IT(cwe).

13 1 11 5 3 0 4 2 6 7 8 9 10 12 14 15

2. **Key Matrix Arrangement:** Initially key matrix is arranged as follows.

Table 5. Key Matrix

9	12	14	15
4	6	7	8
0	3	11	2
13	1	10	5

3. **Encryption/Decryption:** The first byte of the plaintext is S (HEX value: 53). The nibble values 5 & 3 of S are applied on the key matrix and the output nibbles are 1 & 2. So the output cipher is of HEX value 12 which resembles DC2 (Device Conctol 2).

4. **Matrix Rotation:** Based on the key matrix of Table-5 and previous plaintext character S (Ascii value: 83) the matrix is rotated. So the current key-matrix is as follows.

Table 6. Key Matrix

3	9	12	14
0	4	6	15
13	11	7	8
1	10	5	2

5. **Encryption/Decryption:** The second byte of the plaintext is *w* (HEX value: 77). The nibble values 7 & 7 of *w* are applied on the key matrix and the output nibbles are 8 & 8. So the output cipher is of HEX value 88 which resembles the symbol '?’.

These two steps of Encryption/Decryption and Matrix Rotation will be repeated for each bytes of the plaintext / ciphertext. For the plaintext of this example it will be repeated for 118-times.

And the corresponding cipher is ➔ *ι?\$.:-B??p9H?#øÖB?äç@æι??}ιιRι
¼úÖiÇ©|é?ι4ÔÊ?¬v?¬?ι©α-;“?qÑªQδEιø[ÆÛv?;P£b]Æ+É~;Öbιw!ι
æAđo tä5 ùL_uUâX?öÂ\¶§¬ιk*

3 Experimental Results

The proposed Enhanced Binary Playfair algorithm is implemented on java platform [11] and a number of tests are considered to observe the encryption efficiency in terms of certain parameters like avalanche effect, key randomness and time. Two more algorithms DES (Data Encryption Standard) [12] and DNA-Playfair [7] algorithms are considered for comparative encryption analysis with the proposed algorithm. In respect of the above mentioned parameters some experimental results are given below.

3.1 Avalanche Effect

The avalanche effect refers to a desirable property of cryptographic algorithms. The avalanche effect is evident if, when an input is changed slightly (for example, flipping a single bit) the output changes significantly (e.g. - 35% of the output bits flip).

In the proposed enhancement of playfair the avalanche effect is tested on a number of randomly generated 100 files. The result of Avalanche Effect lies between 46% - 55%. The ideal case or strict avalanche criterion (SAC) is the probability of 50% bits change. So the obtained result proves the efficiency of the algorithm very strongly.

Though in this algorithm the type or variety of rotation is fully dependent on 'key' and on the 'content of the plaintext file', the result may vary for different files.

A details comparison of Avalanche effect between different algorithms is given in the chart below.

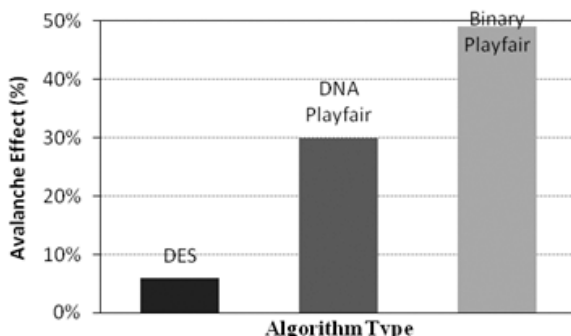


Fig. 2. Avalanche Effect of different algorithms

Figure 2 shows the avalanche effect of different encryption algorithms. It can be observed that the proposed Binary Playfair algorithm gives avalanche effect of almost 49% change, which satisfies nearly the strict avalanche criteria of 50%. Thus, it can be inferred that the Enhanced Binary Playfair algorithm gives comparatively much better result than other two encryption techniques DES and DNA Playfair which show 6% and 30% avalanche effects respectively.

3.2 Repetitions of Characters

The chance of repetition of a character is very low. So, it is safe from frequency tracking. The repetition of a character in the process of encryption of a 16 KB file on its every occurrence a particular byte “#” with hexadecimal value of 23 is encrypted as follows.

37	1F	6C	68	3A	50	C1	DF	0D	9D	95	BD	03	A1
C7	35	01	DB	1B	42	1C	10	84	75	B2	CE	F9	74
35	52	06	7D	37	58	D6	8D	C4	B2	1B	37	DF	BA
FC	71	8F	78	52	05	3D	E2	6B	DC	F6	30		

The above box shows that it is almost impossible to break the cipher using frequency of the character occurrence testing.

A frequency count is also computed on the cipher key matrix to analyze what are the most and least common character occurrences in the cipher. Experimental analysis for character repetition has been conducted on a large set of files of varying sizes and

Table 7. Comparison of Character Occurrences

File Size	Algorithm	Percentage of Repetition
4 KB	Binary Playfair	15 %
	DES	24%
	DNA Playfair	40%
12KB	Binary Playfair	29%
	DES	32%
	DNA Playfair	79%

the proposed algorithm has been compared with DNA Playfair and DES. Two sizes of file, 4 KB and 12 KB are chosen. The percentage values are obtained from the average of large set of small size and large size files. The results are shown as below.

It is observed in Table 7, that for small size file, the frequency tracking of the proposed algorithm is almost difficult and byte repetition is around 15% in comparison with DNA Playfair and DES of values 24% and 40% respectively. Although for large file size, the chances of byte repetition is likely to be more and hence percentage of repetition gets higher. Still Binary Playfair is proved to be more lightweight and efficient as claimed in our algorithm in comparison to DNA Playfair and DES algorithm.

3.3 Random Rotations

After encryption of each byte the matrix can be rotated in $n!$ (i.e. 2092278988800) where $n = \text{row} \times \text{column} = 16$) number of different ways. Among these rotations, in some cases repetitions may occur in the cipher. If row and col represents the number of rows and columns of the key matrix, and N_r represents possible number of unique rearrangements of the matrix. Then in each case of rotation/rearrangement, the actual number of variations where no repetition will occur can be expressed as follows.

$$N_r = n! - [({}^{\text{row}}C_2 \times 2!) \times ({}^{\text{col}}C_2 \times 2!)] \times (n - 4)! \quad (1)$$

In case of this modified version of playfair a 4×4 key matrix is used. So, $\text{row}=4$, $\text{col}=4$, $n = (\text{row} \times \text{col}) = 16$

$$N_r = 16! - [({}^4C_2 \times 2!) \times ({}^4C_2 \times 2!)] \times (16 - 4)! = 43536 \quad (2)$$

This 43536 is a huge number of variations, for a cryptanalyst to search in each byte of the cipher in order to break it. Hence the cipher is secured.

3.4 Low Space Requirement

The space requirement of this modified Playfair algorithm is very low. It uses a 4×4 matrix. Only a buffer of 16 nibble is required. In addition, two bytes are required one for file read/write buffer and the other for matrix manipulation buffer.

3.5 Low Time Requirement

The time requirement for encryption of this modified Playfair algorithm is substantially low. The proposed algorithm has been applied randomly on a set of 100 files of different in order to calculate the average time requirement for encryption / decryption.

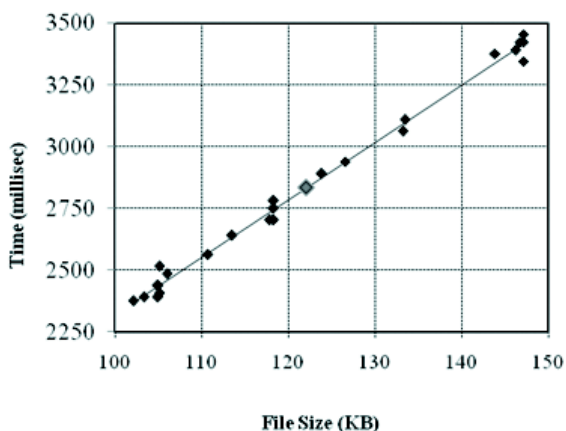


Fig. 3. Average Encryption / Decryption Time

In Figure 3, the average time is depicted to encrypt 24 files of size ranging from 100 KB to 150 KB. It is observed that on an average 2 sec 832 millisecond time is required to encrypt a file of size 122 KB. As a matter of fact, it can be well inferred that average time required to encrypt increased number of large size files will be more.

A detailed comparison of different algorithms regarding time requirement to encrypt / decrypt a file of size 103 KB is given in the chart below.

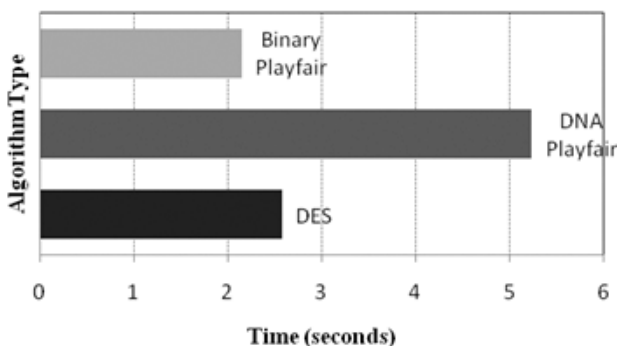


Fig. 4. Time requirement versus different encryption algorithms

Figure 4 depict the average time taken by different algorithms to encrypt files of size 100 KB. A large number of randomly generated text and binary files of size 100 KB was used in the experimentation. It can be clearly observed that the proposed Enhanced Binary Playfair algorithm is taking least time of 2.150 seconds to encrypt the file. For other algorithms like DES and DNA Playfair, the time requirement for encrypting the same file are more, which are around 2.580 seconds and 5.234 seconds respectively.

4 Conclusion and Future Work

The original Playfair cipher uses a digraph substitution technique to encrypt/decrypt alphabets based on a reference 5×5 key matrix which is formed from the given key. The algorithm is strictly restricted to “English Alphabet”, that to either in uppercase or in lowercase character. No numbers, punctuations and other characters are supported. Several modification attempts have focused on elimination of several limitations. Some methods have increased the character-set of the key matrix, others have used the ASCII values and others have incorporated randomness. But these modifications stand strong in their own purposes. The overall limitations of the cipher were not eliminated by these individual modifications.

This paper proposes a modification of the Playfair algorithm which strongly increases the security of the cipher. The proposed algorithm uses the 8-bit ASCII values of the plaintext file to encrypt/decrypt in the binary mode. Thus it supports any type of plaintext be it any alphabet of any language, any symbol, any number system, any type of media file or anything else. This algorithm acts as a stream cipher rather than conventional poly alphabetic block cipher. There is no need to adjust reminder offset or odd length word. Randomness is incorporated by means of random specific rearrangement of key matrix. The algorithm uses a 4×4 key matrix. So space efficiency is achieved. It uses simple XOR, shift and assignment operations straightway. The time complexity of the algorithm is $O(n)$, where n is the file size. For these features, this lightweight encryption algorithm can be used, where security requirement is high but resources are limited in terms of memory, computing time, computing power, battery supply.

The future work can focus on a larger key matrix which might enhance the security further and also reduce the time complexity. An additional indexing can be introduced to sort out the order and total number of rearrangement techniques based on the key / password. This algorithm can further be implemented in chip level to embed it in mobile sensors networks.

Acknowledgement. Authors acknowledge UPE-II program of Jadavpur University for partially supporting this work.

References

1. Playfair Cipher, http://en.wikipedia.org/wiki/Playfair_cipher (October 21, 2011)
2. Ravindra Babu, K., Uday Kumar, S., Vinay Babu, A., Aditya, I.V.N.S., Komuraiah, P.: An Extension to Traditional Playfair Cryptographic Method. *International Journal of Computer Application* 17(5), 34–37 (2011)

3. Srivastava, S.S., Gupta, N.: Security aspects of the Extended Playfair cipher. In: International Conference on Communication Systems and Network Technologies, pp. 144–147 (2011)
4. Umakanta Sastry, V., Ravi Shankar, N., Durga Bhavani, S.: A Modified Playfair Cipher Involving Interweaving and Iteration. *International Journal of Computer Theory and Engineering* 1(5), 597–601 (2009)
5. Umakanta Sastry, V., Ravi Shankar, N., Durga Bhavani, S.: A Modified Playfair for a Large Block of Plaintext. *International Journal of Computer Theory and Engineering* 1(5), 592–596 (2009)
6. Ravindra Babu, K., Udaya Kumar, S., Vinaya Babu, A., Reddy, T.: A Block Cipher Generation Using Color Substitution. *International Journal of Computer Applications* (0975 - 8887) 1(28), 25–27 (2010)
7. Sabry, M., Hashem, M., Nazmy, T., Khalifa, M.E.: A DNA and Amino Acids-Based Implementation of Playfair Cipher. (IJCSIS) *International Journal of Computer Science and Information Security* 8(3), 129–136 (2010)
8. Murali, P., Senthilkumar, G.: Modified Version of Playfair Cipher using Linear Feedback Shift Register. In: 2009 International Conference on Information Management and Engineering, pp. 488–490 (2009)
9. Saeed, F., Rashid, M.: Integrating Classical Encryption with Modern Technique. *IJCSNS International Journal of Computer Science and Network Security* 10(5), 280–285 (2010)
10. Mondal, U.K., Mandal, S.N., PalChoudhury, J.: A Framework for the Development Playfair Cipher Considering Probability of Occurrence of Characters in English Literature. 2009 *International Journal of Computer Science and Network Security* 8(8) (August 2008)
11. Bishop, D.: Introduction to cryptography with Java applets (2005)
12. Stallings, W.: *Cryptography and network security: principles and practice*, 5th edn. Prentice Hall International (2010)

Tuning of a Knowledge-Driven Harmonization Model for Tonal Music

Mariusz Rybniak¹ and Wladyslaw Homenda²

¹ Faculty of Mathematics and Computer Science, University of Białystok,
ul. Sosnowa 64, 15-887, Białystok, Poland

`mariuszrybniak@wp.pl`

² Faculty of Mathematics and Information Science, Warsaw University of Technology,
Pl. Politechniki 1, 00-661, Warsaw, Poland

`homenda@mini.pw.edu.pl`

Abstract. The paper presents and discusses direct and indirect tuning of a knowledge-driven harmonization model for tonal music. Automatic harmonization is a data analysis problem: an algorithm processes a music notation document and generates specific meta-data (harmonic functions). The proposed model could be seen as an Expert System with manually selected weights, based largely on the music theory. It emphasizes universality - a possibility of obtaining varied but controllable harmonies. It is directly tunable by changing the internal parameters of harmonization mechanisms, as well as an importance weight corresponding to each mechanism. The authors propose also indirect model tuning, using supervised learning with a preselected set of examples. Indirect tuning algorithms are evaluated experimentally and discussed. The proposed harmonization model is prone both to direct (expert-based) and indirect (data-driven) modifications, what allows for a mixed learning and relatively easy interpretation of internal knowledge.

Keywords: Harmonization, data analysis, expert system, musical work, supervised learning, tonal music.

1 Introduction

Harmony is an important element of tonal music, it defines a vertical relation between notes [1], and by definition is opposed to *melody* - a horizontal succession of notes in a specific voice. In fact, however, the harmonic relations of the leading melody (regarding mono- or homophony) are largely depending on the horizontal succession of notes and *melodic intervals* between them. Harmonic *passages* that follow harmonic *chords* (with smaller or larger deviations) can be frequently detected in a melody. Similar, but obviously much stronger harmonic relations are to be found in an *accompaniment*, where the melody is less (or even not at all) important, as the main goal is to define a background for the leading melody. The obvious exception to these assumptions are polyphonic musical works. They tend to cultivate two or more independent voices, that compete for attention,

but also have to cooperate harmonically, usually indicating strong harmonic relations.

Automatic harmonization can be seen as a problem crossing two different areas: 1) theoretical music knowledge and 2) formal mathematical computations - presented in form of algorithms or slightly less formal Artificial Intelligence solutions. From the technical point of view it could be seen as analyzing of a music notation document and producing meta-data (harmonic functions). Over the years, many various approaches and techniques were used to solve the problem (or similar). The most popular paradigms are: Expert Systems [2][3], Neural Networks [4], Constraints and probabilistic approaches [5][6][7], evolutionary algorithms [8]. Due to excessive complication and uncertainty, related to such sensitive subject as music (feelings and sensitivity still tend to evade scientific approaches), no approach seems to fully explore and describe the subject. In fact, some of the approaches focus on a particular style of musical works, as for example typically Baroque pieces of J. S. Bach in [4]. This allows for relatively narrow specialization and therefore eases the algorithmic description of the problem.

Frequent limitations for above-mentioned approaches seem to be: lack of universality (results limited to a specific area), an unpredictability, a need for a large learning examples database and an extensive amount of calculations. The authors' approach is mostly based on an advanced theoretical music knowledge, especially in the area of harmonization, and is aimed at solving these disadvantages. The authors propose a model that may be customized at various levels and does not need any learning examples, as they are replaced by an expert knowledge of harmony, incorporated in model. It may be defined initially as Expert System with manually selected weights, however the authors also propose a scheme for tuning the general harmonization model. With enough knowledge of *harmony* one can also directly modify the model using internal parameters.

In this paper the authors concentrate on tuning procedures of their harmonization model introduced in [9] and extended in [10]. The paper is organized into five sections; Section 1 being an introduction. Section 2 presents basic concepts concerning harmony and harmonization in tonal music. Section 3 describes the harmonization model and explains mechanisms used. Section 4 discusses various tuning possibilities of a proposed model and presents a sample of experimental results. Finally, Section 5 concludes the paper, discusses configuration, universality properties and suggests future work.

2 On Harmony in Tonal Music

The process of creating accompaniment to a *homophony* (lone melody - a single-layered progression of sounds in time) can, in practice, be divided into two phases. The first is *harmonization* (determination of harmonic functions and chords corresponding to them), the second phase is creation of accompaniment using previously determined chords. It is important to stress, that in vast number of cases there is no single, ultimate harmonization for a given melody. There

are many possibilities, from the most simple and obvious to the highly complicated and non-trivial ones (e.g. improvised music, jazz). The decision depends mostly on a style of music. It also relies on capabilities of the available instrument/orchestration, and the abilities of performers/harmonizers. There are also cases when no additional information is needed to perform the music, except for the leading melody and harmonic functions. A common example is an improvised *à vista* accompaniment to songs played on a guitar or a piano.

2.1 The Tonal Harmony

For the purposes of this paper the authors have focused on a *tonal system* with two basic *scales*: *major* and *minor* (*natural minor* as well as *harmonic* and *melodic* modifications). The authors' considerations and experiments are based on seven diatonic *harmonic functions* (built on 1st, 2nd, 3rd, 4th, 5th, 6th and 7th grade of major/minor scales) with common modifications: adding the *seventh*, *ninth*, and *sixth* (all three minor and major).

The harmonic relations in tonal system tend to be well defined in terms of *consonants* (chords 'pleasant' to human ear, stable) and *dissonances* (chord slightly 'unpleasant', unstable, introducing a tension that needs to be *resolved* to consonants). The authors have omitted alterations and higher intervals (alike *eleventh*, *thirteenth*), that may be considered as *dissonances* meant to *resolve*, rather than intrinsic harmony.

3 Proposed Harmonization Model

The proposed harmonization model is based on several mechanisms that closely follow music theory:

1. **Particular note can have various harmonic importance** (based on the note's relative length, the notes placement in measure, surrounding notes, volume, etc.);
2. **Each note excites (fits to) several harmonic functions**, based on pitch and function components taken into consideration (components higher than 9th are usually very rare and considering them is very difficult);
3. **Some harmonic functions are more likely to occur than others** (the simplest example being *Tonic* - a base and consolation for vast majority of tonal music, therefore usually occurring most frequently). It is preferable to prioritize the commonly used functions;
4. **Some specific successions of harmonic functions are more or less probable**, therefore it is possible to prioritize the more likeable (frequent) successions (e.g. *Dominant* → *Tonic* or *Tonic* → *Subdominant*).

The mechanisms are implemented in an independent way (when possible) and weighted using a standard range [0; 1], in order to easily configure (or eliminate) their degree of influence.

3.1 Data Representation of Harmonic Functions

Each harmonic function is stored in a structure that contains a vector of *Harmonic Function Strengths* (since we consider 7 different diatonic functions its length is 7) and a corresponding vector of function modifiers. *Harmonic Function Strength* is a factor used to determine probability of occurrence of a specific diatonic harmonic function. In the authors' experiments the function modifiers are mostly limited to *sevenths*, with occasional *ninths* and in rare cases to *sixths*. *Sixth* is present in so-called *Chopin's chord: Dominant* with *seventh* and natural or augmented *sixth* instead of *fifth*, resolving down to the first degree of scale. The diatonic function with modifiers in a given *key* is equivalent to chords e.g. *Tonic* with modifier *seventh* in key G-major is equivalent to G^7 . Some modifiers cover others, e.g. a chord with a *ninth* is a variation of a chord with a *seventh*. Modifiers can be seen as subclasses of the main class (basic diatonic function).

For simplification reasons the authors have decided to determine the harmonic functions in constant intervals: twice per measure in the case of 3 or 4 beats and once per measure in the case of 2 beats. This is sufficient in most cases, as in practice the harmonic functions rarely change more frequently. The authors have also experimented with harmonic fragment being equal to the whole measure. Musical piece is decomposed into *harmonic fragments* (defined as indivisible musical unit with a single diatonic function attached to each one of them). The *Harmonic Function Strength* for a *harmonic fragment* is a sum of *function excitations* (determined by note pitch) for all notes in the fragment, according to varied degrees of *Note Importance*.

3.2 The Flow of the Proposed Model

The flow of the proposed model regarding a single *harmonic fragment* is presented in Fig. 1 and proceeds as follows:

1. The harmony of a particular fragment is evaluated by examining all notes that it is composed of. Based on a set of rules from Section 3.3, every note is attached with a *Note Importance* value.
2. Each note excite (add a certain value to *Function Strengths* vector) several corresponding diatonic functions by being a specific chord component, with degree defined by the *Function Excitation Matrix* (Section 3.4).
3. After all the notes are processed, the vector of *Function Strengths* is elementwise multiplied by a *Function Popularity Vector* (Section 3.5). This primarily serves as a method of favouring the more popular functions. The *Function Popularity Vector* can also be a way to get a rich and uncommon harmony.
4. A *Function Successions* mechanism is applied after the individual processing of all of the harmonic fragments is complete. *Function Succession* matrix (Section 3.6) defines the degrees of desire for a specific succession of harmonic functions. They serve as additional modifiers (regarding direct predecessor and successor) for vectors of the *Function Strengths*. Every fragment is modified twice, first as a predecessor and the second time as a successor; only

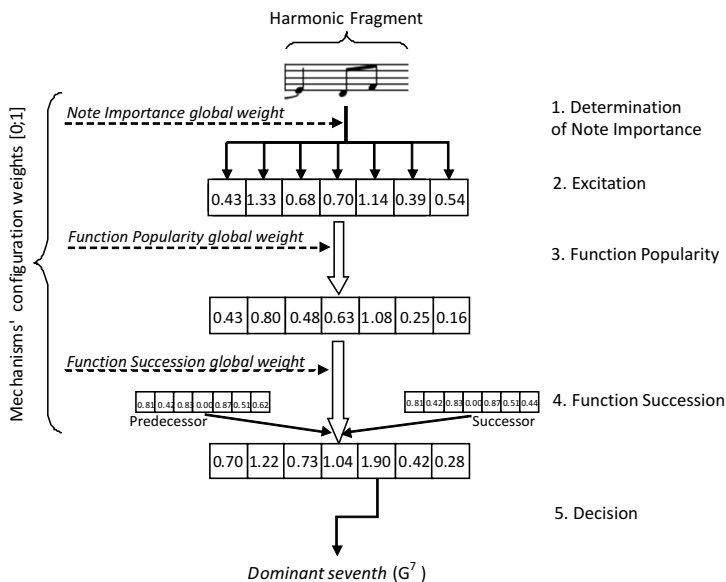


Fig. 1. Harmonization model flow

the outmost fragments are changed once. The modifications are calculated for every possible function succession (a total of 49 combinations is possible for 7 recognized functions, the modifiers are not taken into account). They are multiplied by both corresponding function strengths and additionally by a 1/12 factor what potentially keeps them in the original *Function Strengths* range.

5. A classification decision is made for each *Harmonic Fragment*. It is based on values of *Function Strengths*. *Winner-takes-all* decision was used for simplicity reasons; the classification-like relations are already hidden inside the above mentioned mechanisms. It is important to stress that any classifier can be used for this purpose (including direct use of function subclasses, described in Section 3.1). Determination of classifier for this purpose is a large topic exceeding the scope of the paper. The winning *diatonic function* is translated into *chord*, regarding the *key*. Finally it is attached with chord modifiers (e.g. *seventh*, *ninth*), that were stored independently during *Excitation*.

Each omissible mechanism (*Note Importance*, *Function Popularity*, *Function Succession*) is attached with an overall *Global Configuration Weight* from the range [0;1] (Fig. 1). It can be used to fine tune or disable the influence of a specific mechanism on the final harmonization result. *Excitation* is an essential mechanism to obtaining *Function Strengths*, therefore it may not be weighted in a similar manner. The mechanisms are based on matrices (*Function Excitation*, *Function Succession*), vectors (*Function Popularity*) or rules (*Note Importance*)

that offer substantial, direct and indirect configuration possibilities. The following subsections describe in detail the above-mentioned mechanisms.

3.3 Note Importance Determination

Note Importance may be determined by:

- **the length of note** - longer notes have, in general, greater influence on the harmony. The note length is used as an initial value for further importance weighting. Values corresponding to neighboring note lengths (alike *eighth* note and *quarter* note) have common ratio, empirically stated as 5/6.
- **position in measure** - notes at inherently accented parts of measure - *on-beat* - are very important. Notes at generally unaccentuated parts of measure - *off-beat*, are less significant. Notes occurring in-between these main beats are harmonically even less important.
- **notes that are at the end of ties have minor contribution to the harmony**; when played on string instruments (e.g. grand piano, upright piano, guitar) they are not hit again. They are therefore relatively quiet.
- **notes that are easily-heard by human ear are placed in extreme voices** (highest and lowest notes); notes in the middle voices are slightly harder to hear and therefore contribute less to the local harmony;
- **accentuation increases the volume** and therefore harmonic value of the note.

Homophony harmonization was chosen for the experimental evaluation, therefore the authors have determined a set of *Note Importance* rules described in Table 1. In this case, rules regarding voice position are irrelevant, and are not taken into consideration. These rules would be however important for music works with many *voices*. Corresponding *note length* from upper part of the Table 1 is taken as an initial value of *Note Importance*. Initial values for non-standard note lengths (e.g. a half note with a dot) are determined proportionally. The weight corresponding to the first fitting rule from lower part of the Table 1 serves as the multiplier for the initial value.

3.4 Implementation of Functions Excitation by Notes

A well-defined *diatonic function* contains at minimum three components (typically *root note*, *third* and *fifth*). More complicated diatonic functions contain also additional or altered intervals. Determination of exact *pitches* of these function components requires detection of the piece *key* and reading of the current *key signature*.

Two matrices of *Function Excitation* are used to define the excitations of diatonic functions with notes (from the melody or the accompaniment). The matrix for major scales is presented in Table 2. Each diatonic function is excited by the particular matching note to a specified degree. This, obviously, is relative to the pitch of the note in question, and to the pitches of the notes occurring in the tonal functions themselves. The authors have defined the weights of excitation in the range of $[0, 1]$, assigning:

Table 1. Determining Note Importance

Note length	Initial value
Whole note	1.440
Half note	1.200
Quarter note	1.000
Eighth note	0.833
Sixteenth note	0.694
Thirteenth-second note	0.579
Condition	Importance weight
Note at the end of a tie	0.4
Note on the 1 st beat	1.0
Note on the 3 rd beat	0.9
Accentuated note	0.8
Note on the 2 nd or 4 th beat	0.7
Other notes	0.6

- [0.8; 1.0] - greater degrees to the common chord components (e.g. *root note*, *third* and *fifth*);
- [0.4; 0.6] - moderate degrees to the less common components: *sixth*, *seventh*, *ninth* with occasional modification of natural *third*. Making major chords from naturally minor chords using augmented *third* serves frequently as *tonicization* - a local *Dominant* → *Tonic* resolution);
- [0.1; 0.2] - small degrees to the rare modifications of natural *sixth seventh* and *ninth*.

Table 2. Excitation weights for major scales

	Absolute shift from function <i>root note</i> [in semitones]											
	0	1	2	3	4	5	6	7	8	9	10	11
<i>D</i> ₇	1	0	0	0.9	0	0	0.8	0	0	0	0	0
<i>T</i> ₆	1	0	0	0.9	0.1	0.1	0	0.8	0	0	0.4	0
<i>D</i>	1	0.5	0.2	0.1	0.9	0.1	0	0.8	0	0.4	0.6	0.1
<i>S</i>	1	0	0	0.1	0.9	0.1	0	0.8	0	0	0.4	0
<i>T</i> ₃	1	0	0	0.9	0.1	0.1	0	0.8	0	0	0.4	0
<i>S</i> ₂	1	0	0	0.9	0.1	0.1	0	0.8	0	0	0.4	0
<i>T</i>	1	0	0	0.1	0.9	0.1	0	0.8	0	0	0.4	0

A similar matrix has been prepared for minor scales. Due to space limitation it is not presented in this paper. The obvious differences in reference to major scales are the natural qualities of diatonic functions. The less straightforward dissimilarity is a frequent conversion from naturally minor *Subdominant* and *Dominant* to major. This commonly occurs in a *melodic* variation of scale, and sometimes, in *harmonic* variations.

3.5 Function Popularity

The goal of determining and applying weights corresponding to *Function Popularity* is to directly prioritize frequently occurring tonal functions (e.g. *Tonic*, *Subdominant*, *Dominant*). This is a direct way to make them more frequent than less popular functions, as inherently specified by music theory. The goal can be indirectly obtainable by using lower coefficients in the *Functions Succession* matrix (described in Section 3.6). Obviously, the direct method makes controlling the process much easier. For experimental studies the authors have used the *Function Popularity* vector, specified in Table 3.

Table 3. Function Popularity vector weights

Tonal function	
Name	Weight
T	1.0
S_2	0.6
T_3	0.7
S	0.9
D	0.95
T_6	0.65
D_7	0.2

3.6 Succession of Harmonic Functions

The succession of harmonic functions (horizontal relations between neighboring harmonic functions) is implemented using encouragement of more probable combinations, with moderation of less likable ones. This is done using a *Function Succession* matrix. The matrices proposed for major and minor scales are presented in Table 4. The matrix for minor scale is similar and due to space limitations not presented here.

The matrices values were determined in order to prioritize the most likable successions. The exemplary common successions are: *Tonic* into *Subdominant*, *Subdominant* into *Dominant* and *Dominant* into *Tonic* - cadence (the simplest and most common). Another example is less common *deceptive cadence* (*Dominant* into *Tonic*₆). The matrices also support *tonicization* (e.g. succession from *Subdominant*₂ into *Dominant*). It is worth mentioning that the most supported quasi-succession is maintaining the current function (no change) with the maximum degree of support: 1.0. It allows more efficient handling of common cases, where harmonic functions change less frequently than the arbitrary *harmonic fragment* length (e.g. two beats). It is also important to mention that the support of succession occurs only between harmonically determinable *harmonic fragments*, excluding these that do not contain notes at all (only pauses), or contain only ongoing tied notes from previous beats. It is assumed that such fragments continue the previous harmonic function (which is a slight oversimplification as

Table 4. *Functions Succession* weights for major scales

	Transfer into:						
	T	S_2	T_3	S	D	T_6	D_7
D_7	0.5	0.3	0.5	0.2	0.9	0.1	1.0
T_6	0.4	0.8	0.3	0.8	0.7	1.0	0.2
D	0.9	0.4	0.7	0.3	1.0	0.8	0.3
S	0.7	0.4	0.4	1.0	0.9	0.5	0.1
T_3	0.3	0.2	1.0	0.5	0.6	0.8	0.1
S_2	0.2	1.0	0.2	0.5	0.9	0.4	0.2
T	1.0	0.4	0.3	0.8	0.9	0.6	0.1

it does not have to be always true). The succession support occurs forwards and backwards for each possible succession, with a degree defined by the sum of products of the neighboring *Function Strengths* and a succession weight from the *Functions Succession* matrix.

4 Model Tuning and Experimental Results

The authors have implemented the proposed model and tested it using music documents in MusicXML file format [11]. This section discusses various tuning possibilities and presents exemplary homophony harmonizations with the use of various tuning approaches.

The model has been applied to musical works containing a single melody (*monophony*) with no *key* changes (no modulations). It may be considered as a greater challenge (more harmonic uncertainty) than using musical works with accompaniment or several independent voices, where harmonic functions are generally easier to detect. The proposed harmonization model is viable for almost every musical piece, regardless of the number of voices, as long as it is maintained in a tonal system, e.g. uses either major or minor scale (with possible modifications like harmonic, melodic, Dorian, etc.). In case of many voices or chords, ideally, a customized *Notes Importance* determination is required (as described in Section 3.3), in order to detect and prioritize more important voices, and attenuate the less important ones. Efficient harmonization of a musical works with *key* changes (*modulations*) requires detection or indication of such changes, and an adequate update of tonal root and/or minor/major scale properties.

The authors propose the following model modifications:

1. **changing values of *Global Configuration Weights*** (respectively: *Note Importance*, *Function Popularity*, *Function Succession*), exemplary applications are: moderation of less popular functions, introduction of uncommon function successions;
2. **direct tuning** - changing values in the configuration matrices (*Function Excitation* matrix, *Function Popularity* vector, *Function Succession* matrix) and defining custom *Note Importance* determination rules. The initial values are meant to be universal, changing them directly influences the behavior of

the model and produced results in a specific direction, but requires experience and advanced knowledge of harmony;

- 3. **indirect tuning by learning from *examples*** - modification of a part of proposed approach (*Function Popularity* vector and *Function Succession* matrix) using non-direct modification (tuning) of matrices. The *example* is a harmonized music piece: a music document with attached harmony functions corresponding to class labels. A simple statistical analysis of the occurring harmonic functions are used to update the values in the *Function Popularity* vector. Analysis of occurring harmonic function successions in the *example* are used to update *Function Succession* matrix. Total extinction of rare functions and successions may occur (depending on the harmony relations included in the examples), therefore the authors propose to limit the minimal values to a fixed threshold 0.05. The learning process is independent regarding the sequence of examples - similarly to *batch learning*, as *sequential learning* would prioritize the recent examples over the previous ones.

The results in Fig. 2 present alternative harmonizations produced using different tuning procedures. The fragment of musical piece (A. De Vita and H. Sharper - *Softly, as I leave you*) was harmonized with *harmonic fragments* set to 2 beats (half of measure). The first and second harmonies were obtained for original model with various values of *Global Configuration Weights* (respectively: *Note Importance*, *Function Popularity*, *Function Succession*): {0.4, 0.4, 0.4} in the first case, {0, 0, 0.8} in the second case. In the second case less popular functions are encouraged to appear. Third harmony was obtained by using *direct tuning of the model* (by changing *Function Excitation* matrix to produce more seventh and ninth chords, and *Function Popularity* vector to encourage rare functions. Fourth harmony was obtained by using *indirect tuning by learning from examples*, with ten examples being modern popular music pieces of unspecified genre. The learning set has relatively simple harmonies, therefore the harmonization results (produced by the tuned model) are also simplified.

Regarding indirect tuning: it is possible to tune the original model into a specific direction by training with a preselected set of *examples* (i.e. jazz pieces with complicated harmony or musical pieces from a specific period). Such approach requires a number of carefully selected examples and is problematic as musical styles are often ambiguous.

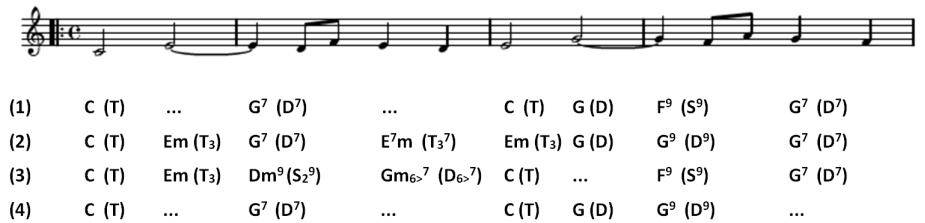


Fig. 2. Alternative harmonizations using different tuning procedures

5 Conclusion and Future Work

In this paper the authors have presented music theory based harmonization model with emphasis on model tuning. The main goal of the approach is universality but also control over harmonization process and results. Various possibilities of the initial model modifications are discussed with given examples.

5.1 Remarks on Universality

The proposed model offers numerous configuration possibilities and universality: as it is not data-driven but knowledge-driven, it provides multiple levels of model control and, to some extent, results of harmonization. As opposed to the data-driven approaches (taught with examples) the authors' methodology does not need to be fed with large or representative sample of data (requiring data gathering, selection and frequently producing uncontrollable results). It may be used in many variants relying only on the theoretical knowledge. With modifications of underlying configuration data one can achieve many valuable tasks, such as:

- defining the *Note Importance* rules tuned to various musical pieces;
- defining the *Function Excitation* matrix to generate rare or simple functions/modifications, resulting in a complicated or simplistic harmony;
- defining the *Function Popularity* vector to moderate, forbid or encourage specific tonal functions, directly limiting or increasing probability of their occurrence;
- defining the *Function Succession* matrix to moderate, forbid or encourage specific harmonic function successions;
- changing the *Global Configuration Weights* of harmonization mechanisms in order to tune the model to the specific needs and expectations;
- iterate through configurations in order to quickly generate various (possibly interesting) harmonies for the same musical work.

The authors propose also data-driven tuning of the initial model, what could provide more viable harmonizations or (with preselected set of examples) produce specific harmony style.

5.2 Future Works

Future works in the area will be conducted in the following directions:

- continuous development and evaluation of the presented harmonization model;
- determination of configuration matrices and parameters for different styles of musical works (like jazz, classical music, popular music, etc.) using indirect tuning with preselected set of examples (what requires relatively large and representative examples database);

- development of the *Function Succession* mechanism with use of chord modifiers;
- development of the *Function Excitation* mechanism using also relations between notes, rather than independent notes;
- developing criteria for automatic evaluation of the obtained harmony;
- automatic parametrization of the harmonization model based on the above-mentioned criteria;
- further development of model tuning procedures, using well established machine learning paradigms, e.g. Artificial Neural Networks.

Acknowledgement. This work is supported by The National Center for Research and Development, Grant no. N R02 0019 06/2009.

References

1. Sikorski, K.: Harmony. Polskie Wydawnictwo Muzyczne (2003) (in Polish)
2. Cope, D.: An expert system for computer-assisted music composition. *Computer Music Journal* 11(4), 30–46 (1987)
3. Ebcioglu, K.: An expert system for harmonizing four-part chorales. In: *Machine Models of Music*, pp. 385–401. MIT Press (1993)
4. Hild, H., Feulner, J.M.W.: Harmonet: A neural net for harmonizing chorals in the style of J.S. Bach. In: *Advances in Neural Information Processing 4* (1992)
5. Pachet, F., Roy, P.: Musical harmonization with constraints: A survey. *Constraints Journal* 6(1), 7–19 (2001)
6. Pachet, F., Roy, P.: Mixing constraints and objects: a case study in automatic harmonization. In: *TOOLS Europe 1995*, pp. 119–126. Prentice-Hall (1995)
7. Paiement, J.-F., Eck, D., Bengio, S.: Probabilistic Melodic Harmonization. In: Lamontagne, L., Marchand, M. (eds.) *Canadian AI 2006. LNCS (LNAI)*, vol. 4013, pp. 218–229. Springer, Heidelberg (2006)
8. De Prisco, R., Zaccagnino, R.: An Evolutionary Music Composer Algorithm for Bass Harmonization. In: Giacobini, M., Brabazon, A., Cagnoni, S., Di Caro, G.A., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Fink, A., Machado, P. (eds.) *EvoWorkshops 2009. LNCS*, vol. 5484, pp. 567–572. Springer, Heidelberg (2009)
9. Rybnik, M., Homenda, W.: Knowledge-driven Harmonization Model for Tonal Music. In: *Proceedings of the 4th International Conference on Agents and Artificial Intelligence (ICAART 2012)*, pp. 445–450 (2012)
10. Rybnik, M., Homenda, W.: Extension of Knowledge-driven Harmonization Model for Tonal Music. In: *2012 International Joint Conference on Neural Networks (IJCNN 2012)*, Brisbane, Australia (2012)
11. Good, M.: MusicXML for Notation and Analysis. In: Hewlett, W.B., Selfridge-Field, E. (eds.) *The Virtual Score: Representation, Retrieval, Restoration*, pp. 113–124. The MIT Press (2001)

Efficient Processing the Braille Music Notation

Tomasz Sitarek and Wladyslaw Homenda

Faculty of Mathematics and Information Science
Warsaw University of Technology
Plac Politechniki 1, 00-660 Warsaw, Poland
Tomasz.Sitarek@ibspan.waw.pl
<http://www.mini.pw.edu.pl/~homenda>

Abstract. Problem stated here is connected with music information processing, especially with Braille music notation. The main objective of this paper is usage of semantics for optimization of Braille music scores processing. This issue is important in the area of huge Braille music scores. Our approach is based on structuring – both syntactical and semantic – in spaces of music information. Optimization would not be possible without such structuring. The main idea is connected with logical score partitioning into smaller pieces that are weakly dependent between each other. Optimization is based on closing changes to small syntactical and semantic items of the structure. Each change during editing touches on of such small items instead of processing significant parts of the whole structure.

Keywords: Data understanding, knowledge processing, music representation, music data processing.

1 Introduction

In this paper we discuss the problem of processing Braille music notation. Braille music notation is an example of language of communication between people and, in this case, it is called a language of *natural communication*, c.f. [26]. Communication is seen as intelligent exchange of information, which involves information understanding by all sides of communication. Nowadays technologies support information processing helping to improve interpersonal communication. Languages of communication can be seen as tools for constructing vehicles carrying information. These vehicles are texts of natural languages, scores of music notation, scores of Braille music. This remark brings a way to deal with information processing including personal and automatic processing. Such processing requires information structuring as well as identification of structures of information.

In this study we discuss syntactic and semantic structuring of Braille music notation. The discussion is aimed on optimization of automatic processing of big scores of Braille music notation. Namely, we investigate a possibility of avoiding processing of big parts of the whole score in editors of Braille music. Such optimization is worth attention when big scores are edited. However, due

to properties of music information, for many symbols of music notation, simple editor's tools do not allow for limited processing. It is necessary to employ deep syntactic and semantic analysis of the score or its fragment corresponding to editing activities in order to limit to necessary minimum the spectrum of processed symbols.

The paper is structured as follows. Fundamentals of syntactic and semantic tools involved in structuring music information are recalled in section 2. In section 3 we describe shortly formats of music information. We only recall general purpose music representation formats and describe to some extent formats related to Braille music notation. These formats are crucial for the main track of this study. Introduced efficiencies in Braille music processing are studied in section 4. On the basis of illustrative examples we describe syntactic and semantic methods leading to minimization of parts of Braille music score subjected to processing during editing operations. Expansion of presented examples to other symbols of music notation is straightforward. Finally, we come to conclusions anchored in real Braille music processing editor developed in frames of the acknowledged research project.

2 Syntactic and Semantic Mappings

Any intelligent processing of constructions of languages of natural communications requires uncovering structures of raw data. There are different ways leading to structuring. Our interest is focused on employing syntactic and semantic structuring of music information with special emphasis put on Braille music notation. It is obvious that raw data without any structuring is useless in intelligent communication. Otherwise the processing covers some characters from alphabet without any meaning. The aim is to create generic method for integrate syntactic and semantic structuring of music information. This structuring allows for optimized processing of music information described in different languages including Braille music notation and printed music notation. For people with good eyesight we bind Braille music notation with printed music notation in this study. The method is an extension of a likewise study in [4].

2.1 Syntax

Syntactic structuring of music information is the first stage of the analysis process. We will utilize context-free grammars for syntactic structuring of Braille music notation and printed music notation. We refer to and will continue discussion of syntactic structuring outlined in [4].

Let us recall that we use formal grammars, which are systems $G = (V, T, P, S)$ where: V is a finite set of nonterminal symbols (nonterminals), T is a finite set of terminal symbols (terminals), P is a finite set of productions and S is the initial symbol of grammar, $S \in V$. In general productions can be seen as a finite binary relation $P \subset (V \cup T)^+ \times (V \cup T)^*$. A grammar G is context-free one (CFG) $\iff (\forall p)(p \in P \Rightarrow p \in V \times (V \cup T)^*)$.

Since there is no evidence that Braille music notation is a context-free language, we do not attempt to construct a context-free grammar generating the language of Braille music notation. Instead we use context-free grammars covering the language of Braille music notation. Such grammars will generate all constructions of Braille music notation and some others, which are not valid Braille music constructions. This approach cannot be used in generating Braille music scores or parts of scores or in checking their correctness. However, since we employ context-free grammars for processing scores, which are assumed to be correct, the approach is proven. A discussion on construction of context-free grammars covering printed and Braille music notations is outlined in [4].

2.2 Lexicon

Lexicon is the space of language constructions, each of them supplemented with possible derivation trees, also known as parsing trees. Lexicon includes relations between items of this space. Such a tree satisfies the following rules:

- it is a subtree of the derivation tree of the whole score,
- it is the minimal tree generating the given language construction,
- the minimal tree can be extended by a part of the path from the root of this tree toward the root of the score derivation tree

Due to the last condition, usually there are many trees for a given language construction. We do not recall the meaning of parsing tree in a context-free grammar, refer to [5] for definition of it.

Different trees supplementing a given language construction describe different context of the language construction. For instance, if we consider a sequence of consecutive notes, the minimal derivation tree for these notes matches all such sequences in the whole score, if more than one is present. If the minimal tree is extended to the root of the derivation tree of the whole score, than it represents only this given sequence of notes, c.f. [3] for details. The concept of the lexicon can be applied, for instance, for better understanding and better performing of structural operations, e.g. *find* operation.

2.3 The World: The Space of Hearing Sensation

Languages allows to describe a real world of things, sensations, thoughts, ideas etc. Braille music notation describes the space of hearing sensations, which can be outlined as the space $B \times D \times P$ of triples (b, d, p) . Each triple defines the performed sound, where b is beginning time, d is duration and p is pitch of this sound. In general, objects of the real world may be outlined with much richer set of features, but this simple triples are sufficient for our discussion, c.f. [4].

Above mentioned approach is very generic, refers to physical essence of a sound and has not any links to a particular notation. This structure can be used for any music notation, especially Braille music notation. This definition of the space of hearing sensation is also very useful in case of other structural operations, e.g. *conversion*, c.f. [4].

The purpose of using the world of real objects is to tie meaning to syntactic structures, i.e. to tie meaning to lexicon elements of the Braille music notation. This assumption allows us to cast different descriptions music information and different formats representing music information onto the space of hearing sensation. In this way, it is possible to construct collaborating methods, which operate on these different descriptions and formats, c.f. [1]. We apply the idea for formats used in a real processing of Braille music accomplished in frames of the Braille Score project, with the BMF format described in next sections as the space of music sensations.

2.4 Semantics

As mentioned in the previous section, descriptions of music notation expressed in different languages and representation of music notation in different formats are cast on the world of hearing sensations. Such casts are called semantics of descriptions and representations of music information. Formally, let B is the lexicon of Braille music notation and H is the space of hearing sensation. Semantics S is a relation:

$$S \subset B \times H$$

The Braille music notation is tightly connected with objects in internal format. Here we consider the BMF format as the example. Every element of the Braille music notation (a character, a Braille cell) contains information about its owner – element from BMF. It is obvious that every Braille cell has such element, because:

- characters present in notation **represent** some data from BMF. Otherwise they are redundant and would be thrown away from notation,
- characters in notation **create** some data in BMF. Otherwise they are redundant, carries no information and would be thrown away from notation

Every element in notation has related element(s) in BMF, and every element in BMF has related element(s) in notation. This relation is many-to-many. One element from BMF may create many Braille cells and one Braille cell may create many elements from BMF. Nevertheless, one meaning is chosen, often not at once, but depending on neighboring Braille cells.

3 Formats Description

Musical data is stored as files or physical scores (manuscripts or printed ones – images/photos of scores). The files are written in different formats. Rough division enumerates formats that are printed score (e.g. MusicXML files), Braille score (files with Braille music) or pure sound (e.g. MIDI files) oriented.

When processing music data it is convenient to use more than one internal format. Internal formats should meet external formats of data or to make processing easier.

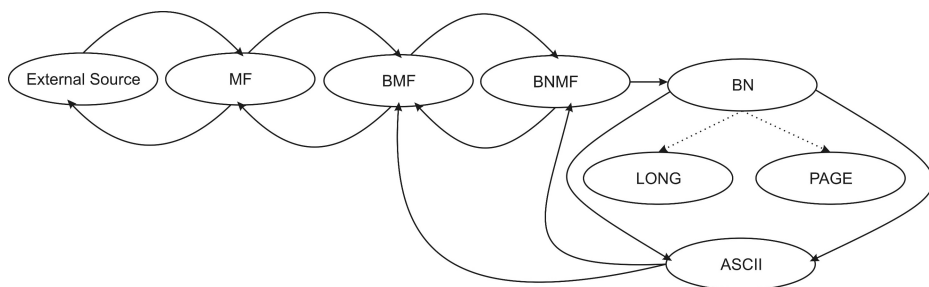


Fig. 1. A typical data flow in Braille Score

A set of the internal formats is presented in the Figure 1. These structures store musical information in varied way: external types, printed music oriented, Braille music oriented and Braille music files. Between the most of these types is available bidirectional conversion, except one way conversion $\text{BNMF} \rightarrow \text{BN}$. There exist implicit conversions $\text{BN} \rightarrow \text{ASCII} \rightarrow \text{BNMF}$.

Bubble „ASCII” represents Braille music written as ASCII file. Bubbles „LONG” and „PAGE” indicate displayed notation in two manners: as infinite line or as broken line. These two bubbles do not handle structured data, in opposite to other bubbles. Each of the other bubbles from the Figure 1 is described in next paragraph.

3.1 External Source

This data source is used to create the main internal format called „MF”. There is possibility to build MF from following sources:

- MusicXML – a popular file format used to write music scores (c.f. [7]),
- MIDI – file format that contains only sound information (c.f. [8]),
- .bmp, .jpg, .tif, .png – images of scores, which after recognition process will create MF,

3.2 MF

MF (Figure 2) is the main internal format. It is not tied with Braille notation. MF can be displayed and edited as printed music notation. MF consists of several abstract data types, the main are:

- **Score** – represents whole score, contains **Staffs**, **MusicSystems**, time signature and many other information about page parameters and metadata (such as title, author, composer)
- **Staff** – contains information about instrument, key signature and clef
- **Music System** – contains barlines information and **Measures** (this construction can be misleading because of different naming convention: sometimes multiple staves is called measure; in Braille Score each measure contains one staff, many staves creates system)

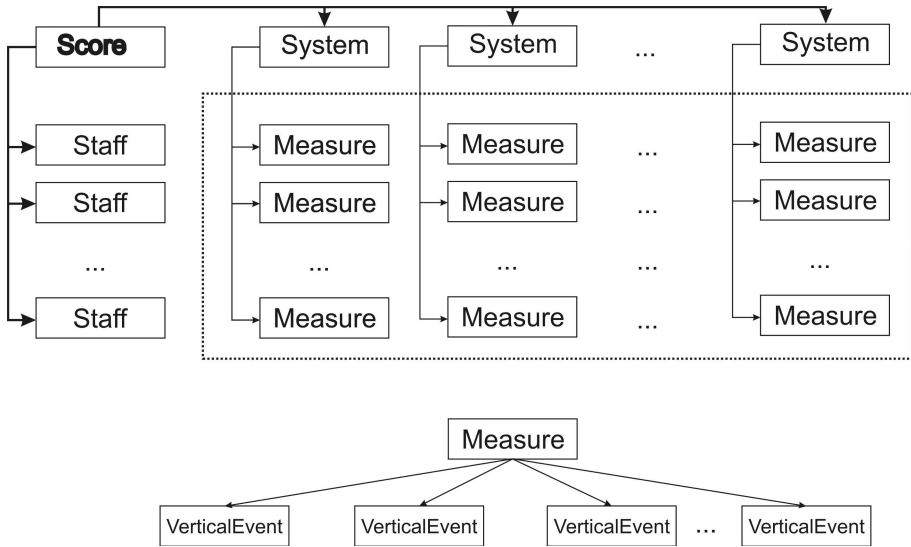


Fig. 2. MF

- **Measure** – contains **VerticalEvents**
- **VerticalEvent** – elements that represents one particular event during given measure's time, contains voices slots and additional dynamic/ornamentation information; each voice slot may contain note/rest element

This above description is general draft of MF structure. Mentioned elements are emblazoned by additional properties, such as ornamentation, articulation, dynamic symbols.

3.3 BMF

BMF (Figure 3) is the main internal format in Braille music editor. It handles Braille music information encapsulated at high abstraction level. This format's structure is similar to „MF“, but there are some differences in regard of Braille music notation. BMF consists of:

- **Score** – represents whole score, contains **Staffs**, **Systems**, time signature and many other information about page parameters and metadata (such as title, author, composer, interval reading direction)
- **Staff** – contains **Measures**, instrument names
- **System** – contains **Measures**, barlines, time signatures
- **Measure** – contains **Voices**, key signature
- **Voice** – contains notes/rests

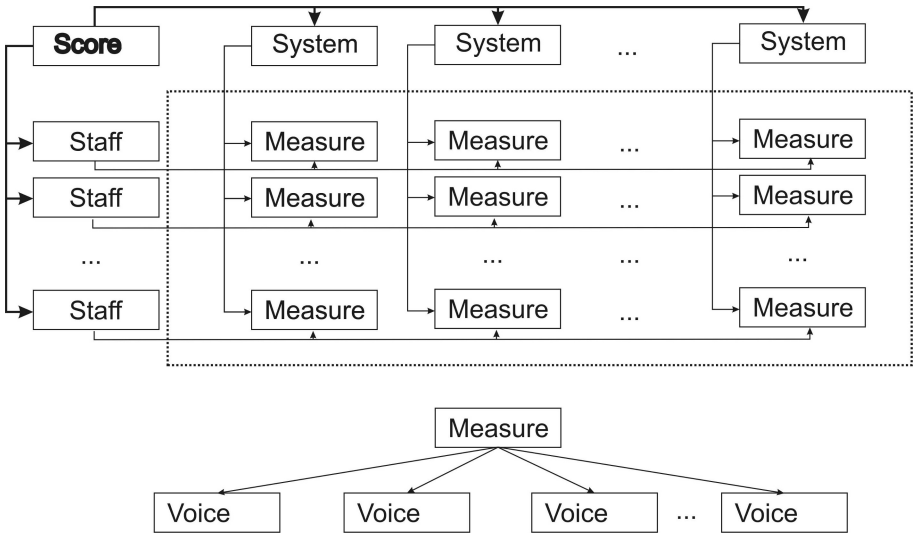


Fig. 3. BMF

3.4 BNMF

BNMF (Figure 4) is simple format that represents Braille dots at high level abstraction with semantic attached. This format is organized as follows:

- **Score** – represents whole Braille music score, contains **Staffs**
- **Staff** – contains **Measures**
- **Measure** – contains **BSymbols** and ending **Small Context**
- **BSymbol** – element that represents Braille dots
- **Small Context** – set of elements, that allow to process notation more efficiently, will be described in the next paragraph

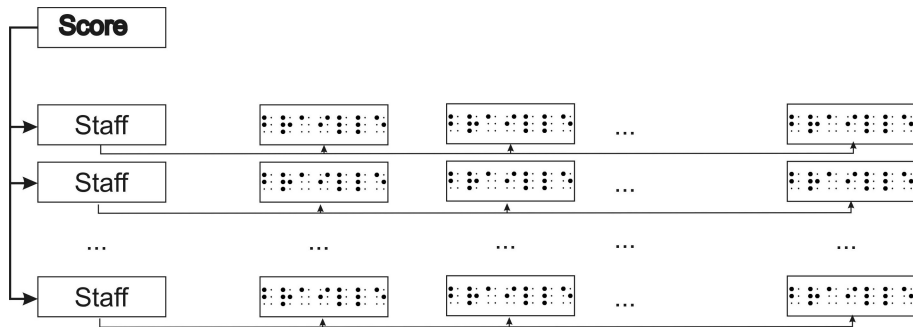


Fig. 4. BNMF

Each BSymbol has information about protuberant dots of this cell. It has connection to element from BMF, which produces this Braille cell and gives the meaning to it.

This format was chosen to be the target of editing operations. Changes in Braille music notation are made in this format. Then, if necessary, changes are propagated in other formats.

3.5 BN

BN (Figure 5) is an intermediate layer, that contains both: Braille cells with semantics as an infinite line and Braille cells with semantics as a broken line. This format is an element, that can be displayed or written into the file in both manners: as an infinite line or as a broken line.

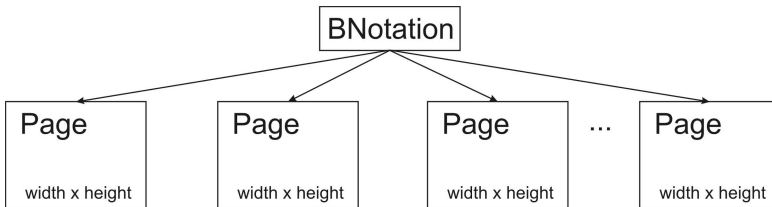


Fig. 5. BN

3.6 Context

Context is used when reading Braille music notation (e.g. from ASCII file or from BNMF format). It contains information about signs read previously which has influence on current sign.

Elements of Context have different scope. Arbitrary division itemizes inter-measure elements (33 units) and extern-measure elements (7 units). The first ones have scope limited to one measure, i.e. they have no influence on signs from other measures. Extern-measure elements have scope limited to one staff and will be called „Small Context“. To sum up: Context consists of Small Context and inter-measure elements.

Small Context contains following elements:

- opened slurs count
- opened wedges count
- intervals reading direction
- slurs numbering
- short slurs error check
- last read note (pitch and octave)
- last read attributes (key and time signature)

Small Context Equality. Two Small Context are equal if and only if their respective fields are equal.

4 Efficient Processing the Braille Music Notation

Very often editing operation changes only one measure, and that operation has small scope – single measure or a few measures in the neighborhood. There is no need to process whole score or staff. Using the Small Context, there is an ability to avoid expensive computing.

4.1 Draft

Thanks to Context construction, i.e. Small Context included in Context, notation can be faster processed. All changes are made on BNMF, because of presence of pure notation in this case, and operation touches directly notation.

Because of Lexicon construction (rejection from consideration some structures, e.g. slurs between staves), any of the staves can be process independently. Context is used during processing every measure of the score. Each measure remembers Small Context that occurred at the end of processing this measure. Small Context – because it is a subset of Context and contains all extern-measure data. Remembering of the ending (except beginning) Small Context because:

- for each measure k , except the first measure, the beginning context is the ending context of measure $(k-1)$
- for measure number 0 the beginning context is new context
- knowledge of ending state at the end of staff

4.2 Mechanism of Processing

Processing the Braille music notation occurs in two situations:

- processing a notation the first time – no knowledge about ending contexts for the measures
- processing a notation again – knowledge about ending contexts for the measures because of previous processing

The first case does not benefits from this method for efficient processing. Processing the whole notation is all and enough what can be done.

The second case allows to avoid processing whole notation. This situation occurs when notation has been changed; the changed measures are marked. We assume that change affects only one measure. In other case, we can treat multiple position change as multiple changes.

Algorithm 1. Algorithm for efficient score processing

```

1: for all s in Staves do
2:   ctx = new Context
3:   for all m in s.Measures do
4:     if m.HasChanged OR m.SmallContext == NIL then
5:       if m != s.Measures[0] then
6:         ctx = s.Measures[m.Number-1].SmallContext
7:       end if
8:       process measure m with Context ctx
9:       m.HasChanged = false
10:      if m.Number < s.Measures.Count - 1 AND !ctx.Equals(m.SmallContext)
then
11:        s.Measures[m.Number+1].HasChanged = true
12:        m.SmallContext = ctx
13:      else if m.Number == s.Measures.Count - 1 then
14:        m.SmallContext = ctx
15:      end if
16:    end if
17:  end for
18: end for

```

4.3 Deferred Semantic Bounding

In paragraph 4.2 was made assumption, that each change is processed separately. That approach causes sometimes correct but inefficient processing.

There is a need to design a special mode which allows to defer semantic bounding. This mode prevents score processing till it become switched off. Just after that the score is processed and semantic is bounded too.

The mentioned above mode is useful when paired symbols included in Small Context are added, i.e. slurs or wedges beginnings and endings. It is worthwhile to process score after full symbol insertion, i.e. beginning and ending symbol.

Otherwise it causes instability at the end of measure where insertion occurs: Small Contexts are not equal to ctx. This instability is propagated from the changed measure to the end of the staff (in pessimistic situation). The insertion of the second of the paired symbol causes the same: processing remaining measures. The second phase reverts almost all changed ending Small Contexts to the state from before the first processing.

To sum up:

- **with** deferred semantic bounding the processing affects a few required, changed measures
- **without** deferred semantic bounding the processing may affect whole staff twice in pessimistic case (insertion of paired symbols and processing them separately)

Deferred Semantic Bounding Example. This section presents example to illustrate deferred semantic bounding issue. Figure 6 shows editing operation

Fig. 6. General description of the editing activity

that is aimed to add slur in the notation. This process is partitioned in three stages: before operation, after inserting beginning sign of the slur and after inserting ending sign of the slur. Each phase of the slur insertion is described in sequence by: general overview draft, printed music notation and Braille music notation. Correspondence between beginnings and endings of slurs in printed music and signs in Braille music is marked in Figure 6 by bold arrows.

The initial score contains three slurs (first stage). In the second stage beginning of the slur is added to the first note. That makes the inserting slur to be opened from one of the ends. In terms of processing the Braille music it creates slur that never ends, that lasts till the score ends.

Ending of the slur is inserted in the third stage. After this activity the slur is defined by beginning and ending. The score has four slurs. All changes touch one measure. It means that it is enough to process this particular measure. Efficient processing is performed in that case thanks to deferred semantic bounding.

Figure 7 illustrates inefficient processing. Slurs are marked in the same way as in the Figure 6 – by vertical, bold segments connected in pairs by dotted lines. At the end of each measure the number of opened slurs is marked.

The big arrow in the measure rectangle indicate processing this measure (bounding semantic). The big and dark arrow is connected with changed measure, which should be always processed because it contains the change and is the smallest item of syntactical and semantic structuring. The light arrow matches

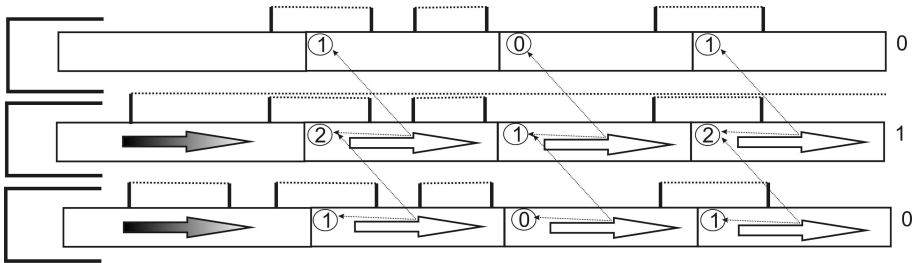


Fig. 7. Inefficient processing during edition

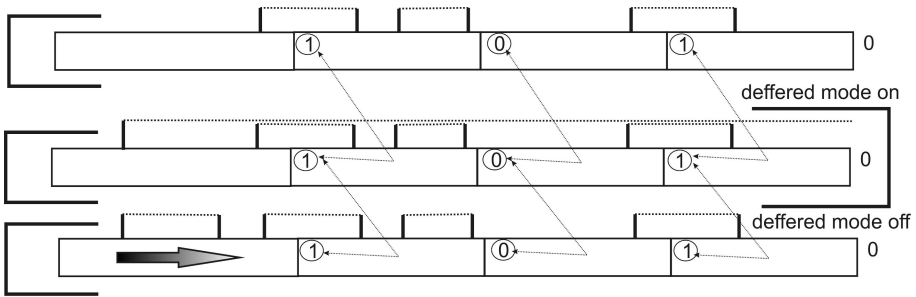


Fig. 8. Efficient processing during edition

processing that occurs in the measure as a result of changes of small context (refer to algorithm 4.2, step 10). The light arrows appear because of absence of the deferred semantic bounding. In all that cases occur difference between number of opened slurs in first to second and second to third stages for respective measures.

Please notice that the light arrow processings are redundant because numbers of opened slurs in the third stage are the same as in the first stage. That means no final change in opened slurs in measures marked with light arrow.

The above conclusion brings as to deferred semantic bounding. This term means that notation is not processed. This implies no changes in small contexts. Finally, this implies no processing in measures, except one measure where editing operation was performed.

Correct processing is shown in the Figure 8. The dark arrow occurs once because of insertion the beginning sign for slur in deferred semantic bounding mode. There is no light arrows.

5 Conclusions

This paper is a description of the idea of efficient processing Braille music notation. Appropriate algorithm was proposed to show described method and illustrate additional consequences such as deferred semantic bounding. Efficient

processing way can be used with other music languages, e.g. printed music notation. It demands only small context definition and smallest syntax and semantic structures selection.

Introduced method and deferred semantic bounding mode are implemented in Braille Score – application that allows to process music data. Nowadays deferred semantic bounding mode is activated manually, but we consider to automatize this feature.

Acknowledgement. This work is supported by The National Center for Research and Development, Grant no. N R02 0019 06/2009.

References

1. Breaking accessibility barriers in information society. Braille Score - design and implementation of a computer program for processing music information for blind people, the reserach projectconducted in the Systems Research Institute, Polish Academy of Sciences, 2009-2012, supported by The National Center for Research and Development, Grant no. N R02 0019 06/2009
2. Homenda, W.: Integrated syntactic and semantic data structuring as an abstraction of intelligent man-machine communication. In: ICAART - International Conference on Agents and Artificial Intelligence, Porto, Portugal, pp. 324–330 (2009)
3. Homenda, W., Rybnik, M.: Querying in Spaces of Music Information. In: Tang, Y., Huynh, V.-N., Lawry, J. (eds.) IUKM 2011. LNCS (LNAI), vol. 7027, pp. 243–255. Springer, Heidelberg (2011)
4. Homenda, W., Sitarek, T.: Notes on Automatic Music Conversions. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS (LNAI), vol. 6804, pp. 533–542. Springer, Heidelberg (2011)
5. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley Publishing Company (1979, 2001)
6. Krolick, B.: How to Read Braille Music, 2nd edn. Opus Technologies (1998)
7. Castan, G., Good, M., Roland, P.: Extensible Markup Language (XML) for Music Applications: An Introduction. In: Hewlett, W.B., Selfridge-Field, E. (eds.) The Virtual Score: Representation, Retrieval, Restoration, pp. 95–102. MIT Press, Cambridge (2001)
8. MIDI 1.0, Detailed Specification, Document version 4.1.1 (February 1990)

ETSeM: A Energy-Aware, Trust-Based, Selective Multi-path Routing Protocol

Manali Chakraborty and Nabendu Chaki

Department of Computer Science & Engineering, University of Calcutta, India
92 APC Road, Kolkata 700009, India
manali4mkolkata@gmail.com, nabendu@ieee.org

Abstract. Multi-path routing protocols are used for different types of wireless networks primarily to enhance reliability of packet delivery. The frequency of route discovery is also less for multi-path routing protocols as these are more fault-tolerant. However, the overhead of route discovery in terms of congestion and energy requirement is much higher for multi-path routing as compared to single-path routing. In this paper, a restricted multi-path routing algorithm has been proposed that dynamically selects the number of neighboring nodes through which packets would be transmitted. The selection and degree of multi-path depends on multiple factors like the remaining energy of the node, trust value of that node, number of already existing paths through that node etc. The protocol is designed in such a way, that the burden of routing is lower on the weaker nodes and the nodes with more resources will have to perform more tasks. Consequently, the lifetime of the network would be higher as compared to multi-path routing protocols. Besides, the data reception rate, defined as the ratio of the total number of packets received by the sink node and the total number of packets sent by the source node, is much higher for the proposed protocol than any single path routing. While the routing load is balanced among the nodes, the multiple routes also increase the reliability.

Keywords: Selective multi-path routing, back-up path, fault tolerance, trust.

1 Introduction

The main objective of different types of wireless networks, such as Sensor networks, MANETs, Wireless Mesh Networks is to transfer data from one node to another. This communication must be reliable. In order to ensure reliability, multi-path routing protocols can be utilized [5]. Due to the unpredictability of the environment, and unreliability of wireless medium, a single path routing is more prone to failure in a wireless network. This makes the protocol unreliable [7]. On the other hand, Multi-path routing can easily recover path failure by utilizing alternative routes. The frequency of route discovery is also less in a network with multi-path routing, since the network can tolerate one or few link failures, and still continue working with alternative paths [8]. It also provides the benefits of fault tolerance, load balancing and bandwidth aggregation etc [7].

In spite of such benefits, the overhead of route discovery and maintenance is often higher in multi-path routing than single-path routing. In this work, the proposed selective multipath routing protocol aims to combine the benefits of both the methods. Besides, it is expected to offer more efficient load balancing. The word selective means that every node has the capability to select a subset from its neighboring nodes through which it can transmit. The selection is based on the remaining energy, and trust value of that node as well as on issues like the number of existing paths through the node, etc.

The novelty in route selection for the proposed protocol is that the source and the intermediate nodes may choose zero, one or more nodes for the next hop transmission dynamically depending on the current status. As for example, just like single-path routing there could be only one selected node for the next hop, if the communication is reliable, trusted and meets other criteria. An intermediate may not forward the RREQ packet at all depending on the status of the factors mentioned. However, depending on the status, an intermediate may decide to broadcast to its entire neighborhood too. The selection is run-time and dynamic.

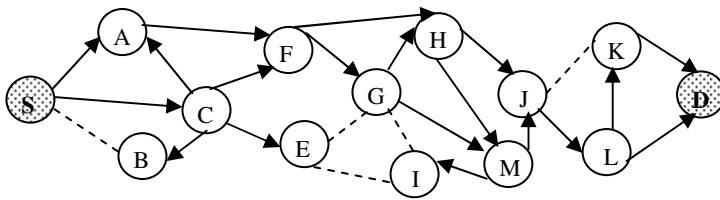


Fig. 1. Selection of next-hop for Routing

In figure 1, for example, source node S transmits to nodes A, C and not to B. Node A transmits to F only, while C transmits to all its neighbors A, B, E and F. It is assumed that initially all the nodes are connected through bi-directional channels as shown by dotted lines. The directed continual lines are the selected links that eventually sets up routes up to D. The routes that are set for the selection in figure 1 is shown in figure 2.

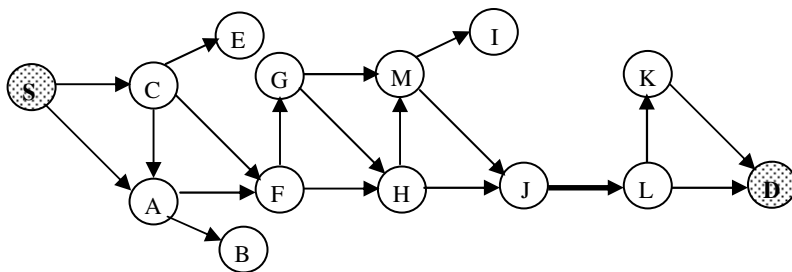


Fig. 2. Multi-path Routing from S to D Routing

In the example, there are two or more alternate paths from the start node S and each intermediate node to destination D, except node J. As evident from this example, all the routes from S to D must be routed through J. There are several alternate routes to reach J from S. However, there is only one selected path between nodes J and the next hop destination L. This is in spite of the fact that both K and L are neighbors of J in the direction of destination D. The proposed protocol permits this. This depends on the reliability of the link J-L and the trust values, energy level, etc. for the two nodes. The underlying premise is that multipath routing may not be needed at all the steps. Multiple next-hop nodes have been used for other intermediate nodes depending on the status. The path selection is dynamically decided in a reactive manner.

The rest of the paper is organized as follows: section 2 describes related works, section 3 describes the routing process, the performance analysis of this algorithm is presented in section 4, the simulation results are presented in section 5 and section 6 concludes this work, with acknowledgement in section 7.

2 State of the Art Review

A brief review of existing Multi-path Routing algorithms in different Wireless Networks is presented in this section. The route discovery and selection of multiple routes is one of the fundamental issues in multi-path routing. Two broad approaches are suggested for this purpose; node-disjoint and link disjoint [7]. In different node-disjointed approaches, multiple routes are created with an assurance that no common node can exist between them. In the link-disjointed methodologies, common nodes may exist between several routes, but links between two nodes do not overlap [5].

Multi-path routing algorithm is of three types. In the first method a back up path is used only when the primary path between a pair of nodes is down. The back up path is set up simultaneously with the primary path. In the second method, multiple paths are used simultaneously to balance the load of the primary path. Initially the primary path is used for data transmission, but when it has heavy traffic, the other paths also participate in packet transmission to reduce the burden from that primary path. In the third method, every node disjoint path between a pair of nodes is used to increase the end to end performance by transmitting data among several paths [1].

A meshed multi-path routing M-MPR [2] was proposed to provide mesh connectivity among the nodes. It also uses selective forwarding of packets among multiple paths. The selection is based on the condition of downstream forwarding nodes. End to end forward error checking (FEC) is used to reduce the overhead of retransmitting the packets based on acknowledgement. Besides being energy efficient, higher throughput achievement has been claimed in [2] as compared to any other node disjoint multi-path routing protocol.

Another multi-path [1] routing for wireless networks combines the idea of clustering and multi-path routing together. Clustering is used to speed up the routing by structuring the network nodes hierarchically, and multi-path routing is used to provide better end to end performance and throughput. The solution in [1], according to its authors, is less prone to interference, than conventional multi-path routing. It is also

quite simple as each path in the CBMPR just passes through the heads of clusters, resulting in a simple cluster level hop-by-hop routing. A reliable and hybrid multi-path routing, RHMR for MANET was proposed in [5]. It uses a proactive-like routing for route discovery and reactive routing for route recovery and maintenance.

LIEMRO [6] is another node-disjoint multi-path routing based on event based sensor network to improve QoS in the terms of data reception rate, lifetime, and latency. The primary path from source node to sink node is consist of the nodes with minimum packet transmission cost at each step. Similarly, the second path is established using the second best nodes at each step. Extra routes are only established if they don't decrease data reception rate at the sink node.

MHRP [3] is a Hybrid Multi-path Routing protocol that was designed to properly exploit the inherent hybrid architecture of WMNs. It uses Proactive Routing protocol in mesh routers and reactive routing in mesh clients. By efficiently using the resourceful router nodes in route discovery and security mechanism, it reduces overhead from client nodes, which are mobile and have fewer resources.

Another multi-path routing for WMNs was proposed in MRATP [4]. It uses a traffic prediction model based on wavelet-neural network. The main idea of this paper is to set up one primary and some backup paths between a pair of nodes. The primary path is used to transmit the data, until any node on that path generates a congestion signal. Then the back-up paths is used to balance the load in the network. It is claimed that [4] reduces end to end delay and balances the load of the whole network efficiently.

3 The Proposed Algorithm

Wireless Ad Hoc networks were developed for reliable data communication and load balancing. Multiple path communication is the basic need behind these two objectives. If these attributes of Wireless networks are not utilized properly; one cannot achieve the best out of this network paradigm. Moreover, multi-path routing assists in achieving security in routing protocols. Most of the proposed schemes are not able to minimize the overhead of storing extra routes, through the life time and the maintenance cost of those routes. These limitations urge a need of a routing protocol which can manipulate the degree of multi-paths according to the energy level, number of paths through a node and trust value of a node.

In this section, the working of the proposed algorithm has been described. There are mainly two parts, 1. Neighbor Discovery, and 2. Route Establishment. The working of this algorithm is based on the following principles:

- Every node has to maintain two arrays; HEALTH and TRUST and two variables; ENERGY and PATH.
- Every node maintains the health information about its 1 hop neighbors.
- HEALTH is a linear function of the remaining energy of that node, number of paths already exists through the node and trust value of the node.
- Every node has to send its energy and path metrics to its 1 hop neighbors.

3.1 Neighbor Discovery

At first every node obtains some information about its neighbors. Each node broadcasts a HELLO packet to identify its neighbors within one-hop distance. On receiving the HELLO packets, each node replies with a ACK packet, containing the remaining energy of that node, the value of a counter variable PATH, which denotes the number of paths already existing through that node and the trust value of that node, given by its neighbors. We assume that, every node send its information reliably. After that, every node calculates the value of the variable HEALTH of its neighbors in terms of their remaining energy, number of paths already passing through the node and the trust value of that node.

$$\text{HEALTH} = f(\text{remaining-energy, trust, PATH})$$

3.2 Route Establishment

Figure 3 shows the route establishment process for this algorithm. To establish a route between a source node and a sink node, the source node sends a ROUTE_REQUEST packet towards the sink node. Each node checks for the healthiest node among its neighbors. If the value of the variable HEALTH for the healthiest node is greater than 90%, then it forwards the ROUTE_REQUEST packet through that node only. If the value of HEALTH is in the range of 75% to 90%, then the sender forwards the ROUTE_REQUEST packet through two nodes, the healthiest one and the second healthiest one. If the value of HEALTH is in the range of 60% to 75%, then the ROUTE_REQUEST packet is forwarded through the first three healthiest nodes in the neighborhood. If the value of HEALTH is in the range of 45% to 60% then the ROUTE_REQUEST packet is forwarded through the first four healthiest nodes in the neighborhood. Otherwise, that is, when the value of HEALTH is below 45% for every node in the neighborhood, the sender broadcasts the ROUTE_REQUEST packet through all the neighbors.

Each node, after receiving the ROUTE_REQUEST packet, forwards it similarly, and they also keep the ID of the node from which this packet has been received and inserts its own ID in the packet, to prevent the looping error. A node can not forward the ROUTE_REQUEST packet to such a node, whose ID is already in the packet. When the ROUTE_REQUEST packet reaches to the sink node, sink node replies by transmitting a ROUTE_REPLY packet to the node from which it receives the ROUTE_REQUEST packet. Every node along through the path from the sink node to the source node, increment the value of the counter variable PATH by one, every time when it receives a ROUTE_REPLY packet. Thus the PATH variable denotes the number of path passes through this node.

Upon receiving the ROUTE_REPLY packet, the source node confirms a path to the sink node, and uses this path to transmit data. Each node distributes the load equally through all the paths starting from that node towards the sink node. i.e. if some node have only one path towards the sink node, it transmits the data through that path only, and if some node have two or more paths towards the sink node, it divides the data equally, and transmit through each route. Every node gain some rewards from its neighbors, when it forwards the data packets successfully.

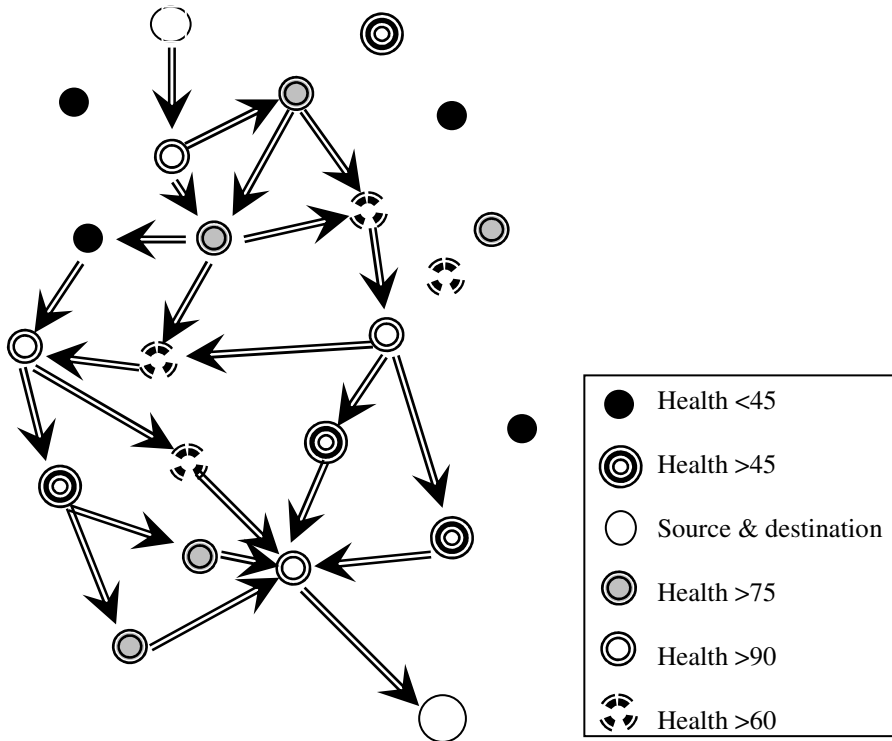


Fig. 3. Selection of route depending on HEALTH

4 Performance Analysis

In traditional multi-path routing, one has to maintain all the routes, though it may be the case that, maximum of the routes will remain inactive through out the lifetime of the network. Thus, the maintenance and setup cost of multiple routes are become overhead. On the other hand, in this algorithm, the degree of multi-path depends on the health of the nodes of the network itself. The number of multi paths is also dynamically changes with the overall condition of the nodes in the network. A node only increases the number of routes towards the sink node, when it finds that the next hop nodes are not capable enough to carry the total load. Thus, it can be said that it is a much intelligent approach for multi-path routing. Due to this feature, the data reception rate, defined as the ratio of the total number of packets received by the sink node and the total number of packets send by the source node, is much higher than any single path routing. Also the end to end delay is decreased in this algorithm, due to the reason that the number of routes will increase with the decreasing health of nodes. Thus the load is balanced among the nodes and the multiple routes also increase the reliability. The lifetime of the network also increases with this algorithm, as the nodes

with less remaining energy are released from the burden of transmitting the total load. The burden of each node is decreased proportionally with its remaining energy.

5 Simulation Results

The proposed algorithm has been successfully simulated using the standard Network Simulator QualNet. The findings are based on simulation results. We have taken the results by varying the node density from 10 to 50 nodes with the fixed mobility 30 mps. The simulation scenario and settings are described as follows:

Table 1. Simulator parameter settings

Parameter	Value
Terrain area	1500X1500 m2
Simulation time	100 sec
Mac Layer protocol	DCF of IEEE 802.11b stan-
Traffic Model	CBR
Number of CBR applications	10 % of the number of nodes
Mobility Model	Random Waypoint
Trust Value of normal nodes	0-10
Initial Energy Value of normal	5000

5.1 Packet Delivery Ratio

The Packet delivery ratio (PDR) is an important metric to analyze the performance of a routing protocol. PDR is defined as the ratio of the total number of packets send by the source node and the total number of packets received by the destination node

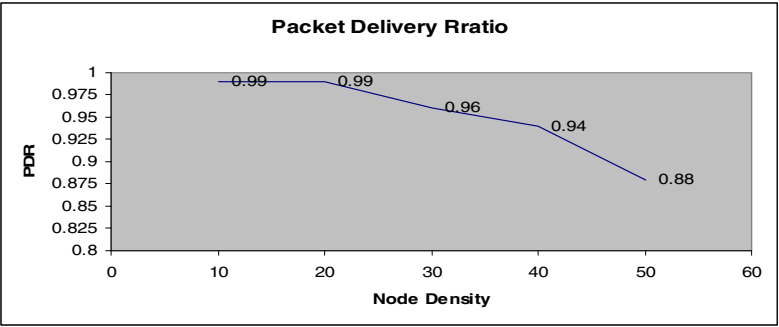


Fig. 4. Packet Delivery Ratio Vs Node Density

As shown in fig. 4 the packet delivery ratio is almost 100% for less node density. But the packet delivery ratio decreases with increasing node density, because of the increasing congestion in the network. Still, with higher node density the packet delivery ratio is quite stable.

5.2 Throughput

The throughput is defined as the total number of bits received by the receiver per second. The size of each data packet is 512 bytes and one data packet is sent in every second by the sender.

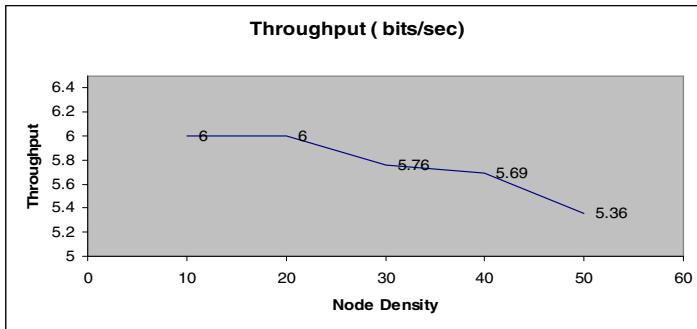


Fig. 5. Average Throughput Vs Node Density

Fig. 5 shows that the average throughput of the nodes in different node density. The average throughput of nodes decreases as the node density increases. The rate of decrease in throughput is almost 2.5%-3% in average with 100% increment in the node density. However, it still gives a moderate result.

5.3 End to End Delay

The end to end delay can be defined as the time that a data packet takes to traverse the distance between the sender and the receiver.

It can be seen from figure 6, that the average end to end delay is quite stable with the node density.

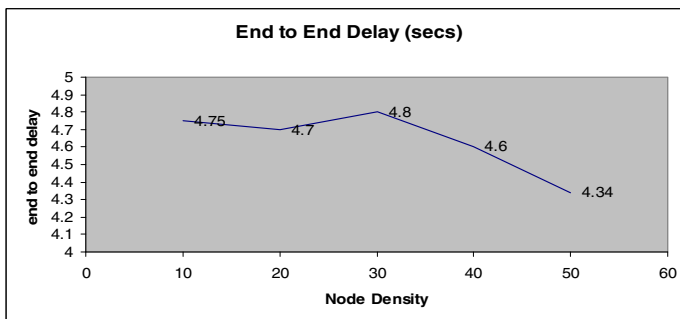


Fig. 6. Average End – to – end Delay Vs Node Density

5.4 Jitter

Jitter is expressed as an average of the deviation from the network mean latency.

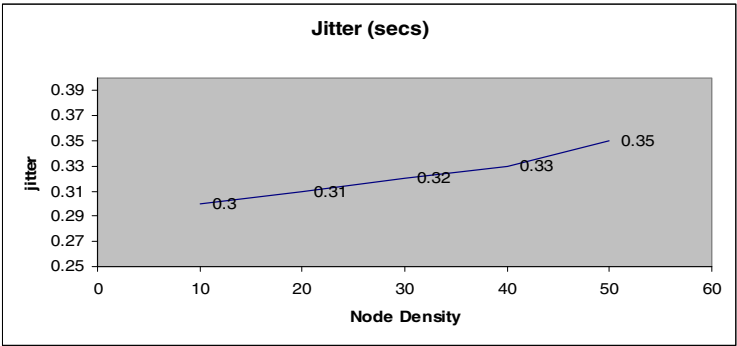


Fig. 7. Jitter Vs Node Density

The jitter in the network is increasing quite linearly with the increment in node density. It can also be seen from figure 7, that the value of the jitter is much lower with our algorithm.

5.5 Average Energy Consumption

The average energy consumption of the network under various node densities is also simulated.

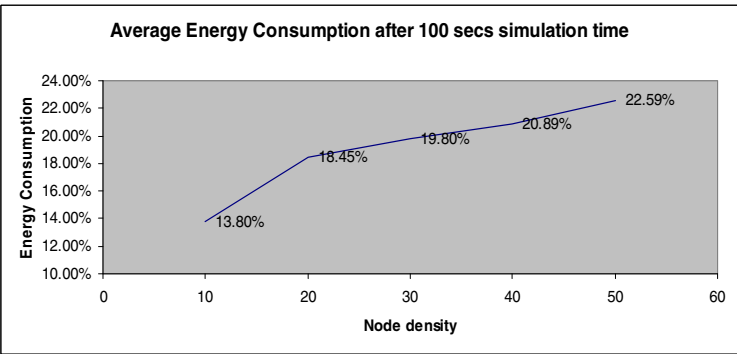


Fig. 8. Average Energy Consumption Vs Node Density

The energy consumption of the network increases with the number of nodes, because, with the increasing number of nodes, every node has to broadcast more hello messages and also has to calculate the health and trust of those nodes. However, the power consumption is much lower than the other multi-path algorithms, as the proposed algorithm tries to divide the load of the network equally among the nodes, so that no node can be suffer.

6 Conclusion

In this paper, a new selective, multi-path routing for effective load balancing is presented. In this algorithm, every node selects its next hop forwarding nodes very intelligently. The selection procedure helps to improve the lifetime of the whole network, and as the selection is also based on trust values, the algorithm provides reliability. Moreover, the multi-paths also increase the data delivery rate. The effectiveness of this algorithm is validated through theoretical performance analysis as well as simulation results using QualNet. The proposed methodology builds the foundation for several meaningful extensions in future. The incorporation of different security mechanisms to this routing protocol is left as future work. In this paper, the evaluation of health parameter of each node depends on three parameters: energy, path and trust. However, the impact of other factors like interference, latency on the health of each node could be an interesting extension of the proposed work.

Acknowledgement. We would like to thank the Advanced Technology Cell, DRDO Cell for approving our work as a Defense – related project. The contingency and Fellowship provided by the Advanced Technology Cell has played a vital role in the completion and publication of this work.

References

1. Zhang, J., Jeong, C.K., Lee, G.Y., Kim, H.J.: Cluster-based Multi-path Routing Algorithm for Multi-hop Wireless Network. *International Journal of Future Generation Communication and Networking* (2009)
2. De, S., Qiao, C., Wu, H.: Meshed multipath routing with selective forwarding: an efficient strategy in wireless sensor networks. *Computer Networks*, 481–497 (2003)
3. Siddiqui, M.S., Amin, S.O., Kim, J.H., Hong, C.S.: MHRP: A Secure Multi-path Hybrid Routing Protocol for Wireless Mesh Network. In: *Military Communications Conference, MILCOM 2007*, Orlando, FL, p. 105 (October 2007)
4. Li, Z., Wang, R.: A Multipath Routing Algorithm Based on Traffic Prediction in Wireless Mesh Networks. *Communications and Network* 1(2), 82–90 (2009)
5. Lee, S.-W., Choi, J.Y., Lim, K.W., Ko, Y.-B., Roh, B.-H.: A Reliable and Hybrid Multipath Routing Protocol for Multi-Interface Tactical Ad Hoc Networks. In: *The Military Communication Conference*, pp. 1531–1536 (2010)
6. Radi, M., Dezfouli, B., Razak, S.A., Bakar, K.A.: LIEMRO: A Low-Interference Energy-Efficient Multipath Routing Protocol for Improving QoS in Event-Based Wireless Sensor Networks. In: *Fourth International Conference on Sensor Technologies and Applications, SENSORCOMM*, pp. 551–557 (2010)
7. Mueller, S., Tsang, R.P., Ghosal, D.: Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges. In: Calzarossa, M.C., Gelenbe, E. (eds.) *MASCOTS 2003*. LNCS, vol. 2965, pp. 209–234. Springer, Heidelberg (2004)
8. Ganjali, Y., Keshavarzian, A.: Load Balancing in Ad Hoc Networks: Single-path Routing vs. Multi-path Routing. In: *IEEE International Advance Computing Conference, IACC*, pp. 32–34 (2009)

Weighted Energy Efficient Cluster Based Routing for Wireless Sensor Networks

Soumyabrata Saha¹ and Rituparna Chaki²

¹ Dept. of Computer Sc. & Engg.
University of Calcutta, West Bengal, India
som.brata@gmail.com

² Dept. of Computer Sc. & Engg.
West Bengal University of Technology, West Bengal, India
rituchaki@gmail.com

Abstract. Wireless sensor network comprises of numerous tiny sensor nodes to form an ad hoc distributed sensing and data propagation network to collect the context information on the physical environment. Wireless sensor networks are used for remotely monitoring tasks and effective for data gathering in a variety of environments. Minimizing energy dissipation and maximizing network lifetime are one of the central concerns to design applications and protocols for wireless sensor networks. Cluster based architectures are one of the most practical solutions in order to cope with the requirements of large scale of sensor networks. Clustering results in a reduced number of messages that propagate through the network in order to accomplish a sensing task. In this paper, we have presented a brief review of the state of the art scenario of various routing protocols and propose a weighted energy efficient cluster based routing for wireless sensor networks. Conducted simulation experiments on different scenarios shown its ability to obtain effective & efficient communications among different clusters and achieves scalability in large scale of wireless sensor networks.

Keywords: Wireless Sensor Networks, Weighted, Energy, Cluster, Routing.

1 Introduction

Wireless sensor networks are formed by densely and usually randomly deployed large number of sensor nodes either inside or very close to the phenomenon that is being monitored. The potential applications of sensor networks are highly varied, such as natural phenomena, environmental changes, controlling security, estimating traffic flows, monitoring military application, and tracking friendly forces in the battlefields.

Wireless sensor networks consist of sinks and sensors. Sinks play a role of collecting data which is transmitted by sensors. Sensor nodes sense the desirable physical phenomenon and locally do the data aggregation to avoid communication of redundant data. Using routing protocol sensor nodes determine the path for sending data to sink. A sensor node is comprised of four basic components: sensing unit, processing unit, radio unit and power unit. The sensing unit is used to measures a

certain physical condition. Processing unit is responsible for collecting and processing signals. The radio unit transfers signals from the sensor to the user through the gateway. All previous units are supported by the power unit to supply the required energy in order to perform the mentioned tasks.

Clustering in wireless sensor networks provide scalability and robustness for the network. It allows spatial reuse of the bandwidth, simpler routing decisions, and results in decreased energy dissipation of the whole system by minimizing the number of nodes that take part in long distance communication. A cluster is a group of linked nodes working together closely for same purposes and belongs to same topological structure. A cluster head is responsible of resource allocation to all nodes belonging to its cluster and directly associated to its neighbor clusters for performing various task of intra and inter cluster communication. Sensor network used to be designed as clustering structure to minimize transmission costs and energy usage to prolong the network lifetime.

In this paper we propose a weighted energy efficient cluster based routing for wireless sensor networks. This proposed routing scheme comprises of five different steps. At first cluster head selection mechanism has been executed. The main motive of dynamic cluster head rotation mechanism is to evenly distribute the energy load among all the sensor nodes so that there are no overly utilized sensor nodes that will run out of energy before the others. Cost calculation technique has been introduced in second stage. Depends upon the dynamic cluster head selection and cost estimation technique, cluster formation mechanism has been presented at third step. Communication procedure has been described at step four. In step five route maintenance mechanisms take an important role to maintain the routes to deliver the messages from sender to receiver.

The rest of the paper is organized as follows. A comprehensive survey of related works of different routing techniques in Wireless Sensor Networks is presented in Section 2. In Section 3, we have design and describe a new routing algorithm, which incurs the transitions among these five different phases and improves energy efficiency to prolong the whole network lifetime. Energy Estimation mechanism is described in section 4. Intensive result analysis is presented in section 5. Finally, we conclude our paper with final remarks in Section 6.

2 Related Works

In recent years, cluster based architectures are one of the most suitable solutions to cope with the requirements of large scale wireless sensor networks. The clustering protocols have been extensively studied in this literature, which mainly differ in the selection of cluster heads, reformation strategies, hopping limits and routing scheme. In this section we take a brief look at some of the common clustering algorithms applicable for wireless sensor networks.

In HCBQRP [1] the main objective was to design a cluster based routing algorithm for sensor networks to find route from source to destination. HCBQRP [1] was designed based on the query-driven approach and provides the real time communication between sensor nodes.

Maximum Degree and Negotiation Strategy Based Clustering Algorithm for wireless sensor networks [2] was proposed by Qiang Wang et.al. MXAD-N [2] is

divided into two phases; one is communication and computation of node degree and other is cluster head election and cluster formation. MAXD-N [2] selects candidate based maximum degree and determines cluster head according to the negotiation strategy.

Babar Nazir et.al. proposed Mobile Sink Based Routing Protocol [3] for prolonging network lifetime in clustered wireless sensor network. In MSRP [3] mobile sink moves within cluster area to collect sensed data from cluster heads and maintains information about the residual energy of the cluster head.

In [4] Tian Ying et.al. presented Energy Efficient Chain Cluster Routing protocol for wireless sensor networks. ECRM [4] protocol divided into backbone setup phase, cluster formation phase and steady communication phase. ECRM [4] includes efficient cluster head selection mechanism to prolong the network lifetime and improve the energy efficiency.

Asif U. Khattak et.al. proposed a Two Tier Cluster Based Routing Protocol [5] for wireless sensor networks. TTCRP [5] configures the nodes in the form of clusters at two levels. TTCRP [5] introduced a power control algorithm to allow the isolated sensor nodes as well as cluster heads to dynamically change their transmission power and provides network robustness.

LU Li fang et.al proposed Weight Based Clustering Routing Protocol for wireless sensor networks [6]. In WCR [6] cluster head selection algorithm is designed for periodically select cluster heads based on the node position and residual energy. WCR [6] is used to minimize the energy dissipation of intra cluster and inter cluster data transmission.

In [7] Awwad et.al. presented adaptive time division multiple access scheduling and round free cluster head protocol called Cluster Based Routing Protocol for Mobile nodes in Wireless Sensor Network [7]. CBR Mobile-WSN [7] is energy aware scheme and used to handle packet loss and reduces the energy consumption.

An Energy Level Based Routing Algorithm of Multi sink Sensor Networks [8] was proposed by Zhongbo et.al. In ECR [8] two different hierarchy structure of the network topology are presented. The operation of ECR [8] can be divided into three steps, cluster-formation phase, cluster head selection phase and steady state phase.

Lu Cheng et.al. proposed an Energy Efficient Weight Clustering Algorithm [9] to reduce energy consumption by perfecting cluster formation procedure. In EWC [9] residual cluster energy, location and node degree and coefficients are takes an important role in cluster head selection stage and nodes' having minimal combined weight becomes cluster head.

In [10] A. Martirosyan et al. presented an Energy Efficient Inter Cluster Communication based routing protocol for WSNs. ICE [10] uses acknowledgement based approach to faulty paths discovery and provides QoS by finding a path with the least cost for high priority event notification messages.

An Adaptive Decentralized Re-clustering Protocol for wireless sensor networks [11] was proposed by Sethares et.al. ADRP [11] incurs high overhead in forming clusters due to the information exchanged between the nodes and the sink

Boukercheet. al. proposed Clustering PEQ [12], configures the dissemination tree using the PEQ's [12] mechanism and an additional field contains the percentage of nodes that can become cluster head. CPEQ [12] employs an energy aware cluster head selection mechanism in which the sensor nodes with more residual energy are selected as cluster head and increases the network lifetime.

Kyung Tae Kim et.al. presented an Energy Driven Adaptive Clustering Hierarchy for WSNs [13] to increase the lifetime of sensor networks. EDACH [13] protocol is based on partitioning of the network according to the relative distance to the base station for assigning different probability of cluster head.

In [14] Proxy Enabled Adaptive Clustering Hierarchy proposed by Kim et.al. A proxy node assumes as cluster head in place of weak cluster head during one round of communication. In PEACH [14], the clustering scheme avoids long range communication, data fusion and saves energy by compressing the data and rotation of cluster head allows to prolong the lifetime of every node.

An Energy Efficient Unequal Clustering [15] routing protocol was proposed by Hee Yong Youn et.al. EEUC [15] is a distributed competitive algorithm, where cluster heads are elected by locally and clusters are closer to the base station are expected to have smaller cluster sizes.

In [16], according to free space model, the propagation loss is modeled as inversely proportional to d^i (where $i=2$, if $d < d_0$). In the multi-path fading channel model [16], the propagation loss is modeled as inversely proportional to d^i (where $i=4$, if $d \geq d_0$). At the time of transmitting message, the amount of energy dissipated by sensor node has been defined as; $EP_{tx}(p, d) = EP_{elec} * p + \epsilon_{fs} * p * d^i$

To receive this p -bit message the energy dissipated; $EP_{rx}(p) = EP_{elec} * p$

Where p is the size of message, d is the distance between source and destination node, EP_{elec} is the circuit energy cost for transmitting or receiving purposes, ϵ_{fs} is the amplifier parameter.

In [17], the first order radio model is used, which consists of three main models: the transmitter, the power amplifier, and the receiver. The amount of energy consumed by each sensor node while transmitting message, is defined as

$$EP_{tx}(p, d) = EP_{elec} * p + EP_{amp} * p * d^2, \text{ where } EP_{amp} \text{ is the amplifier coefficient.}$$

Heinzelman et.al. proposed Low Energy Adaptive Clustering Hierarchy [17] algorithm for sensor networks. Two layers architecture was introduced in LEACH [17]. One used for communication within the clusters and the other was between the cluster heads and sink. Due to the random selection, there exists the probability of unbalanced cluster head selection and selected cluster head may be present in one part of the network, making other portion of the network unreachable.

Through the extensive literature review it is observed that the selection of cluster heads can greatly affect the performance of the whole network and well selected cluster heads not only decrease the energy consumption but also prolong the network lifetime. Energy efficiency is unanimously considered as core design issue and in order to improve the deficiencies of aforementioned schemes, the challenge is to develop a routing protocol that can meet these conflicting requirements while minimizing compromise. In the next section we are going to propose a new weighted energy efficient cluster based routing protocol for wireless sensor networks.

3 Proposed New Routing Protocol

The previous section leads to the observation of different cluster based routing protocol for wireless sensor networks. Clustering involves in the cluster formation techniques where low energy nodes are assigned the task of sensing. The main aim of

cluster based routing algorithm is to efficiently maintain the energy consumption of sensor nodes by involving them in multi hop communication.

The assumptions made for this protocol are as follows: all sensor nodes in the network are immobile, homogeneous and are equipped with power control capabilities; sensor nodes are generally energy constrained and they are capable to operate in an active mode or a low power sleeping mode and have enough processing power to support the different protocols and data processing tasks; each sensor node has a unique identifier and uniformly deployed over the target area to continuously monitor the environment; every sensor node can directly communicate with its immediate neighbor; the transmission range of each node is same on one condition.

Wireless sensor network is usually data centric. Collected data are periodically transmitted from the sensor node to the remote sink node. It is assumed that the sensed information is highly correlated, thus the cluster head can always aggregate the data gathered from its members into a single length fixed packet. The correlation degree of sensed data from different clusters is relatively low.

In the proposed weighted cluster based routing algorithm protocol, WSN perceived as a network partitioned into different clusters and we present a new robust approach of routing. The proposed weighted cluster based routing scheme is divided in some subsections. In section 3.1 cluster head selection procedures has been presented. Section 3.2 is used for cost calculation technique. In section 3.3 we have described cluster formation mechanism. Section 3.4 presented communication procedure. Route maintenance technique has been presented in section 3.5.

3.1 Cluster Head Selection Procedure

In our proposed scheme we introduce a weight based cluster head selection procedure. The main objective of choosing cluster head that guarantees both the intra-cluster and inter-cluster data transmission with energy efficient manner. Many energy consuming activity have to perform by the cluster heads, such as, data collections from member nodes and forwarding processed data to other neighbors or to the sink node. To select the cluster head, we have considered some parameters like; number of neighbor nodes, distance between cluster heads and sink node, degree difference of node; residual energy of sensor node is another important parameter for cluster head selection process. Every sensor node takes part in cluster head selection procedure. A node is selected as a cluster head and its energy level is periodically monitored. Once the energy level goes below the threshold value, it is released from its' special responsibilities.

We assume that N number of sensor nodes are scattered randomly over the area of interest and left unattended to continuously sense and report events. We denote the i_{th} sensor by s_i and the corresponding sensor node set $S = \{s_1, s_2, \dots, s_N\}$, where $|S|=N$, where N is the total number of sensor nodes in the network.

Definition 1: At time instance T_n , to send a p-bit message from transmitter to receiver at the distance d_i and to receive the p-bit message at same distance d_i , the total energy dissipated from node s_i is E_{req}

$$E_{req} = \{(E_{Pelec} * p + \epsilon_{fs} * p * d^2) + (E_{Pelec} * p)\} \quad (i)$$

And the remaining energy for node s_i is E_r

$$E_r = \{(E_{init}) - (E_{req})\} \quad (ii)$$

After calculating total dissipated energy and remaining energy of node s_i for time instance T_n , store all these different estimated energy in the ND_ENGY_TBL $\{S_i, E_{init}, E_{req}, E_r, T_n\}$ table.

Algorithm 1. Algorithm for Cluster Head Selection

- Step1: Find the neighbors of each node s_i
 $\dot{N}g\dot{N}_{s_i}[] = \sum \dot{s}_i \in \dot{N}, s_i \neq \dot{s}_i \{ dist(s_i, \dot{s}_i) \leq t_r \}$ (iii)
- Step2: Compute the distance ds_i for each node s_i as:
 $d_{s_i} = dist(s_i, d_{snk})$
- Step3: Compute the degree difference for every node s_i
 $\Delta s_i = \sum \dot{s}_i \in \dot{N}, [\{\dot{N}g\dot{N}_{s_i} - \dot{\epsilon}\} \leq t_r]$ (iv)
- Step4: Read initial energy E_{init} and residual energy E_r of node s_i for each time instance T_n from ND_ENGY_TBL and update this energy table.
- Step5: Calculate the total weight of each node s_i as:
 $w_{s_i} = [w_1 * \dot{N}g\dot{N}_{s_i} + w_2 * d_{s_i} + w_3 * \Delta s_i + w_4 * E_{req} + w_5 * T_{tach}]$ (v)
- Step6: Store all value of w_{s_i} in the $w_s[]$ and select minimum value of w_{s_i} as Ch_Id_i
- Step7: END.
-

By using the above technique, cluster head selection has been executed.

3.2 Cost Calculation

In this section we propose a method of calculating combined cost between various sensor nodes. After a brief survey of network environment, we are going to define some network parameters. These parameters reflect the cost of hopping to its neighbors, i.e., the cost of a link leading from the node to its neighbors. The parameters are described below:

- tp_i The throughput of sensor network. This can be estimated by counting total received and send packet by node i in a time interval.
- bw_i The available bandwidth of NIC of node i .
- sb_i Indicates the saturation of sending buffer at node i . If the length of buffer queue reaches 85% of its total size then sb_i set to “1”; otherwise sb_i set to “0”.
- rb_i Indicates the saturation of receiving buffer at node i . If the length of buffer queue reaches 85% of its total size then rb_i set to “1”; otherwise rb_i set to “0”.

By using the above parameters, we can estimate communication cost between two nodes as follows:

$$Cost_{comm} = \left[\left\{ \left(tp_i / bw_i \right) * w_s \right\} + sb_i + rb_i \right] \quad (vi)$$

The total cost can be calculated using this following equation:

$$Cost_{i,j} = [\{Cost_{comm}\} + \{Cost_{energy}\}] \quad (vii)$$

3.3 Cluster Formation Mechanism

After cluster head selection process is completed, each cluster head broadcast a CM_FRM_MSG {*Ch_Id, Msg_Id, Ch_Wg, TTL*} message to its neighbor nodes.

If a node receives this type of message from more than one cluster head, then it chooses to join the membership of the cluster head causing minimum cost. Accordingly, the node sends a reply CM_RPLY_MSG {*Ch_Id, Cm_Id, Ch_Wg, Msg_Id, Cost_{i,j}, TTL*} message to the corresponding cluster head with minimum total cost.

If any other cluster head receives this CM_FRM_MSG message from the other cluster head, then it will truncate the message and don't send any reply or acknowledgement to the sender cluster head. If any cluster member of other cluster again receives this message, then it will follow the same procedure. The total process is described below.

Algorithm 2. Algorithm for Cluster Formation

Step 1: Ch_{id_i} broadcast a CM_FRM_MSG message for cluster formation.

For each node in the network:

Step 2: If the node is already another cluster head or member of another cluster

Then it discards the previous message.

Else

Step I: Calculates total cost between it and Ch_{id_i}

Step II: If total cost < Th_{cost}

Then it sends a reply CM_RPLY_MSG message to Ch_{id_i}
and added to Ch_{id_i}_Mem [].

Else

Then it discards the previous message.

Step 3: END.

By using the above procedure any non-member nodes can join to the cluster head.

3.4 Communication Procedure

In this section we have discussed the communication procedure between sensor nodes. Every cluster head maintains the NGH_CLSRHD_TBL {*Ch_Id, Ch_Wg, Ch_Comm_Cost, TTL*} table and also maintains their cluster members' information in CM_INF_TBL {*Cm_Id, Msg_Id, Cm_Wg, Cost_{comm}*} table.

When any sensor node requires some information, it initiates communication. At first it communicates with its corresponding cluster head and cluster head searches in its CM_INF_TBL table. If required information is not available then it searches in NGH_CLSRHD_TBL table. The cluster head with lowest communication cost is selected as next hop. The algorithm is described below:

Algorithm 3. Algorithm for Communication Procedure

Step 1: Node N send a request message to corresponding Ch_i .
 Step 2: Ch_i searches in its CM_INF_TBL.
 Step 3: If required information is available
 Step I: Then it appends corresponding node's Cm_Id in RT_INFO table.
 Step II: That node sends message to sink and updates its energy level.
 Step III: Communication end.
 Else
 Step I: Searches in NGH_CLSRHD_TBL
 Step II: The Ch_i with lowest communication cost selected as next hop.
 Step III: Go to step 2.
 Step 4: END.

After successful execution of the above algorithm we find the node with required information. This node becomes source node and sends a message including the required information through selected path from RT_INFO {*Snd_Nd_Id*, *Nxt_Hp*, *Dst_Id*} table.

3.5 Route Maintenance Techniques

If in a particular route, a sensor node is damaged for a certain time period or link is broken, the route will be damaged. To handle this kind of situation we have incorporated route maintenance mechanism to overcome this problem.

After receiving any information from any sensor node, the receiving node has send ACK_MSG {*Snd_Nd_Id*, *Recvr_Nd_Id*, *Msg_Id*, *Ack_Id*, *Tm_Stmp*} to the sender node within a certain time period. These ACK_MSG message based repair mechanisms consist of two parts. One is failure detection and other is selection of alternative node.

Depends on the reception time of ACK_MSG sensor nodes can detect if its neighbor nodes are functioning properly or not. If ACK_MSG not received within a specified time, the sensor node initiates the communication procedure to find out the alternative route.

By using the above discussed algorithm any sensor node can communicate with any another nodes through the cluster head within that network.

Algorithm 4. Algorithm for Route Maintenance

Step1: If ACK_MSG is not received within specified time interval
 Then execute communication procedure for finding alternative route
 Else terminate the process.
 Step2: END.

Table 1. Data Dictionary

Parameter	Details
Ch_Id	Cluster head id
Ch_Wg	Weight of cluster head
Th _{cost}	Threshold cost
TTL	Time to leave
E _{dg}	Data aggregation cost
Recvr_Nd_Id	Receiver node id
t _r	Transmission range
Nxt_Hp	Next hop
Cm_Id	Cluster member id
Msg_Id	Message id
Dst_Id	Destination node id
Snd_Nd_Id	Sender node id
Ack_Id	Acknowledgement id
Tm_Stmp	Time stamp
d _{ch}	Average distance between cluster head & sink
T _{tach}	Time for select as cluster head
W _s []	Store all the calculated weight value of w_{s_i}
Ch_comm_cost	Communication cost of cluster head
Cm_Wg	Weight of cluster members
Ch_idi_Mem []	Node belongs to member list of cluster head.
d _{chm}	Average distance between cluster members & cluster head

4 Energy Estimation

Definition 2: In the proposed cluster based network, assuming there are n numbers of cluster members and each member having the capacity of sending at most p -bit message. The total number of the received message at the cluster head is m ;

$$m = n * p \quad (\text{viii})$$

$$E_{cm} = [EP_{elec} * p + \varepsilon_{fs} * p * d_{chm}^2] \quad (\text{ix})$$

Definition 3: Eqn (x) represents average energy consumption per cluster member, assuming there are n numbers of cluster members are present within the cluster.

$$E_{am} = \{1/n \sum_{i=1}^n E_{cm} (i)\} \quad (\text{x})$$

Definition 4: The energy consumed by a cluster head is obtained by Eqn (xi). E_{ch} is the energy consumption as the cluster head transmits the aggregated data to the sink node.

$$E_{ch} = EP_{elec} * m + \varepsilon_{fs} * m * d_{ch^2} \quad (xi)$$

Definition 5: Eqn (xii) represents average energy consumption per cluster head, assuming there are c number of clusters are present in the network. Since E_{ach} change over time, the threshold is calculated in every data collection and transmission phase.

$$E_{ach} = 1/c \sum_{i=1}^c E_{ch}(i) \quad (xii)$$

Definition 6: For our proposed weighted energy efficient cluster based routing scheme, the energy dissipated during a round is given by Eqn (xiii).

$$E_{chm} = [(EP_{elec} * m + \varepsilon_{fs} * m * d_{chm^2}) + (\sum_{i=1}^n \{EP_{elec} * p + \varepsilon_{fs} * p * d_{chm^2}\}) + (E_{dg} * n * p) + (EP_{elec} * c * m)] \quad (xiii)$$

Where c is the number of clusters present in the network and m is the number of transmitted messages.

Definition 7: Assuming that the sensor nodes are uniformly distributed, it can be shown that; where q^2 is the area of the monitoring filed.

$$d_{chm^2} = \int_0^{x_{max}} \int_0^{y_{max}} (x^2 + y^2) * \rho(x, y) dx dy \quad (xiv)$$

$$d_{chm^2} = \left\{ \left(q^2 / 2 * \Pi * r \right) * d \right\} \quad (xv)$$

Definition 8: The total energy dissipated in the network is equal to:

$$E_{tot} = [(\sum_{i=1}^n \{EP_{elec} * p + \varepsilon_{fs} * p * d_{chm^2}\}) + \sum_{i=1}^c E_{chm}(i) + (E_{dg} * m) + (EP_{elec} * c * m)] \quad (xvi)$$

The total consumed energy of the sensor network is depends up on the construction of the cluster, if the clusters are not constructed in an optimal way, the total consumed energy of the sensor network per round is increase.

5 Result Analysis

The simulation model consists of a network model that has a number of wireless nodes, which represents the entire network to be simulated. The network simulator, NS-2 is used to evaluate the performance of the proposed protocol WECRP and present the following metrics for comparing the performance of WECRP with the AODV.

End2End delay is the average difference between the time the first data bit is originated by an application and the time this data bit is received at this destination. Throughput is defined as the total number of message sent or received per time unit. In this experiment, throughput is measured in terms of bits delivered per time unit. End2End of the data packet is one of the performances metric in our proposed scheme. The results for this metric are shown in fig 1.

Table 2. Simulation Environment Parameters

Parameter	Value
Channel	Wireless Channel
Propagation	Two Ray Ground
Antenna	Omni Antenna
Phy	Wireless Phy
Mac	802_11
Number of nodes	10
WSNs field	(0,0)~(500,500)m
Initial energy	0.5 J
Data package size	2100 bits

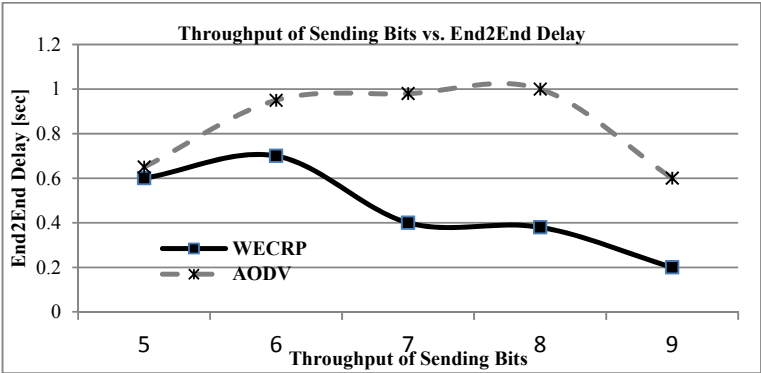


Fig. 1. Throughput of Sending Bits vs. End2End Delay

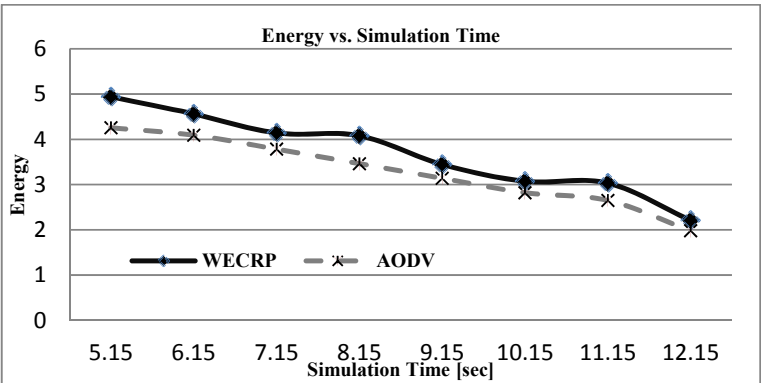


Fig. 2. Energy vs. Simulation Time

From this graph we have observed that in all situations the delay of WECRP is lesser than AODV. Though our proposed protocol induces a lower routing load than AODV, as a result data transmissions are less delayed. Thus our approach with lower End2End delay offers better QoS.

In fig 2, we have depicted a relation between energy vs. simulation time. It has been observed that when more number of packet or data bit is transferred, corresponding energy are less dissipated compared to AODV.

6 Conclusions

In this paper we have summarized the generic characteristics of some well-known cluster based routing protocols for WSNs and present a new routing scheme, weighted energy efficient cluster based routing for Wireless Sensor Networks to achieve real-time communication and high energy efficiency. As the cluster head of each cluster acts as a local coordinator for its cluster, performing inter-cluster routing, data forwarding and has to undertake heavier tasks so that it might be the key point of the network. In our proposed routing scheme only cluster head can take part to share information with their neighboring cluster heads. In WECRP we have considered weight factor of every sensor nodes to select the cluster head. According to WECRP nodes having least weight factor will select as a cluster head. Using cost estimation technique each cluster head selects its neighbor cluster members. Within the same cluster or different clusters, communication procedure has been carried out and route maintenance takes an important role in this routing algorithm.

The routing of messages using the proposed WECRP is achieved in an efficiency manner as observed in the performance graph. The parameters used for performance analysis are energy efficiency, transmission delay and throughput. A comparative analysis of the proposed WECRP against AODV has been carried out. It has been observed that the proposed method shows better performance in each cases and achieved energy efficiency, scalability, information sharing and route maintenance mechanism.

References

1. Saha, S.B., Chaki, R.: HCBQRP: Hierarchical Cluster Based Query-Driven Routing Protocol for Wireless Sensor Networks. In: SPRINGER International Conference on Information Systems Design and Intelligent Applications, INDIA (2012)
2. Wang, Q., Wang, C., Wang, Y.: A Maximum Degree and Negotiation Strategy Based Clustering Algorithm for Wireless Sensor Networks. In: IEEE Instrumentation and Measurement Technology Conferences, I2MTC (2011)
3. Nazir, B., Hasbullah, H.: Mobile Sink based Routing Protocol for Prolonging Network Lifetime in Clustered Wireless Sensor Network. In: IEEE International Conference on Computer Applications and Industrial Electronics, ICCAIE (2010)
4. Ying, T., Yang, O.: A Novel Chain-Cluster Based Routing Protocol for Mobile Wireless Sensor Networks. In: 6th IEEE International Conference on Wireless Communications Networking and Mobile Computing, WiCOM (2010)

5. Khattak, A.U., Shah, G.A., Ahsan, M.: Two-Tier Cluster Based Routing Protocol For Wireless Sensor Networks. In: IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing, EUC (2010)
6. Lu, L.F., Lim, C.-C.: Weight-based Clustering Routing Protocol for Wireless Sensor Networks. In: IEEE International Symposium on IT in Medicine & Education, ITIME (2009)
7. Awwad, S.A.B., Ng, C.K., Noordin, N.K., Rasid, M.F.A.: Cluster Based Routing Protocol for Mobile Nodes in Wireless Sensor Network. In: IEEE International Symposium on Collaborative Technologies and Systems, CTS (2009)
8. Wu, Z., Zhang, C., Chen, H.: Energy level based Routing Algorithm of Multi sink Sensor Networks. In: IEEE International Conference on Networking, Sensing and Control, ICNSC (2008)
9. Cheng, L., Qian, D., Wu, W.: An Energy Efficient Weight-clustering Algorithm in Wireless Sensor Networks. In: IEEE Japan-China Joint Workshop on Frontier of Computer Science and Technology (2008)
10. Boukerche, A., Martirosyan, A.: An Energy Aware and Fault Tolerant Inter cluster Communication based Protocol for Wireless Sensor Networks. In: Globecom, Washington, D.C. (2007)
11. Wen, C.Y., Sethares, W.A.: Adaptive Decentralized Re-Clustering for Wireless Sensor Networks. In: IEEE International Conference on Systems, Man and Cybernetics, SMC (2006)
12. Boukerche, A., Pazzi, R.W., Araujo, R.B.: Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments. *Journal of Parallel and Distributed Computing* 66(4), 586–599 (2006)
13. Kim, K.-T., Youn, H.Y.: Energy-Driven Adaptive Clustering Hierarchy (EDACH) for Wireless Sensor Networks. In: Enokido, T., Yan, L., Xiao, B., Kim, D.Y., Dai, Y.-S., Yang, L.T. (eds.) *EUC Workshops 2005*. LNCS, vol. 3823, pp. 1098–1107. Springer, Heidelberg (2005)
14. Kim, K.T., Youn, H.Y.: PEACH: Proxy-Enable Adaptive Clustering Hierarchy for Wireless Sensor Networks. In: *Proceedings of ICWN 2005*, pp. 52–56 (2005)
15. Youn, H.Y., Kim, K.T.: An Energy Efficient Unequal Clustering Mechanism for Wireless Sensor Networks. In: *IEEE International Conference on Mobile Ad hoc and Sensor Systems Conference* (2005)
16. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: An Application-Specific Protocol Architecture for Wireless Micro sensor Networks. *Wireless Communications* 1(4), 660–670 (2002)
17. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Micro sensor Networks. In: *Proc. 34th Hawaii Int'. Conf. Sys. Sci.* (2000)

A Business Process Modeling Notation Extension for Risk Handling

Bartosz Marcinkowski and Michal Kuciapski

University of Gdansk, Department of Business Informatics

Piaskowa 9, 81-864 Sopot, Poland

{bartosz.marcinkowski,michal.kuciapski}@ug.edu.pl

Abstract. During the years of prosperity, numerous organizations neglected numerous aspects of risk management. As systematic approach to handling identified risks is crucial to achieving success by the organization, modern business modeling standards and techniques are supposed to take risk-related features into account. The article is devoted to elaborating and exemplifying an extension aimed at risk handling for OMG's Business Process Modeling Notation (BPMN), one of the most prospective standards for business process modeling. After an introduction, key risk management concepts are discussed. Section 3 discusses extensions introduced within BPMN meta-model, while section 4 exemplifies proposed concepts. The article is concluded with a summary.

Keywords: Business Process Modeling Notation, Risk Management, BPMN Extension.

1 Introduction

Continuous and lasting for decades evolution of all-purpose and domain-centric notations resulted in creating modern modeling standards, such as Unified Modeling Language (UML), Business Process Modeling Notation (BPMN) or BPMS method, integrated in ADONIS business management solution. There is however plenty of room for further improvement. Needless to say, a few BPMN extensions were proposed by academic community and business modeling community.

Early versions of BPMN – although often classified by practitioners as powerful on their own (comp. Harrison-Broninski, 2006) – were enriched with business process goals as well as performance measures by Korherr and List (2007). (Rodriguez, Fernandez-Medina and Piattini, 2007) presented a BPMN 1.0 meta-model with core element and extension that allowed incorporating security requirements into Business Process Diagrams aimed at increasing the scope of the expressive ability of business analysts. (Magnani and Montesi, 2009) addressed relevant limitations of BPMN regarding weak data representation capabilities in comparison to competing standards, designed or adapted to business process modeling needs – such as ADONIS BPMS method or OMG's Unified Modeling Language profiles and custom approaches (comp. Przybylek, 2006). Some of the concepts proposed were included later in

BPMN 2.0 specification (Object Management Group BPMN, 2011), which was a significant revision that led the emerging standard the UML-alike multi-diagram way (comp. Wrycza, Marcinkowski and Wyrzykowski, 2005). (Stropi, Chiotti and Villarreal, 2011) offered solutions to strengthen the resource perspective of a business process model elaborated using BPMN 2.0 in order to improve the communication of resource structure, authorization and work structure between business analysts and technical developers. (Zor, Schumm and Leymann, 2011) enhance BPMN 2.0 modeling capabilities within manufacturing domain.

Business Process Modeling Notation extension capabilities are to be used by Object Management Group itself as a UML Profile for BPMN Processes Request For Proposal document (Object Management Group RFP, 2011) was issued. Having that said, the profile is not intended to develop notational capabilities but to provide a mapping between BPMN semantics and the profiled UML semantics as well as define XSLT transforms between the UML XMI for the profile and the BPMN 2 XSD and QVT transforms between the UML and BPMN 2 meta-models.

During the years of prosperity, numerous organizations neglected numerous aspects of risk management. As systematic approach to handling identified risks is crucial to achieving success by the organization, modern business modeling standards and techniques are supposed to take risk-related features into account. The article is devoted to elaborating an extension aimed at risk handling for Business Process Modeling Notation, one of the most prospective standards for process modeling.

2 Basic Concepts of Risk Management

According to (ISACA, 2006), risk management is the process of identifying vulnerabilities and threats to the information resources used by an organization in achieving business objectives, and deciding what countermeasures, if any, to take in reducing risk to an acceptable level, based on the value of the information resource to the organization. Risk itself is defined as the combination of the probability of an event and its consequence (International Organization for Standardization, 2009).

Risks may be divided into (International Business Machines Corporation, 2007):

- business-driven risks, strategic in nature and aimed at protecting the business and keeping it accessible whenever and from whoever in support of continuous business operations as well as compliance with industry and government regulations;
- data-driven risks, dealing with the availability of data and information in all of its different forms as used by the organization, including paper-based data;
- event-driven risks, focusing on actual events that create risk to business continuity and viability, such as natural disasters, thefts and IT attacks.

It is risk mitigation procedure that is one of the relevant aspects of risk management to be included in the elaborated extension. Based on the canon of literature in the field of risk management, four standard ways to handle risk are commonly acknowledged (comp. DeLoach and Temple, 2000):

- reduce,
- retain,
- avoid,
- transfer.

(Husdal, 2009) proposes to set up a wider framework of risk management by discussing exploit and ignore strategies and adding it to the list.

3 Risk Handling in BPMN – Standard Features and Extensions

Distinctive features of BPMN include, in particular, very extensive semantics of events. In addition to the division of events into start, intermediate and end ones, the notation development team proposed twelve types of events, distinguished events that are thrown and caught as well as proposed the possibility of interrupting or continuing the flow of source activity at the time of event occurrence. It is the error event that is particularly important type of event from the perspective of risk management. Such event type points to an exception in the underlying activity. The functionality of the described event may in fact be used to assign identified risks to processes, sub-processes or activities to which these risks apply. Indication of potential points of error and how to design their handling is therefore the starting point of expansion of the standard BPMN business process diagram.

BPMN standard, however, does not support formal specification of the identified risks. A business analyst can attempt fulfilling that task by introducing text annotations, which are among the build-in BPMN artifacts, but it is a technique of low clarity and precision. Therefore, the standard was extended with the modeling category of risk factor, characterizing a potential risk in terms of the type, likelihood and impact on business process as a whole. Analogously as in the related publications (Kuciapski and Marcinkowski, 2011), (Kuciapski, 2010) to both the likelihood and impact ranks from 1 to 5 range were assigned, with a value of 1 indicates a low occurrence probability (impact), while the value of 5 – very high occurrence probability (impact).

Risk factor is designed as an independent modeling category due to its complex nature. From the perspective of BPMN meta-model, it was reasonable to assign it to a group of artifacts. Therefore, the *RiskFactor* is implemented as a child element of *Artifact*, and supplemented with the additional properties – *occurrenceProbability* and *impact* (see Fig. 1). A single *RiskFactor* can be associated with multiple types of risks, but the specification of at least one *RiskType* is mandatory. Types of risk are classified as integral parts of risk factors, as highlighted with the use of the composition. *RiskType* is treated in terms of an abstract modeling category. Proposed extension of the standard recognizes five types of risks, but this list may be expanded according to the needs of the end user. Each proposed type of risk is assigned an individual notation. *PhysicalResourceRisks* can be related to a list of *Resources*, as *HumanResourceRisk* – a list of *Participants*.

Risk factors are assigned to BPMN sequence flows. For this purpose, a standard modeling category of association or placement directly by the relevant sequence flow

is used. It should be emphasized that with version 2.0 of the BPMN notation, an *Association* is distinguished from a *DataAssociation*, hence the association in the current version of the standard is not document-oriented. A single identifiable *RiskFactor* can be attributed to multiple *SequenceFlows*. Naturally, from the standpoint of a sequence flow, binding risk factors is optional. The extension also includes the possibility of decomposing risk factors onto component risk factors.

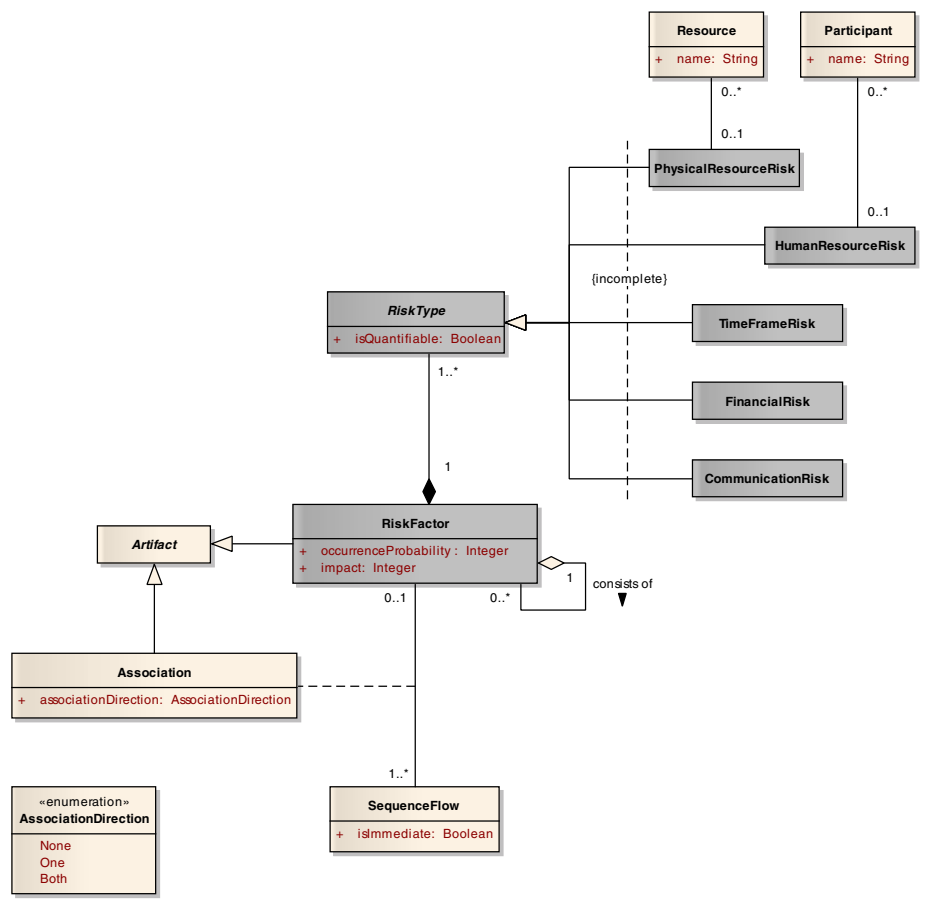


Fig. 1. Risk factors and BPMN meta-model

Each identified risk ought to be assigned with a solution. In the proposed extension, tasks dedicated to handle risk factors are distinguished from tasks forming a natural flow of process or sub-process. In order to achieve that, BPMN meta-model is supplemented with additional kind of *Task*, i.e. *RiskHandler*, along with accompanying markup notation (Fig. 2). In order to provide basic compatibility with ADONIS risk analysis process extension for proprietary BPMS notation, both

RiskHandlers and *RiskFactors* introduce icons elaborated within mentioned solution. Since a *Task* is a special case of an *Activity*, *RiskHandler* inherits the characteristics of activities. In the context of risk management the significance of *Resources* assigned to *Activities* – *Contractors* in particular – should be emphasized.

It is the *mitigationMethod* that is an integral property of a *RiskHandler*. Based on the list of strategies included in section 2 of the current article, six ways to handle risk are proposed: *Reduce*, *Retain*, *Avoid*, *Transfer*, *Exploit* and *Ignore*. As the method is expressed as a String, an enumeration called *RiskMitigationMethod* is designed for the purpose of storing potential values of the property.

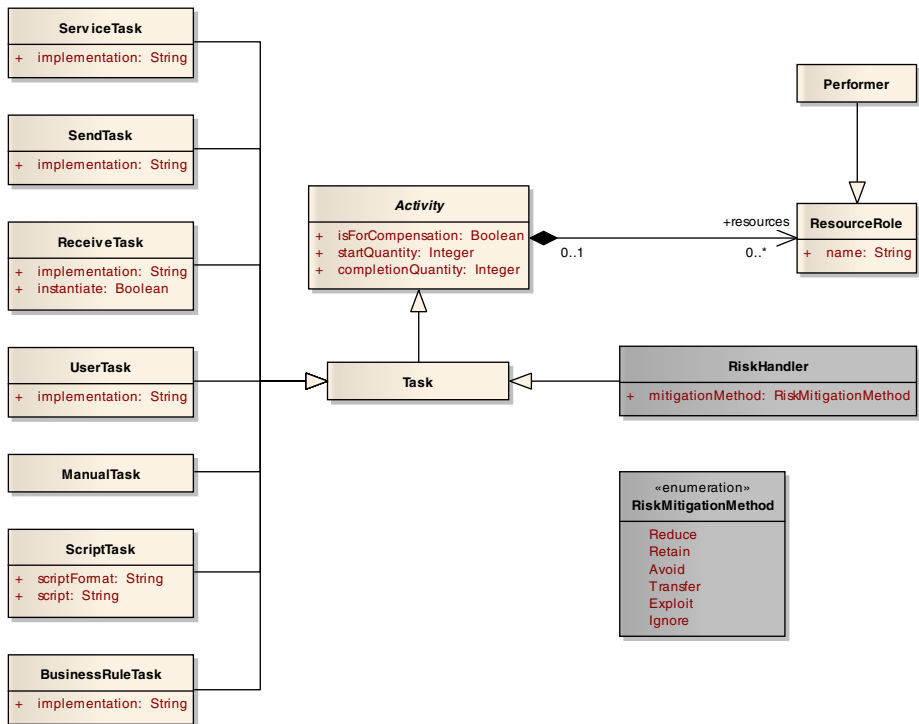


Fig. 2. Risk handler as a child element of BPMN Task

4 BPMN Extension for Risk Handling Exemplified

The proposed extension was tested within business process modeling project, involving specification of diverse business processes for real estate developer companies. It is a sub-process that illustrates the procedure for managing architectural contests that is the business functionality selected for the current paper (Fig. 3).

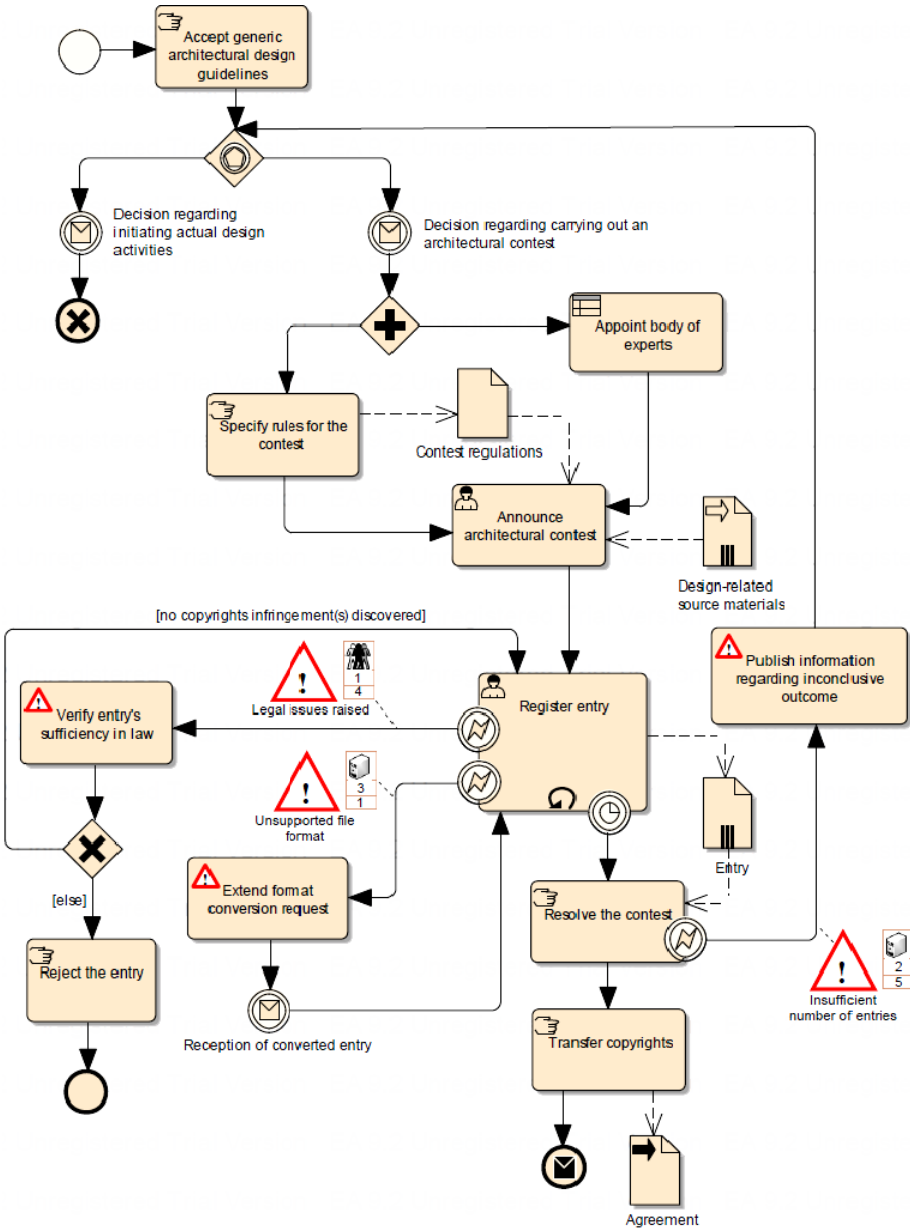


Fig. 3. Procedure for managing architectural contests developed using BPMN Extension for Risk Handling

Analysis of the initial BPMN diagram reveals that it is legitimate to consider a number of risks that the company, responsible for the investment, should foresee in the context of conducting the architectural contest. Thus, risks itemized by the

company included the risk of insufficient number of entries to the contest, receiving entries that are produced using non-standard toolkits (and thus containing files in non-supported graphics formats), or occurrence of legal uncertainties as to the authorship of the project. These risks are addressed by risk factors, respectively, *Insufficient number of entries*, *Unsupported file format* and *Legal issues raised*. It should be noted that within the proposed solution of the risk management issues, each identified risk factor is to be attributed to a separate event. Thus, on the border of an activity a lot of intermediate events, responsible for initiating various factors, may be placed.

Two former risk factors are classified as *PhysicalResourceRisks*, while the latter is classified as a *HumanResourceRisk*. Risk factor *Insufficient number of entries* is rather unlikely to occur (rank 2), but has a great impact on the owning sub-process (5). It is *Publish information regarding inconclusive outcome* that is the task devoted to handling the risk factor. On the other hand, receiving *Unsupported file format* is more common (rank 3) while having virtually no impact on the sub-process at all (rank 1). Should the risk factor occur, risk handler *Extend format conversion request* is invoked. Risk factor *Raising legal issues* is very rare (rank 1) but has significant severity (rank 4). A task *Verify entry's sufficiency in law* was designed to handle the risk.

5 Summary

It was elaborating an extension aimed at risk handling for Business Process Modeling Notation that was the goal of the current article. As BPMN functionality for risk modeling is very limited, even basic risk-oriented framework required introducing custom modeling categories, i.e. *RiskFactors*, *RiskTypes*, *RiskHandlers* as well as *RiskMitigationMethods*. Owing to the consistent practice of publishing meta-models for standards maintained by Object Management Group, the categories were seamlessly integrated with BPMN meta-model and designed so that the subsequent expansion was possible. Practical applications of proposed extension were exemplified by illustrating the procedure for managing architectural contests.

References

1. DeLoach, J.W., Temple, N.: Enterprise-wide Risk Management: Strategies for Linking Risk and Opportunity. Financial Times Prentice Hall (2000)
2. Harrison-Broninski, K.: The Future of BPM. Part 2 (2006), <http://www.bptrends.com/publicationfiles/09-06-ART-FutureBPM20f6-Harrison-Broninski.pdf>
3. Husdal, J.: The Six Ways of Dealing with Risk (2009), <http://www.husdal.com/2009/06/13/the-six-ways-of-dealing-with-risk/>
4. International Business Machines Corporation: Risk Mitigation for Business Resilience White Paper. A Comprehensive, Best-Practices Approach to Business Resilience and Risk Migration (2007), <http://www-935.ibm.com/services/pl/gts/html/pdf/gmw14000-usen-00.pdf>

5. International Organization for Standardization: ISO Guide 73:2009 (2009), http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=44651
6. ISACA: CISA Review Manual 2006. Information Systems Audit and Control Association (2006)
7. Korherr, B., List, B.: Extending the EPC and the BPMN with Business Process Goals and Performance Measures. In: 9th International Conference on Enterprise Information Systems. ACM Press (2007)
8. Kuciapski, M., Marcinkowski, B.: Risk-oriented Modeling in Business Process Specification (in Polish). In: The Risk of Business Ventures. Foundation for University of Gdansk Development (2011)
9. Kuciapski, M.: Risk Management in e-Learning Projects of Courses Development and Implementation. In: Project Management. Selected Issues, Studies and Materials of Polish Society of Knowledge Management (2010)
10. Magnani, M., Montesi, D.: BPDMM: A Conservative Extension of BPMN with Enhanced Data Representation Capabilities. In: Proceedings of CoRR (2009)
11. Object Management Group: Business Process Model and Notation (BPMN) (2011), <http://www.omg.org/spec/BPMN/>
12. Object Management Group: UML Profile for BPMN Processes RFP (2011), <http://www.omg.org/cgi-bin/doc?ab/10-06-01>
13. Przybylek, A.: The Integration of Functional Decomposition with UML Notation in Business Process Modeling. In: 15th International Conference on Information Systems Development (2006)
14. Rodriguez, A., Fernandez-Medina, E., Piattini, M.: A BPMN Extension for the Modeling of Security Requirements in Business Processes. IEICE Trans. Inf. & Syst. E90-D (2007)
15. Stroppi, L.J.R., Chiotti, O., Villarreal, P.D.: A BPMN 2.0 Extension to Define the Resource Perspective of Business Process Models. In: CIBSE 2011 – Congresso Ibero-Americano em Engenharia de Software (2011)
16. Wrycza, S., Marcinkowski, B., Wyrzykowski, K.: UML 2.0 in Information Systems Modeling (in Polish). Helion (2005)
17. Zor, S., Schumm, D., Leymann, F.: A Proposal of BPMN Extensions for the Manufacturing Domain. In: Proceedings of 44th CIRP International Conference on Manufacturing Systems (2011)

Modeling Consumer Decision Making Process with Triangular Norms

Agnieszka Jastrzebska and Wladyslaw Homenda

Faculty of Mathematics and Information Science
Warsaw University of Technology
Plac Politechniki 1, 00-660 Warsaw, Poland
A.Jastrzebska@mini.pw.edu.pl
<http://www.mini.pw.edu.pl/~homenda>

Abstract. Consumer decision making processes are conditioned by various forces. Recognized premises are being constantly reevaluated and future decisions are made in connection with previous ones. Therefore, authors propose an approach to decision making modeling based on pairs of vectors describing attitudes towards certain attributes influencing consumer's decision. As a result, possible is a description of preferences based on multiple evaluations of one feature. Methodological approach allows to reevaluate opinions, which even though were taken in past, but still influence current decisions. In the article we discuss how triangular norms can be applied to decision making modeling based on information encouraging the choice gathered in paired vectors. Developed methodology is based on combining different known triangular norms for given pairs of vectors representing various consumers, who are facing the same decision. The choice of preferences evaluation was performed so that they would represent different real-life situations. Results of applied processing based on combinations of triangular norms are compared. Drawn are conclusions about various processing properties of aggregation operators. Suggested is, in which cases, certain triangular norms describe real-life processes accurately. Shown and described are also examples of operators, which are not suitable at all.

1 Introduction

Decision making processes are driven by various needs. Needs determine our actions, direct us towards reducing motivational tension. As we described in [6], proposed were several hierarchies of how humans proceed with satisfying their needs. A very interesting perspective on the grounds of decision making processes was presented by K. Lewin, [8]. This article continues the discussion started in [5] of how the theory of psychophysical field can be applied to consumer decision making modeling using fuzzy sets and their generalizations.

First, we would like to recall basic notions of how a decision making process can be perceived as a process inside the psychophysical field and how multi-valued logic operators can be applied to model it. In section 3 we discuss various triangular norms, which are applied in section 4 to modeling. Presented is a case

study of one particular choice. Decisions are based on fuzzy information, which in fact is very common. Developed methodological approach allows us to reevaluate consumer's opinions, which even though had been taken in past, but still influence current decisions. We compare aggregation possibilities incorporated in different triangular norms and suggest, which cases describe real-life processes most accurately.

2 Preliminaries

Psychophysical field is an abstract term, describing decision making process. It is formed as a combination of three factors:

1. The field (which can be perceived as a set of conditions). Field contains all motivational stimuli arising when subject thinks about the decision. Conditions of the decision are bounded by available knowledge and previous experiences. The field includes not only premises speaking for or against given choice, but also all conscious and subconscious factors, which may influence the decision.
2. Processes, which describe a way of how one makes decisions. It is directly connected with individual's behavioral patterns. Processes are conditioned by customer's cognitive abilities, rationality and susceptibility to behavioral biases.
3. The decision - called also *the gestalt* - it is the outcome of the decision making process.

In this article we would like to focus on how triangular norms can be applied to decision making modeling. We would restrict our discussion to positive premises only, though our future research will include bipolar information aggregation techniques. Due to this restriction, we would be able to model only some fraction of real-life processes, but presented techniques can be also adapted for cases based on both positive and negative premises.

The field (all conditioning factors) will be grouped into two sets: premises and priorities. These are all recognized motivational stimuli, which ground our decisions. Processes are represented by triangular norms, which we apply. The decision (gestalt) is the output of our computations.

As introduced nomenclature may raise some questions, at first, we would like to explain used notions. While speaking about decision making process, we would describe perceived stimuli in the terms of premises and priorities. A premise, in our understanding, is received and decoded information influencing given decision. We assume that one can evaluate the importance of a premise as a number from the range $[0, 1]$, where the greater is the value of the premise, the stronger is positive influence of the premise on the decision. This allows us to apply a flexible scale of evaluation. For each customer we discuss the same set of premises (the decision is made about the same product), but with different evaluations.

While computing the result of the decision making process we introduce aggregating operators, which allow to obtain a decision as a number from the $[0, 1]$

range. The greater is the output of our computations, the more convinced is given customer about the decision. What did we gain? First of all, discussed models describe real life situations better. We are able to base and compute imprecise information. Even though we may calculate the decision in the crisp form (binary response), we are intentionally highlighting that plenty of real-life decisions are perceived rather as weak or strong attitudes.

Premises describe customer's motivation towards certain products or services. In terms of needs theory (see [6]) premises are motivational stimuli, which elicit, control and sustain certain behaviors. These are all factors, which arise when an individual thinks of given decision in more general terms. It can be somehow called an initial or an *a priori* motivation. Authors treat all premises as a set of infinite amount of opinions or attitudes, from which while discussing a particular decision accounted and considered is only a relatively small and countable subset. Why small? Because people tend to simplify rather than complicate reality. Moreover, facing time constraints, people are aware that in order to efficiently manage one's time, even though there might be million possibilities, only a few are really worth discussing.

Second term present while discussing decision making processes are priorities. Authors intend to use this term in the context of a second set of beliefs (or in other words as a second set of motivational stimuli). Priorities allow us to take into account reassessed attitude towards one particular choice. In this paper, term premises concerns more general case, while priorities describe certain precise choices. Analogically, priorities can be expressed scaled, for example as a number from the interval $[0; 1]$. Priorities are recognized and evaluated later than premises, and their purpose is to provide a perspective of how one particular choice satisfies one's needs. In this context, they may be perceived as an *aposteriori* motivation, arising when subject has faced particular product or service. Of course, a set of priorities evaluations might be drastically different than premises.

3 Triangular Norms

In this section we recall basic notions of fuzzy sets and generalization of fuzzy connectives maximum and minimum to triangular norms. We will be expressing fuzzy sets in the form of membership functions. Namely, a fuzzy set A defined in the universe X is a mapping $\mu : X \rightarrow [0, 1]$ or $\mu_{A,X} : X \rightarrow [0, 1]$ if the names of the set and the universe should be explicitly stated.

The Zadeh's model of fuzzy sets can be described as a system similar to set theory $(F(X), \cup, \cap, -)$, where $F(X)$ denotes fuzzy sets over the universe X and $\cup, \cap, -$ denote union, intersection and complement. This system is clearly interpreted as $([0, 1]^X, \max, \min, 1-)$, where $[0, 1]^X$ denotes all mappings from the universe X into the unit interval $[0, 1]$, i.e. the space of membership functions, and \max, \min and $1-$ applied to membership functions implement union, intersection and complement. We do not pay attention to the interpretation of fuzzy sets in terms of a lattice L^X .

In this paper study of fuzzy sets system was enriched with triangular norms used in place of the max and min operators. Note: max and min are triangular norms as well.

Triangular norms, i.e. t-norms and t-conorms, are mappings from the unit square $[0, 1] \times [0, 1]$ onto the unit interval $[0, 1]$ satisfying axioms of associativity, commutativity, monotonicity and boundary conditions (cf. [7,9] for details), i.e.:

Definition 1. *t-norms and t-conorms are mappings $p : [0, 1] \times [0, 1] \rightarrow [0, 1]$, where p stands for both t-norm and t-conorm, satisfying the following axioms:*

1. $p(a, p(b, c)) = p(p(a, b), c)$ *associativity*
2. $p(a, b) = p(b, a)$ *commutativity*
3. $p(a, b) \leq p(c, d)$ if $a \leq c$ and $b \leq d$ *monotonicity*
4. $t(1, a) = a$ for $a \in [0, 1]$ *boundary condition for t-norm*
 $s(0, a) = a$ for $a \in [0, 1]$ *boundary condition for t-conorm*

t-norms and t-conorms are dual operations in the sense that for any given t-norm t , we have a dual t-conorm s defined by the De Morgan formula

$$s(a, b) = 1 - t(1 - a, 1 - b)$$

and vice-versa, for any given t-conorm s , we have a dual t-norm t defined by the De Morgan formula

$$t(a, b) = 1 - s(1 - a, 1 - b)$$

Duality of triangular norms causes duality of their properties. Note that the min/max is a pair of dual t-norm and t-conorm. The selected known t-norms and their dual t-conorms are given in Table 1. Note that Hamacher product is the chosen representative of the parametric class of Hamacher norms. t-norms and t-conorms are bounded by minimum t-norm and maximum t-conorm, i.e. for any t-norm t , any t-conorm s and any $x, y \in [0, 1]$:

$$t(x, y) \leq \min(x, y) \leq \max(x, y) \leq s(x, y) \quad (1)$$

We will be discussing consumers' decision making process modeled with triangular norms. In this study consumers' decision making is interpreted in the following way:

- first, a consumer attempts purchasing a product or a service considering general needs for it. This solicitude results in a series of *necessity factors* corresponding to needs. The *necessity factors* are expressed as values of a membership functions and we call them *premises*
- then, (s)he confronts the general needs with properties of a concrete item or offer, which fits the general need to some extent. The confrontation produces *fitting factors*, again expressed as values of a membership function. We call the fitting factors *priorities*,
- we assume that the property of a given item/service cannot increase corresponding need, it rather may soften the need. We say that priorities *moderate* premisses and implement the moderation with applying a t-norm. Note that according to formula 1 the result of moderation cannot exceed weaker of the premise and the priority,

Table 1. Selected triangular norms, dual t-norms and t-conorms are placed in one row

t-norm		t-conorm	
minimum	$\min(x, y)$	maximum	$\max(x, y)$
product	$x \cdot y$	probabilistic sum	$x + y - x \cdot y$
Lukasiewicz	$\max(0, x + y - 1)$	bounded sum	$\min(a + b, 1)$
nilpotent minimum	$\begin{cases} \min(x, y) & \text{if } x + y > 1 \\ 0 & \text{otherwise} \end{cases}$	nilpotent maximum	$\begin{cases} \max(x, y) & \text{if } x + y < 1 \\ 1 & \text{otherwise} \end{cases}$
drastic	$\begin{cases} y & \text{if } x = 1 \\ x & \text{if } y = 1 \\ 0 & \text{otherwise} \end{cases}$	drastic	$\begin{cases} y & \text{if } x = 0 \\ x & \text{if } y = 0 \\ 1 & \text{otherwise} \end{cases}$
Hamacher product	$\begin{cases} 0 & \text{if } x = 0 = y \\ \frac{x \cdot y}{x + y - x \cdot y} & \text{otherwise} \end{cases}$	Einstein sum	$\frac{x + y}{1 + x \cdot y}$

- finally we aggregate moderated premisses and priorities with a t-conorm. We assume here that needs vote for purchasing, therefore they either strengthen each other, or at least not weaken themselves. Note that the formula [1](#) shows that the aggregation with any t-conorm will produce a result not weaker than the stronger moderated premise/priority.

In this study we consider only positive premisses. A discussion on negative premisses, i.e. premisses which vote against purchasing, is not in the scope of this paper. We only wish to note that initial discussion on this subject was undertaken in [5](#). That attempt, utilizing balanced fuzzy sets [4](#) in decision making, combines both types of premisses: positive and negative. In frames of another strive positive and negative premisses are aggregating separately and then final decision is made on these aggregation. So called intuitionistic fuzzy sets [11,2](#) or vague fuzzy sets [3](#) can be used as vehicles for such aggregation.

4 Case Study

To be able to compare described in section [3](#) models and observe how their properties are reflected while modeling decision making processes, we introduce a case study. In this paper we will continue a discussion on a decision making process regarding purchase of a car (by analogy to: [5](#)). In order to be able to compare results, we will discuss the same set of eight pairs of premisses and priorities for five different customers.

We will be analyzing following attributes encouraging purchase of a car:

- if you have to take care of babies, it is easier to transport them in a car than by public transport,
- shopping with a car is very convenient,
- in a city, where decision maker lives, car allows you to go faster than by bus or by tram,
- having a nice car manifests consumer's good taste and his wealth,
- with a car you can easily make weekend trips to nearby places,
- car allows you to travel at any time you'd like, you are not dependant on any timetables,
- if the weather is bad, driving a car is better than waiting on the bus stop,
- car allows to transport plenty of luggage without overworking.

First, we describe customers and their vectors of premises and priorities. Next, in subsection 4.2 we discuss methodology of applying triangular norms for computing the decision.

4.1 Consumer's Vectors Description

The analysis revolves around five customers (named A, B, C, D and E). All of them are discussing the same set of attributes regarding a purchase of a car. Selection of vectors of premises and priorities describing customer's preferences was performed by authors. We chose these particular values in order to capture different real-life situations.

Customer A was assigned with following vector of premises:

$$A_{premises} = [0.00, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20] \quad (2)$$

Customer A evaluated all premises as weak ones. First premise is considered as not important at all (is equal to 0). He is not convinced that having a car is necessary. Similarly, vector of priorities for customer A expresses his lack of strong positive opinions regarding one particular car.

$$A_{priorities} = [0.00, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20] \quad (3)$$

A is consequently convinced, that there is no strong motivation for him to buy a car. We expect that consumer his final decision regarding the purchase of a car will be weak.

Customer B was assigned with following vector of premises:

$$B_{premises} = [1.00, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20] \quad (4)$$

B has evaluated first premise as extremely important. He has small babies and having a car would be very helpful to transport them. The rest of premises were evaluated as rather not influencing. Below shown is vector $B_{priorities}$, which gathers customer B's priorities evaluation towards one particular car.

$$B_{priorities} = [1.00, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20] \quad (5)$$

Analogically, first priority was evaluated as extremely important - he believes that this particular car is suitable and would play its role as a help while transporting his family. The rest of priorities still have weak impact on his behavior.

Customer C is described by a following vector of premises:

$$C_{premises} = [0.80, 0.80, 0.80, 0.80, 0.80, 0.80, 0.80, 0.80] \quad (6)$$

All premises were recognized as very important. Analogically, C's vector of priorities $C_{priorities}$ contains high values.

$$C_{priorities} = [0.80, 0.80, 0.80, 0.80, 0.80, 0.80, 0.80, 0.80] \quad (7)$$

C is strongly convinced that buying car is necessary. Moreover, this one particular car he is analyzing meets his expectations. We expect that C's decision should be strong positive.

Customer D has following values attached to premises regarding purchase of a car in general:

$$D_{premises} = [0.80, 0.30, 0.70, 0.00, 0.10, 0.60, 0.90, 0.30] \quad (8)$$

His preferences are varied. There are several premises recognized as important and several recognized as very weak. There is even one attribute (premise number 4: car as a way to manifest one's wealth and social status), which in D's opinion initially does not matter at all. Vector $D_{priorities}$ describes D's priorities towards particular car.

$$D_{priorities} = [0.10, 0.80, 0.20, 0.90, 0.60, 0.10, 0.00, 0.80] \quad (9)$$

This customer's strengths of all priorities are drastically different than strengths of corresponding premises. When D has faced this one particular decision, he drastically reevaluated his opinions. We will observe how different moderating operators would cope with these vectors. Situation captured in $D_{premises}$ and $D_{priorities}$ corresponds to a case, when facing one particular choice we recognize several new important features of given product. In such case, old premises lose their impact on the final decision. This can be caused for example by very persuasive marketing communications, when customer starts to analyze new priorities, which concern one particular car. Successful marketing campaign makes customer D believe that these new projected features of advertised car are even more important than original premises.

Customer E was assigned with following vector of premises:

$$E_{premises} = [0.80, 0.80, 0.80, 0.80, 0.80, 0.80, 0.80, 0.80] \quad (10)$$

Customer E has reviewed his opinions regarding purchase of a car in general and decided that all premises are very strong. But when one particular car was analyzed, he reevaluated the importance of named priorities and it turned out that all attributes are rather insignificant. Vector $E_{priorities}$ presents reevaluated priorities.

$$E_{priorities} = [0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20] \quad (11)$$

Situation described with vectors $E_{premises}$ and $E_{priorities}$ happens when at first we are strongly convinced that some product is a necessity. Next, when we face a particular decision, we are surprised to notice that what has been so appealing has lost its charm and we are rather disappointed.

4.2 Results of Aggregation

In following subsections we discuss results obtained for A, B, C, D and E after applying operators described in section 3. We discuss three different methodologies of moderation and aggregation:

1. various t-norms for premises and priorities moderation and max t-conorm for final decision calculation,
2. min t-norm for premises and priorities moderation and various t-conorms for final decision aggregation,
3. various t-norms for premises and priorities moderation and their dual t-conorms for final decision calculation.

Chosen strategy of applying triangular norms was dictated by literature review and our intuition. Authors picked popular triangular norms and combined them in order to obtain different results.

Moderation with Different t-norms and Aggregating with Max t-conorm. In this subsection we will discuss decisions made for customers A, B, C, D and E basing on maximum t-conorm and vectors of premises and priorities moderated with different t-norms. At first, we would like to show exemplar outputs of moderation. Please note that all calculations were performed according to formulas given in Table 1.

Table 12 contains exemplar results of premises and priorities moderation using minimum t-norm for consumers A and C.

A:	0.0	0.2	0.2	0.2	0.2	0.2	0.2	0.2
C:	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8

(12)

In the same fashion calculated were moderated values for consumers B, D and E. Analogically, we moderated premises with priorities using other t-norms, namely: product, Lukasiewicz t-norm, Nilpotent minimum, Drastic t-norm and Hamacher product. Due to space limitations these calculations are not discussed to greater extent. Please note, that none of results obtained via moderation using t-norms different than minimum would give us values greater than ones received with the minimum (property explicitly stated in formula 1).

Next, we would like to discuss final decisions obtained for consumers A, B, C, D and E after applied maximum t-conorm. Table 13 contains values of decisions aggregated with max for vectors moderated using various t-norms. The top row of table 13 contains information about applied t-norms and t-conorms.

	minimum maximum	product maximum	Lukasiew. maximum	Nilpotent maximum	Drastic maximum	Hamacher maximum
A:	0.20	0.04	0.00	0.00	0.00	0.29
B:	0.90	0.90	0.90	0.90	0.90	0.90
C:	0.80	0.64	0.60	0.80	0.00	0.67
D:	0.30	0.24	0.10	0.30	0.00	0.28
E:	0.20	0.16	0.00	0.00	0.00	0.19

(13)

Results of minimum t-norm and maximum t-conorm (first column) show that only B and C are strongly convinced about the purchase of one particular car. All other customers express weak opinions. Consumer B represents a situation, when gathered is one strong positive information and several weak premises and priorities. Decision computed for consumer C is also strong positive one, what is no surprise, as all evaluations both of premises and priorities are high.

In this first case, consumers A, D and E are not convinced about the purchase of given car. Even though some of them have recognized several strong priorities (or premises) the process of moderation disregarded them and the decision was computed as a weak one.

Second column contains values received by maximum t-conorm aggregation on vector of moderated premises and priorities. Moderation was performed using product t-norm. In the second case, only B and C made strong positive decisions. Through properties of multiplication of numbers not greater than 1 the output of moderation contains mostly low numbers. Therefore, max t-conorm as an aggregation operator is more than necessary so that we can somehow balance low results received through multiplication. Applied product t-norm and maximum t-conorm for E gave us result equal to 0.16. Previously discussed combination of operators also returned equally weak choice. The weakest value of aggregated decision was computed for A. Product t-norm strongly reflected the fact that all premises and priorities for A were weak ones.

Third column of table 13 contains decisions obtained with a combination of Lukasiewicz t-norm and maximum t-conorm. Analogically to the previous case, consumers B and C are the only ones, who are convinced about the purchase of a car. Consumers A and E's decisions are equal to 0 - they are absolutely indifferent towards the decision regarding a particular car. This is first combination of t-norms and t-conorms, which returns such a strict output.

Fourth column in table 13 presents values obtained after applying Nilpotent minimum t-norm and max t-conorm for A, B, C, D, and E. Results of aggregation obtained with Nilpotent minimum t-norm and maximum t-conorm are close to results obtained using Lukasiewicz and maximum norms. B and C are strongly convinced about the purchase of the car, while A, D and E would not buy the car. In this case output for C is more satisfying (is higher) than in the previous case - enhanced was the fact that C's both vectors have all strong positive values.

Fifth column contains the results obtained after applying Drastic t-norm and maximum t-conorm. As we expected, all uncertain cases were eliminated. Positive decision was computed only for consumer B.

Sixth column in table 13 presents the output of moderation and aggregation using Hamacher product t-norm and maximum t-conorm for all customers. Noticeable is that B and C are strongly convinced about the purchase of the car. At the same time A, D and E's decisions are weak positive.

Moderating with Min t-norm and Aggregating with Various t-conorms.

Second part of our experiment was to observe the differences between decisions aggregated using different t-conorms. In this approach in each case premises and priorities were moderated using minimum t-norm. Table 14 compares final decisions obtained using t-conorms as aggregation operators. The top row of table 13 contains information about applied t-norms and t-conorms.

	minimum maximum	minimum prob. sum	minimum bound. sum	minimum Nilpotent	minimum Drastic	minimum Einstein
A:	0.20	0.79	1.00	0.20	1.00	0.89
B:	0.90	0.98	1.00	1.00	1.00	0.99
C:	0.80	1.00	1.00	1.00	1.00	1.00
D:	0.30	0.71	1.00	0.30	1.00	0.81
E:	0.20	0.83	1.00	0.20	1.00	0.92

(14)

Results obtained using min t-norm for moderation and various t-conorms for output calculation are unsatisfactory. Apart from the first and fourth case, when we applied min-max dual operators and minimum/Nilpotent maximum norms, all other results compute strong positive decisions for each customer. Especially undesirable effects obtained were for bounded sum and Drastic t-conorms.

We find second methodology rather unsuitable. Nevertheless, basing on discussed results, we observed how different t-conorms aggregate sequences of the same values. For example, let's discuss aggregation of an 8-element sequence of 0.2s. We observe that two t-conorms, namely maximum and Nilpotent maximum maintain the same result and it is 0.2. All other discussed t-conorms gradually saturate and tend to 1. Norm, which is returning 1 instantly is Drastic t-conorm. Norm, which is tending most slowly to 1 is bounded sum. The higher are the values of aggregated sequence, the faster the result tends to 1. For a sequence of 0.8s, only after three operations the result is saturated. Of course, by definition of triangular norm, the result of aggregation is always bounded by 1.

In consequence, gradual saturation of the result means that certain triangular norms, including bounded sum, probabilistic sum and Einstein sum incorporate following property: the more positive motivational stimuli one recognizes the closer the decision gets to 1. Second corollary is following: one strong positive argument (moderated premise with priority) can be replaced by several weak arguments. Discovering these properties is very important from the point of view of consumer decision making modeling. Presented properties of aggregation operators are highly desirable - economists prove that people tend to simplify cognitive processes. What does it mean in the context of decision making? Saturation of named t-conorms allows us to model a situation, when customer, even though there might be an infinite number of premises, is able to efficiently make

a decision basing on relatively small set of arguments. Presented t-conorms allow to represent nontrivial aspects of the decision making processes, including behavioral biases. We believe that applying them in the neoclassical theory of consumer's choice might be increase its accuracy and would allow us to describe real-life situations more accurately.

Moderating with Different t-norms and Aggregating with Their Dual t-conorms. Last, but not least, we would like to discuss the decisions regarding purchase of a car computed basing on various dual t-norms and t-conorms. Table 15 shows decisions made for consumers A, B, C, D and E. The top row informs us, which t-norm and t-conorm was applied in order to compute particular decision.

	minimum maximum	product prob. sum	Lukasiewicz bounded sum	Nilpotent Nilpotent	Drastic Drastic	Hamacher Einstein
A:	0.20	0.23	0.00	0.00	0.00	0.58
B:	0.90	0.92	0.90	0.90	0.90	0.98
C:	0.80	1.00	1.00	1.00	0.00	1.00
D:	0.30	0.60	0.20	0.20	0.00	0.78
E:	0.20	0.75	0.00	0.00	0.00	0.91

(15)

Obtained results prove again that the choice, of which operators apply for computing the decision has major impact on the output. The differences between particular results are substantial. It is vividly seen in the case of customer C, whose decisions vary from 0 to 1, depending on the applied norm. Variety of properties incorporated in different aggregating operators would allow us to use ones, which would describe decision making phenomena the best.

Decisions obtained for D and E are analogical to ones received with the first methodology. We find outputs computed using operators product/probabilistic sum and Hamacher product/Einstein sum for D and E as too high. These two vectors were constructed to reflect consumers who are undecided (D) or who are disappointed with one particular car described by vector of priorities (E).

Customers B and C were assigned with strong positive decisions in all cases, except from one (Drastic t-norm and t-conorm for consumer C). Comparing the third methodology with the first and the second one, we see that C's decisions were computed before as weaker. For further research left is the topic, which methodology is more suitable.

Finally, we'd like to discuss decisions computed using dual t-norms and t-conorms for consumer A. Third approach computed rather optimistic results. In comparison, for A's decisions, first methodology in all cases computed values non greater than using the third approach. Third approach for the following three dual operators: Lukasiewicz t-norm/bounded sum, Nilpotent minimum/Nilpotent maximum, Drastic t-norm and Darstic t-conorm computed A's decisions as zeros. We find this results as too conservative, since these norms disregard all preferences, which even though are weak, but still they exist. As we mentioned in [5], Kahneman and Tversky proved that people tend to overweight small probabilities and underweight moderate and high probabilities, c.f. [10].

Therefore justified would be the choice of conservative operators for the case of consumer C and rather optimistic operators for A.

5 Conclusions

In the article we discuss how different triangular norms can be applied to model decision making processes based on positive premises only. We support our paper with the case study of five customers who decide about one particular car basing on the same set of attributes. These five cases represent different real-life situations of people, who show varied attitudes towards both a decision regarding purchase of a car in general and different opinions regarding one particular car, about which the decision is made. We apply three methodological approaches of how to calculate the decision using t-norms and t-conorms. First one uses various t-norms for premises and priorities moderation and maximum t-conorm to compute the output. Second approach applies minimum t-norm for vector's moderation and various t-conorms for decision calculation. Third approach uses dual t-norms and t-conorms. First and third methodologies bring us satisfactory results. Second one we find rather not suitable. We noticed that several t-norm/t-conorm combinations would allow us to model more conservative (weak) decisions, while some other are optimistically enhancing the result. An example of a set of operators strengthening the decision is Hamacher product and Einstein sum. Examples of t-norms, which applied to vectors of premises and priorities compute rather weak decisions, for almost each t-conorm, are Lukasiewicz, Drastic and Nilpotent minimum t-norms. Important conclusions were introduced while discussing aggregation of sequences using bounded sum, probabilistic sum and Einstein sum norms. Saturation of decision computed using named operators ideally reflects human's tendency for simplification. Presented triangular norms allow to represent nontrivial aspects of the theory of consumer's choice. Possibility of incorporating behavioral biases into consumer decision making models would allow us to develop a methodology, which would describe real-life phenomena more precisely and more extensively.

Acknowledgment. The research is partially supported by the National Science Center, grant No 2011/01/B/ST6/06478.

References

1. Atanassov, K.T.: Intuitionistic fuzzy sets. *Fuzzy Sets and Systems* 20, 87–96 (1986)
2. Atanassov, K.T.: More on intuitionistic fuzzy sets. *Fuzzy Sets and Systems* 33, 37–45 (1989)
3. Gau, W.L., Buehrer, D.J.: Vague sets. *IEEE Transactions on Systems, Man, and Cybernetics* 23, 610–614 (1993)
4. Homenda, W.: Balanced Fuzzy Sets. *Information Sciences* 176, 2467–2506 (2006)
5. Homenda, W., Jastrzebska, A.: Modeling Consumer's Choice Theory: Using Fuzzy Sets and their Generalizations. In: *Proc. of the 2012 IEEE World Congress on Computational Intelligence (IEEE WCCI 2012), Brisbane (in press, 2012)*

6. Homenda, W., Jastrzebska, A.: Modelling consumer needs. In: Proc. of the 10th International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets, SAP PAN (in press)
7. Klement, E.P., Mesiar, R., Pap, E.: Triangular norms. Kluwer Academic Publishers, Dordrecht (2000)
8. Lewin, K.: Field theory in social science; selected theoretical papers. In: Cartwright, D. (ed.). Harper & Row, USA (1951)
9. Schweizer, B., Sklar, A.: Probabilistic Metric Spaces. North Holland, New York (1983)
10. Wakker, P.P.: Prospect Theory: For Risk and Ambiguity, p. 179. Cambridge University Press, New York (2010)

Neural Network Modeling of a Flexible Manipulator Robot

Rahma Boucetta and Mohamed Naceur Abdelkrim

University of Gabes, Engineering School of Gabes
Omar Ibn Khattab Avenue, Zrig, 6029, Gabes, Tunisia
rboucetta@yahoo.fr, naceur.abdelkrim@enig.rnu.tn

Abstract. This paper presents an artificial neural networks application for a flexible process modeling. A flexible planar single-link manipulator robot is considered. The dynamic behavior of this process is described using Lagrange equations and finite elements method. The artificial neural networks are all variations on the parallel distributed processing (PDP) idea. The architecture of each network is based on very similar building blocks which perform the processing. Therefore, two feed-forward and recurrent neural networks are developed and trained using back-propagation algorithm to identify the dynamics of the flexible process. Simulation results of the system responses are given and discussed in terms of level of error reduction. Finally, a conclusion encloses the paper.

Keywords: Flexible manipulator, dynamic model, finite elements method, Lagrange equations, neural networks.

1 Introduction

Robotic manipulators are generally built using heavy material to maximize stiffness, in an attempt to minimize system vibration and achieve good positional accuracy. As a consequence, such robots are usually heavy with respect to the operating payload. The operation speed of the robot manipulation is limited, so the actuators size is increased boosting energy consumption and increasing the overall cost. Moreover, the robot has a low payload to robot weight ratio. In order to solve these problems, robotic manipulators are designed to be lightweight [13].

Conversely, flexible manipulators exhibit many advantages over their rigid counterparts: they require less material, are lighter in weight, have higher manipulation speed, lower power consumption, require smaller actuators, are more maneuverable and transportable, are safer to operate due to reduced inertia, have enhanced back-drive ability due to elimination of gearing, have less overall cost and higher payload to robot weight ratio [13].

A first wave of interest of neural networks emerged after the introduction of simplified neurons by McCulloch and Pitts in 1943. These neurons were presented as models of biological neurons and as conceptual components for circuits that could perform computational tasks [2, 3, 8].

Most neural networks funding was redirected and researchers left the field. The interest in neural networks re-emerged only after some important theoretical results were attained in the early eighties, and new hardware developments increased the processing capacities.

Artificial neural networks can be most adequately characterized as computational models with particular properties such as the ability to adapt or learn, to generalize, or to cluster or organize data, and which operation is based on parallel processing. The intriguing question is to which extend the neural approach proves to be better suited for certain applications than existing models [1, 3, 5].

A connectionist system consists of a set of interconnected processing elements and is capable of improving its performance based on past experimental information [7]. An artificial neural network is a connectionist system that was originally proposed as a simplified model of the biological nervous system [8]. Neural networks have been shown to provide an efficient means of learning concepts from past experience, abstracting features from uncorrelated data, and generalizing solutions from unforeseen inputs. Other promising advantages of neural networks are their distributed data storage and parallel information flow, which cause them to be extremely robust with respect to malfunctions of individual devices as well as being computationally efficient.

There have been many architectures (i.e. schema consisting of various neurotic characteristics, interconnecting topologies, and learning rules) proposed for neural networks over the last ten years. Simulation experience has revealed that success is problem-dependent. Some networks are more suitable for adaptive control whereas others are more appropriate for pattern recognition, signal filtering, or associative searching. Neural networks that employ the well known back-propagation learning algorithm [9] are capable of approximating any continuous functions with an arbitrary degree of accuracy [9].

This paper is organized as follows: The flexible manipulator system is presented in the first paragraph. The dynamic robot behavior is explained in the next paragraph. The following paragraph deals with applied topologies of artificial neural networks to model the flexible manipulator robot. Simulation results are showing and discussing with each useful architecture.

2 The Flexible Manipulator System

A schematic representation of the single-link flexible manipulator system is shown in figure 1, where a control torque $\tau(t)$ is applied at the hub by an actuator motor with E , I , ρ , L and I_H represent Young's modulus, second moment of area, mass density per unit volume, length, and hub inertia moment respectively [10, 12].

The angular displacement of the link in the X_0OY_0 coordinates is denoted by $\theta(t)$. $w(x,t)$ represents the elastic deflection of the manipulator at a distance x from the hub, measured along the OX axis. X_0OY_0 and XOY represent the stationary and moving frames respectively.

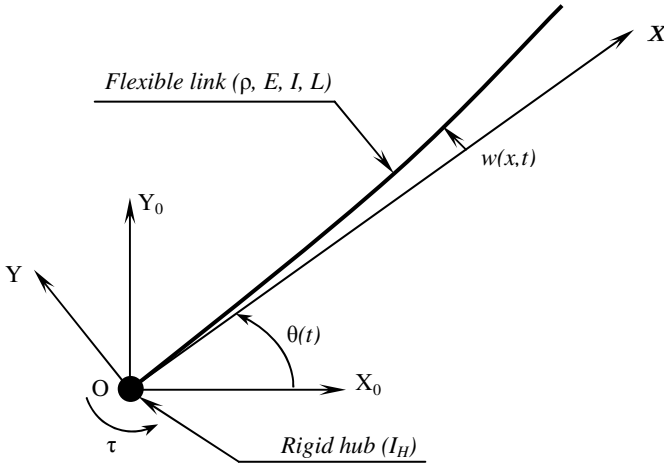


Fig. 1. Flexible manipulator scheme

The height (width) of the link is assumed to be much greater than its depth, thus allowing the manipulator to vibrate dominantly in the horizontal direction (X_0OY_0 plane). To avoid difficulties arising from time varying lengths, the length of the manipulator is assumed to be constant. Moreover, the shear deformation, the rotary inertia and the effect of axial force are ignored. For an angular displacement θ and an elastic deflection w , the total displacement $y(x, t)$ of a point along the manipulator at a distance x from the hub can be described as a function of both the rigid body motion $\theta(t)$ and the elastic deflection $w(x, t)$ [6, 12, 14], i.e.

$$y(x, t) = x\theta(t) + w(x, t) \quad (1)$$

Thus, by allowing the manipulator to be dominantly flexible in the horizontal direction, the elastic deflection of the manipulator can be assumed to be confined to the horizontal plane only.

Kinetic energy of the flexible manipulator, depending of hub rotation and modes rotation in the X_0OY_0 and XOY frames, has the following expression [6, 14]

$$T = \frac{1}{2} I_H \dot{\theta}^2 + \frac{1}{2} (\dot{q} + L\dot{\theta})^T M (\dot{q} + L\dot{\theta}) \quad (2)$$

Potential energy just depending of link flexibility has the following form

$$V = \frac{1}{2} q^T K q \quad (3)$$

After applying Lagrange's equations, the dynamic model can be written as

$$\begin{aligned}(I_H + L^T M L)\ddot{\theta} + L^T M \ddot{q} &= \tau \\ M \ddot{q} + L^T M \ddot{\theta} + K q &= 0\end{aligned}\quad (4)$$

where M , K and L are the mass matrix, the stiffness matrix and the length array respectively, and q is the elastic modes vector.

3 Dynamic Behavior

The dynamic equations can be presented in a state-space form as

$$\begin{aligned}\dot{v} &= Av + Bu \\ y &= Cv + Du\end{aligned}\quad (5)$$

where the state-space matrices are

$$A = \begin{pmatrix} 0 & I \\ -M^{-1}K & 0 \end{pmatrix}, B = \begin{pmatrix} 0 \\ M^{-1} \end{pmatrix}, C = (I \quad 0), D = (0) \quad (6)$$

The state and control vectors are given by

$$\begin{aligned}v^T &= (\theta \quad q_1 \quad q_2 \quad \cdots \quad \dot{\theta} \quad \dot{q}_1 \quad \dot{q}_2 \quad \cdots) \\ u^T &= (\tau \quad 0 \quad \cdots)\end{aligned}\quad (7)$$

In order to simulate the flexible manipulator system, an aluminum flexible link of dimensions $L=0.61\text{m}$ and $S=3\times 10^{-5}\text{m}^2$, with $E=200\times 10^9\text{N/m}^2$, $I=2.5\times 10^{-12}\text{m}^4$, $I_H=4.3\times 10^{-3}\text{kg.m}^2$ and $\rho=7.8\times 10^3\text{kg/m}^3$ is considered. The link is discretized into two elements.

Solving the state-space matrices gives the vector of states v , that is, the hub angle, the elastic modes and their velocities. The derived dynamic model is a nonminimum phase system, not strictly proper, and unstable. Also, the model has zeros very close to the imaginary axis; this deteriorates the time domain performance of the closed-loop system [11, 15].

Generally, linear models of flexible structures used in design of controllers are derived under restrictive assumptions which are often not valid for large motions that occur during slewing maneuvers. Hence, considerable uncertainty in the linear model exists. Another feature characteristic of lightly damped systems is the occurrence of poles ($\pm 159.67j$, $\pm 37j$, 0 , 0) and zeros ($\pm 158j$, $\pm 25j$) very close to the imaginary axis that gives rise to ill-conditioned systems. The state-space matrices arising out of such systems have largely separated singular values, posing considerable computational difficulty in controller design. In the spectral density given by figure 2, the vibration frequencies of the system are obtained as 37rad/s and 160rad/s , i.e. 5.9Hz and 25.46Hz , and the magnitude of frequency response for the two resonance modes are 122dB and 68.8dB .

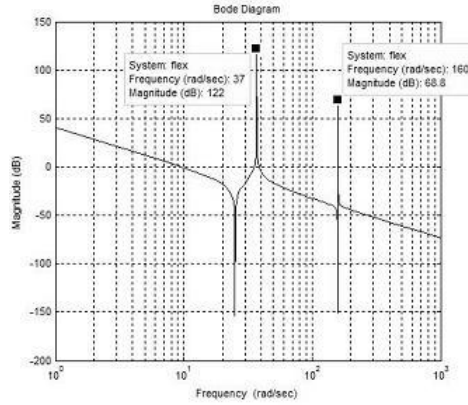


Fig. 2. Open-loop frequency response

4 Neural Modeling

4.1 Introduction

An artificial neural network consists of a pool of simple processing units which communicate by sending signals to each other over a large number of weighted connections [1, 2].

Each unit performs a relatively simple job: receive input from neighbors or external sources and use this to compute an output signal which is propagated to other units. Apart from this processing, a second task is the adjustment of the weights. The system is inherently parallel in the sense that many units can carry out their computations at the same time [4, 7, 8].

In most cases, we assume that each unit provides an additive contribution to the input of the unit with which it is connected. The total input to unit k is simply the weighted sum of the separate outputs from each of the connected units plus a bias or offset term b_k

$$s_k(t) = \sum_j w_{jk}(t)y_j(t) + b_k(t) \quad (8)$$

4.2 Network Topologies

The pattern of connections between the units and the propagation of data can be distinguished into [2, 3]

- Feed-forward networks, where the data flows from input to output units is strictly feed-forward. The data processing can extend over multiple layers of units, but no feedback connections are present, that is, connections extending from outputs of units in the same layer or previous layers.

- Recurrent networks that do contain feedback connections. Contrary to feed-forward networks, the dynamical properties of the network are important. In some cases, the activation values of the units undergo a relaxation process such that the network will evolve to a stable state in which these activations do not change anymore. In other applications, the change of the activation values of the output neurons is significant, such that the dynamical constitutes the output of the network.

4.3 Neural Dynamic Modeling

The parsimonious universal approximation property of neural networks can advantageously be exploited for dynamic modelisation of different processes. Two types of modelisation can be usually distinguished [7, 9]:

- Knowledge models, which the mathematical expression, including small number of adjustable parameters, results from (physical, chemical, etc.) analysis of the process.
- Black Box models, which are determined only from measurements made of the process, without any external knowledge

4.4 Feed-Forward Neural Model of the Flexible Process

The feed-forward neural model is defined by the input $\tau(k)$, the torque applied to the hub motor, its previous values, and the previous values of the output $\theta(k)$, the hub angle of the process. The neural model can be determined, after training, the future output of the process $\hat{\theta}(k)$. The structure of this model is given by the figure 3.

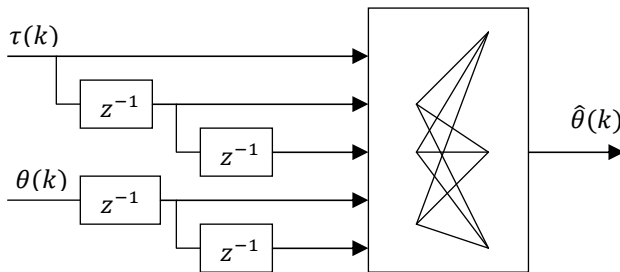


Fig. 3. Feed-forward neural model

The flexible process is excited with a single-switch bang-bang signal of amplitude 0.2Nm input torque, applied at the hub of the manipulator. Figure 4 shows the input signal. A bang-bang torque has a positive (acceleration) and negative (deceleration) period allowing the manipulator to, initially, accelerate, then decelerate and eventually stop at a target location.

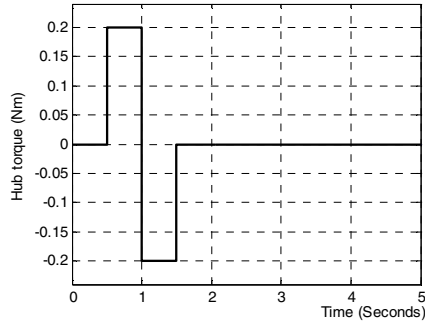


Fig. 4. A bang-bang input torque

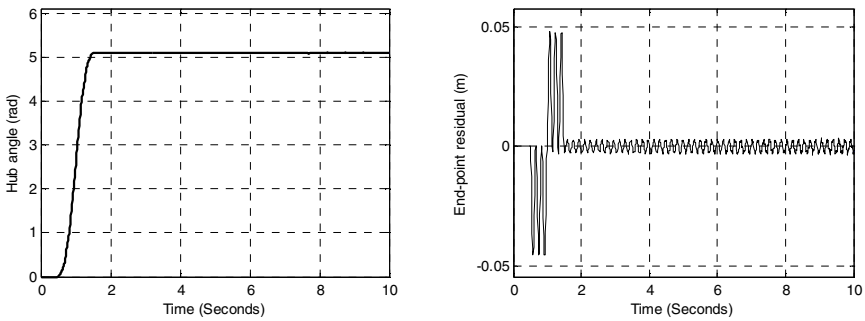


Fig. 5. Hub angle and end-point residual responses

The feed-forward neural network is composed of three layers with a nine neurons hidden layer. It is trained with the back-propagation method. The simulation results of the process responses are given in figure 6. To compare between neural and theoretical models responses, we can notice a good learning of the flexible process dynamics behavior with a hub angle error less than 10^{-3} rad and an end-point residual error less than 1mm.

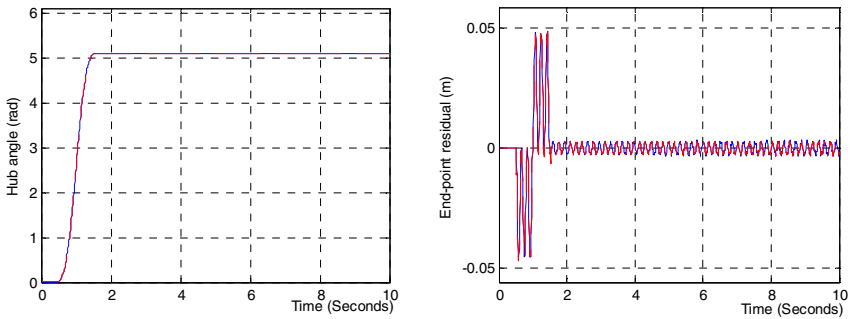


Fig. 6. Hub angle and end-point residual responses with neural and mathematical models

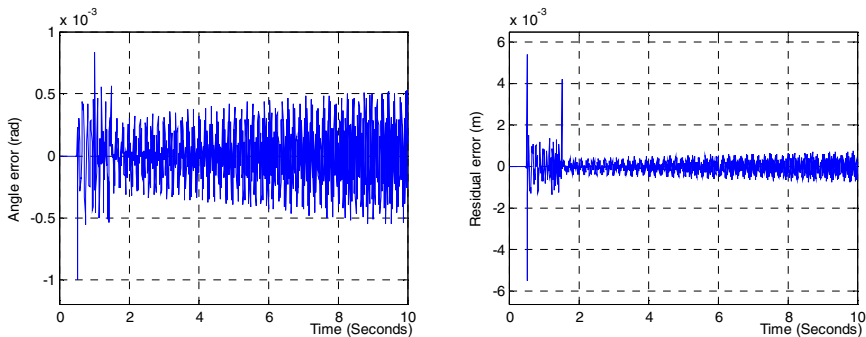


Fig. 7. Hub angle and end-point residual errors

4.5 Recurrent Neural Model for the Flexible Process

The recurrent model is developed in terms of torque input, its previous values, and the previous values of the neural model output. The figure 8 shows the simplified layout of the recurrent model.

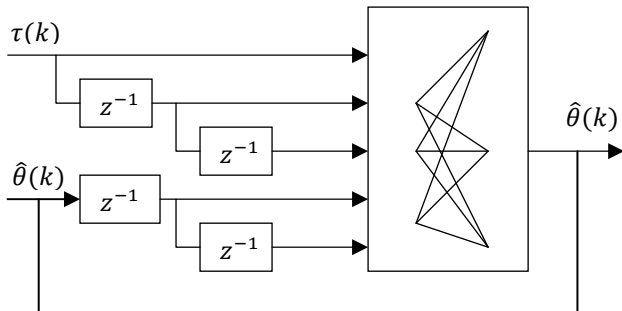


Fig. 8. Recurrent neural network model

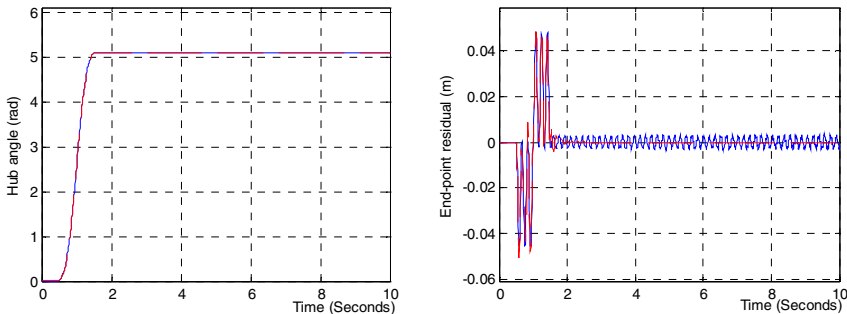


Fig. 9. Hub angle and end-point residual responses

In our case, the recurrent neural model is composed of three layers of neurons; the hidden layer comprised 13 neurons. The learning of the neural network with the back-propagation method can give us the process responses in figures 9 and 10. The error value can be noticed around 0.005rad for the hub angle and 3mm for the end-point residual response.

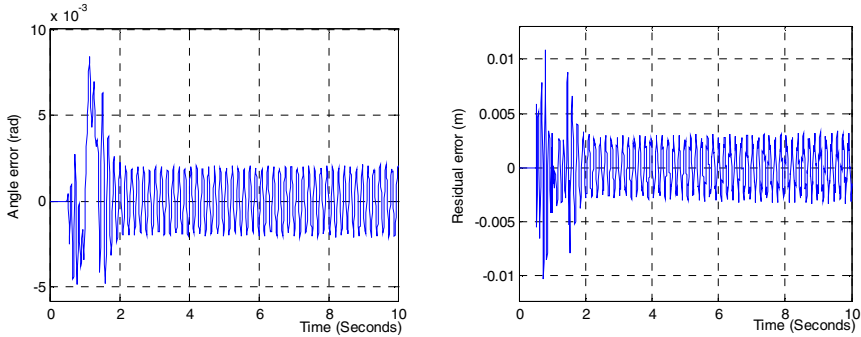


Fig. 10. Hub angle and end-point residual error responses

5 Conclusion

This paper presents a neural modeling of a flexible robot system with feed-forward and recurrent networks methodologies. Dynamic mathematical model of a flexible single-link manipulator robot is derived from Lagrange equations and finite element method. The flexible process is excited with a single-switch bang-bang input torque to consider dynamic behavior and its physical parameters. The learning ability of neural networks allows giving implicit models describing complex dynamic behavior of flexible robots. A feed-forward neural network is first trained with back-propagation algorithm to learn computed dynamic model. Next, a recurrent network is developed to form an independent black box model. A set of simulation results showing behavior of mathematical and neural models is presented. A good learning is revealed in the hub angle and end-point residual responses.

References

1. Abe, S.: Neural Networks and Fuzzy Systems, Theory and Applications. Kluwer Academic Publishers, USA (1997)
2. Hecht-Nielsen, R.: Neurocomputing. Addison-Wesley, Reading (1989)
3. Hertz, A., Krogh, A.S., Palmer, R.G.: Introduction to the Theory of Neural Computation. Addison-Wesley, Redwood City (1991)
4. Narendra, K.W., Parthasarathy, K.: Identification and Control of Dynamical Systems using Neural Networks. IEEE Transactions on Neural Networks, 4-27 (1990)
5. Omidvar, O., Elliott, D.L.: Neural Systems for Control. Academic Press (1997)

6. Usoro, P.B., Nadira, R., Mahil, S.S.: A Finite Element/Lagrange Approach to Modeling Lightweight Flexible Manipulators. *Transactions of the ASME* 108, 198–205 (1986)
7. Simon, H.: *Neural Networks*. Macmillan College Publishing (1994)
8. Zurada, J.: *Introduction to Artificial Neural System*. West Publishing Company (1992)
9. Fu, L.: *Neural Network in Computer Intelligence*. McGraw Hill Book Company (1994)
10. Bickford, W.: *Mechanics of Solids*. Richard D. Irwin, Inc. (1992)
11. Kuo, C.F., Kuo, C.Y.: Modeling and simulation of nonlinear dynamics in a flexible robot arm. *Active Control of Noise and Vibration ASM* 38, 149–156 (1992)
12. Meirovitch, L.: *Elements of Vibration Analysis*, International Student Edition, 4th printing. McGraw-Hill International Book Company (1982)
13. Wang, F.Y., Gao, Y.: *Advanced Studies of Flexible Robotic Manipulators. Modeling, Design, Control and Applications*. Series in Intelligent Control and Intelligent Automation, vol. 4. World Scientific Publishing (2003)
14. Roy Pota, H.: Finite-element/Lagrange Modeling and Control of a Flexible Robot Arm. In: 11th IFAC World Congress, Tallinn, Estonia, USSR, vol. 9, pp. 239–243 (1990)
15. Hashemi, S.M., Borneman, S.R., Alighanbari, H.: Vibration of Cracked Composite Beams: A Dynamic Finite Element. *International Review of Aerospace Engineering (IREASE)* 1(1), 110–121 (2008)
16. Mansour, T., Konno, A., Uchiyama, M.: MPID Control Tuning for a Flexible Manipulator Using a Neural Network. *Journal of Robotics and Mechatronics* 22(1), 82–90 (2010)
17. Abe, A.: Trajectory planning for flexible Cartesian robot manipulator by using artificial neural network: numerical simulation and experimental verification. *Robotica* 29(05), 797–804 (2011)
18. Mahamood, R.M., Pedro, J.O.: Hybrid PD-PID with Iterative Learning Control for Two-Link Flexible Manipulator. In: *Proceedings of the World Congress on Engineering and Computer Science*, San Francisco, USA, vol. II (October 2011)

P Systems for Traffic Flow Simulation

Jiří Dvorský^{2,1}, Zbyněk Janoška¹, and Lukáš Vojáček²

¹ Department of Geoinformatics, Palacký University,
Třída Svobody 26, 771 46, Olomouc, Czech Republic
`jiri.dvorsky@upol.cz`, `zbynek.janoska@cdv.cz`

² Department of Computer Science,
VŠB – Technical University of Ostrava, 17. Listopadu 15,
708 33 Ostrava, Czech Republic
`lukas.vojacek@vsb.cz`

Abstract. Membrane computing is an emergent branch of natural computing, taking inspiration from the structure and functioning of a living cell. P systems, computing devices of this paradigm, are parallel, distributed and non-deterministic computing models which aim to capture processes taking place in a living cell and represent them as a computation. In last decade, a great variety of extensions of model, introduced by Paun in 1998, were presented. In this paper, we focus on modelling the traffic flow by the means of P systems. P systems enable mezosopic representation of traffic flow with individual modelling of each cars behaviour. Theoretical model is presented together with an XML scheme to store the output of the model.

Keywords: Membrane computing, P systems, traffic flow, XML.

1 Introduction

Membrane computing represents new and rapidly growing branch of natural computing, which starts from observation that the processes taking place in a living cell can be understood as a computation. Membrane computing and its computational device – *P system* – were introduced by Păun [9] and gained a lot of interest in last decade. P systems start from observation, that membrane plays a fundamental role in the functioning of a living cell. Membranes act as three-dimensional compartments which delimit various regions of a living cell. They are essentially involved in a number of reactions taking place inside cell and moreover act as selective channels of communication between different compartments of a cell [3].

P systems take inspiration from cell on two levels – the structure and the functioning. Structure of cell is represented by its membranes and functioning is governed by biochemical reactions. Every P system therefore has three main elements: a *membrane structure*, where *object* evolve according to given *evolution rules* [11]. Some authors add fourth basic element of membrane systems – *communication* [3,10]. Communication is always encoded in rules (they are called *communication rules* instead of *evolution rules*) and will be dealt with later in

the text. From the point of view of transportation modelling, communication (e.g. topology) is essential feature.

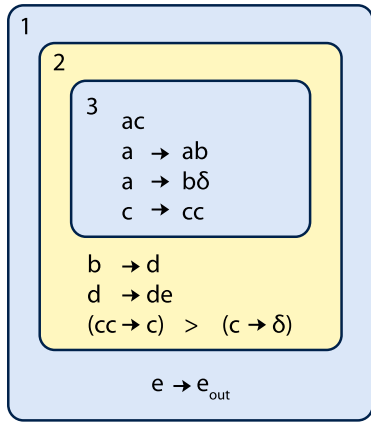


Fig. 1. Graphical representation of P system, [17]

Simple example of P system is depicted in Fig. 1. Membrane structure is hierarchically arranged set of membranes, contained in a distinguished outer membrane, called *skin* membrane. System is surrounded by the *environment*, which may collect objects leaving the system, or in some variants of P systems, the environments can actively support system with objects [24]. Membranes delimit *regions*, with which they are in one-to-one relation. Therefore the terms *membrane* and *region* are mostly interchangeable. Each membrane is identified by its *label*, which can be with membranes in one-to-many relation. The position of inner membranes does not matter; we assume, that in membrane *there is no ordering, everything is close to everything else* [11].

Second basic element of P systems are *objects*. By objects in biological sense are meant chemicals, ions, molecules etc. Those substances are present in a cell in enormous amounts, but the ordering again does not matter. What matters is the concentration, the population, the number of copies of each molecule [11]. Abstracting from biological reality, we represent each substance by a symbol from given alphabet and since the multiplicity matters, instead of objects we use *multisets* of objects. Common notation of multisets in P systems is following: if, for example, objects a, b, c are present in 7, 2 and 5 copies, they will be represented by multiset $a^7b^2c^5$.

In basic variant of P systems, multisets of objects are considered to be floating in inner regions of membrane systems. They evolve by the means of *evolution rules*, which are localised with the regions of the membrane structure. There are three main types of rules [11]: (1) multiset-rewriting rules, (2) communication rules and (3) rules for handling membranes. In this section only first type of rules will be described.

Multiset-rewriting rules take form $u \rightarrow v$, where u and v are multisets of objects. For example, rule $ab \rightarrow cd^2$ says, that one copy of a and one copy of b are consumed and one copy of c and two copies of d are produced. A number of possible extensions of rules exists.

Two crucial features of P systems have to be mentioned at this point. As mentioned earlier, in membranes everything is close to everything else. Therefore, if one instance of an object can be processed by two or more rules, the rule to be applied is chosen *non-deterministically*. All rules have the same probability to be chosen. The rules also have to be used *in maximally parallel manner*.

More specifically, the objects are assigned to rules, non-deterministically choosing the objects and the rules, until no further assignment is possible. An evolution step in a given region of membrane system consists of finding the maximal applicable multiset of rules, removing from region all objects specified in the left hand of the chosen rules and producing the objects on the right hand side of the rules.

After giving short introduction to basic notions of P systems, let us continue with more detailed survey on traffic flow modelling and representation of transportation system by the means of P systems. In Sect. 2, a definition of basic model – transitional P system, is given. Sect. 3 presents possible extensions of such model, which are incorporated into a model, described in Sect. 5. Sect. 4 gives a short introduction into the traffic flow modelling. Finally, an XML scheme designed to store the configuration of developed system is presented in Sect. 6. We will conclude with some final remarks in Sect. 7.

2 Transition P System

P systems based on application of *multiset-rewriting rules* are called *transition P system*. Formally, transition P system is a construct of the form:

$$\Pi = (O, C, \mu, w_1, w_2, \dots, w_m, R_1, R_2, \dots, R_m, i_o), \quad (1)$$

where:

- O is the finite and non-empty alphabet of objects,
- $C \subset O$ is the set of catalysts,
- μ is a membrane structure, consisting of m membranes, labeled $1, 2, \dots, m$; one says, that the membrane structure, and hence the system, is of degree m ,
- w_1, w_2, \dots, w_m are strings over O representing multisets of objects present in regions $1, 2, \dots, m$ of membrane structure,
- R_1, R_2, \dots, R_m is finite set of evolution rules associated with regions $1, 2, \dots, m$ of membrane structure,
- i_o is either one of the labels $1, 2, \dots, m$ and then the respective region is the *output region* of the system, or it is 0 and then the result of the computation is collected in the environment of the system.

A sequence of transitions of P system constitutes a *computation*. A computation is successful if it halts, it reaches a configuration where no rule can be applied to the existing objects, and output region i_o still exists [11].

The rules are of form $u \rightarrow v$, where $u \in O$ and $v \in (O \times Tar)$, where $Tar = \{here, in, out\}$. Target indications Tar extend transition P system in following way: rule $ab \rightarrow c_{here}d_{in}e_{out}$ consumes one instance of each a and b and produces one copy of c in current membrane, one copy of d in a child of current membrane and one copy of e in the parent of current membrane. If current membrane is skin membrane, object e is send to environment of the system. If current membrane does not have a child, rule can not be applied.

Another extension comes from the existence of *catalysts*. Catalysts are objects, which participate in a chemical reaction, but are not consumed or produced by it. They just enable the application of rule. Rule with catalysts takes following form: $ac \rightarrow bc$, with object c being the catalyst.

3 Possible Extensions of P Systems

In this section we will mention some elementary extensions of P system, which however constitute only a fracture of possibilities. We refer reader to The P systems Webpage [16] for complete list of publications and further information. Already in the text, three types of rules were mentioned. Evolution rules were briefly covered in previous sections.

Communication rules were introduced in [10]. Basic idea of communicating P systems is, that computation is achieved only by transporting object between membranes. Direct inspiration from biology are *symport* and *antiport*. When two chemical pass through membrane only together, in the same direction, the process is called *symport*. When the two chemicals pass only with help of each other, but in opposite directions, the process is called *antiport* [10]. Symport rules take a form (ab, in) or (ab, out) , and antiport rules take a form $(a, out; b, in)$, where a, b are object from alphabet of all possible objects. Meaning of rules is following: for symport rule (ab, in) or (ab, out) , if objects a, b are present in current membrane, they are sent together into child (or parent, in second case) of the membrane. For antiport rule $(a, out; b, in)$, if a is present in current membrane and b is present in the parent of a membrane, than a exits current membrane and b enters it. Universality of P systems with symport and antiport have been proven [10] and simplified version of communication, *conditional uniport* have been studied [14]. Comprehensive review of communication strategies in P systems can be found in [15].

Third type of rules are *rules for handling membranes*. Dissolution of membranes has already been mentioned, but other ways to obtain dynamical membrane structure, evolving during the course of computation, have been presented. Most simple of those is assigning *electrical polarization* $+, -, 0$ to each membrane. Polarization replaces target indicators *in, out, here*. Polarization of objects is introduced by the rules and polarized objects can enter only membranes with opposite polarization. For example, $ab \rightarrow c^+d^-$ means, that one instance of c enters inner membrane with negative polarization and one instance of d enters inner positive membrane. Rules can also be used to change the polarization of membranes during the computation.

Two issues seem essential, when P systems are used to simulate real-world phenomena rather than for computation. Non-determinism is first of the issues. Some chemical reactions are more likely to occur than others. First attempt to solve this is by assigning priorities to rules. Firstly, set of rules with the highest priority is chosen and according to the principle of maximal parallelism, all rules which can be applied, are applied. Then, the rules with second highest priority are selected and the procedure repeats, etc.

Second approach is to assign probabilities to all rules. Probability can be introduced to P system on different levels [8], but here we will mention only probability on the level of rule selection. Different approaches have been proposed [2,6,12]. Basic idea is to associate each rule with a constant k , so the rule takes a form $u \xrightarrow{k} v$, where u, v are multisets of objects and k can be interpreted either as a probability, or as a “stoichiometric coefficient”, using which the true probability is calculated.

Last extension, which we will mention at this point, is representation of time of P systems. In real world, every biochemical reaction takes some time. Representation of time in P systems is similar to representation of probability. A constant t is assigned to each rule, so the rule takes the form $u \xrightarrow{t} v$, where u, v are multisets of objects and t is number of time units, which must pass to complete the application of the rule [5]. In the first time step, multiset u is consumed and removed from the current membrane. After $t - 1$ more time steps, multiset v is introduced into the system. Time can also be introduced into P systems as a lifetime of objects or even membranes [1].

For the sake of brevity, we will not discuss more extensions of P systems, although many possibilities were explored within this framework. P system based model of traffic flow will take advantage of three main features mentioned in this section – membrane polarization, probability and time-dependence.

4 Traffic Flow Simulation

Hoogendoorn and Bovy [7] distinguish three main levels of modelling of transportation systems:

- **microscopic simulation model** describes both the space-time behaviour of the systems’ entities (i.e. vehicles and drivers) as well as their interactions at a high level of detail (individually). Sometimes submicroscopic models are mentioned separately.
- **mezoscopic model** does not distinguish nor trace individual vehicles, but specifies the behaviour of individuals, for instance in probabilistic terms. To this end, traffic is represented by (small) groups of traffic entities, the activities and interactions of which are described at a low detail level.
- **macroscopic flow model** describe traffic at a high level of aggregation as a flow without distinguishing its constituent parts. For instance, the traffic stream is represented in an aggregate manner using characteristics as flow-rate, density, and velocity. Individual vehicle manoeuvres, such as a lane-change, are usually not explicitly represented [7].

While microscopic models describe accurately behaviour of small-scaled systems such as lanes or intersections, their use to simulate spatially extensive areas, such are cities or agglomerations, are limited. Macroscopic models, on the other hand, are capable of simulation on low-scale level, but lack detailed description of individuals. Mezoscopic models are a reasonable compromise between detail and

robustness, which is desirable for certain applications, such are public transport modelling [13].

In this study we propose a simulation model for traffic flow, aiming to describe accurately vehicle behaviour at the intersections, but ignoring detailed description of cars behaviour at road segments between intersections. Such model should be individual based, representing vehicles as agents instead of populations, but at the same time should be robust enough to enable simulation of large-scaled road networks. Such model could be used for example for optimalization of traffic lights at signalized intersections.

5 P System Based Model of Traffic Flow

In this section we describe P system model of traffic flow. This model aims to capture detailed behaviour of cars at the intersection, while ignoring behaviour at road segments.

5.1 Definition of a System

Within P systems paradigm, road network can be described as a graph with membranes at the nodes. This representation is similar to the one of communication P systems [15]. There are three types of membranes, based on their function. *White holes* act as generators of cars. If the systems represents, say, a city, than white holes will be the roads, bringing cars to the city. Opposite reactions take place in *black holes*, where the cars leave the system. White and black holes does not necessarily have to be only at the periphery of the system – also residential, industrial or business areas within the city can act as white or black holes, and they can therefore be placed arbitrarily in the graph. The third type of nodes is an *intersection*. At intersection, cars are distributed to other parts of the network.

Within this paradigm, cars are represented as objects. For the sake of brevity, we will consider only one type of object – an arbitrary vehicle. In praxis, however, different types of vehicles (cars, buses, motorcycles etc.) can be represented by different objects.

The behaviour of objects is described by a set of rules. In classical P systems, rules enable objects to be created, destroyed or qualitatively changed. In the proposed system, each of these is encapsulated in one type of membranes. In white holes, objects are created, in black holes, they are removed from the system, and at the intersections, they are changed (meaning they can change their state from moving to stopping or vice versa). At the intersections, objects are communicated to other membranes in the system.

Clearly, the behaviour of cars is not strictly deterministic, therefore stochasticity must be introduced at some point. There are two ways to represent the behaviour of a car in the network.

- When an object is introduced to the system at white hole A , a target destination B is chosen from set of available black holes. The shortest path

between nodes A and B is calculated and vector of intersection, which must be passed by an object is assigned to it. For each white hole, a set of probabilities is assigned to all available black holes. Therefore, some paths are preferred by the cars then others.

- For each intersection, a set of probabilities is assigned to all intersecting roads. When a car is present at the intersection, a road is chosen and car is send to next node of a graph. Each road is associated with a certain probability of being chosen, therefore some links are more preferred by cars than others.

First approach is preferable in case, where the preferences of cars in the network are well known. If we lack the knowledge about drivers preferences and know only vehicle intensities at the roads, second approach seems more suitable.

Also, travel time have to be taken into account. Time necessary to travel from one node of a graph to another is represented by a cost of the link. Graph is oriented, therefore travel time between two nodes can differ in both directions.

Last issue is associated with representation of traffic lights. At intersection, only cars approaching from a certain direction are allowed to pass. Within P systems framework, membrane polarizations are suitable tool for selective allowance of objects into a membrane. The polarization of each intersections represents red/green lights and periodically changes, therefore simulating real world situation.

5.2 Formal Definition of a Model

Traffic P system is a construct:

$$Traffic\Pi = (O, \mu, syn, (s_{(i,j,t)}))_{(i,j) \in syn; t \in \mathbb{N}}, R), \quad (2)$$

where:

- $O = \{veh, veh_s\}$ is an alphabet of objects, veh representing moving vehicle and veh_s representing stopping vehicle,
- μ is a membrane structure, consisting of m membranes, labeled WH, BK, INT ; representing white holes, black holes and intersections,
- $syn \subseteq \{(i, j, t) | i, j \in \{0, 1, 2, \dots, m\}, i \neq j, t \in \mathbb{N}\}$ is a subset of synapses – links between the membranes, where i and j are membranes from the set μ and t is time constant associated with each synapse.
- R is a set of rules associated with membranes from μ . The rules differ for membranes labeled as white holes, black holes and intersections.

- White holes – vehicles veh are generated and sent to one of neighboring membranes.

$$[]_{WH} \xrightarrow{t_i} [veh^{WH}]_k,$$

where $\{k, WH, t_i\} \in syn$ and application of a rule takes t_i time steps. A label of white hole will be associated with a vehicle veh when approaching intersection k , which will be used to decide, whether a vehicle should stop or pass the intersection.

- Black holes – vehicles *veh* are removed from the system.

$$[veh]_{BH} \xrightarrow{t=1} [],$$

application of such rule takes 1 time unit.

- Intersection – several rules are associated with intersections. Here we describe a situation, where car makes a decision about its target destination at each intersection.

Rule:

$$[veh^k]_m \xrightarrow{t_i, p_i} [veh^m]_n$$

describes a situation, where car coming from membrane k approaches an intersection m with polarization k . Since the polarization and label of *veh* correspond, a *veh* can pass the intersection and proceed to membrane n , with probability p_i . *veh* will be assigned with label of passed membrane m . Application of rule will take t_i time units.

Rule:

$$[veh^k]_m \xrightarrow{t=1, p=1} [veh_s]_m$$

describes a situation, where moving car *veh* coming from membrane k approaches an intersection m with polarization l . Since the polarization and label of *veh* do not agree, the vehicle *veh* must stop and therefore changes to *veh_s*, with probability 1 and taking 1 time step.

Rule:

$$[veh_s^k]_m \xrightarrow{t_i, p_i} [veh]_n$$

describes a situation, where stopping car *veh_s* from membrane k is in membrane m with polarization k ; it can pass the intersection and continue to membrane n . This is enabled by changing polarizations of membranes.

Global clock for the system are assumed. The polarization of membranes changes according to a given schedule. All other parameters of the system - creation rate at white holes, probabilities and time constants associated with the synapses can also be time-dependent.

6 XML Structure for Proposed Model

We used XML language for creating topology, creating rules and setting initialization values. The reason why we chose this way is simple, because we can simply and clearly describe all necessary attributes. A code sample is shown in Listing 1.1, where we divide file to 3 parts.

1. First part stores the topology of P-system. Connections between membranes are defined, I/O membranes are chosen etc.
2. Second part describes the initial configuration of membranes.

3. The third part contains the rules for membranes. It consists of left side, right side and additional information about conditions of use, delays, etc. Left side contains items needed for rule to be applied, right side describes the result of rule application.

Listing 1.1. Example XML file

```

<?xml version="1.0" encoding="utf-8"?>
<Program Version="1.0">
  <Topology>
    <Membrane ID="1">
      <Type Membrane="White"/>
      <LinkTo ID="3" ValuePath="5"/>
    </Membrane>
    <Membrane ID="4">
      <Type Membrane="Black"/>
      <LinkTo ID="3" ValuePath="5"/>
    </Membrane>
    <Membrane ID="3">
      <Type Membrane="Green"/>
      <PolarizationComing IDs="1,4" Time="10"/>
      <PolarizationComing IDs="2,5" Time="5"/>
      <LinkTo ID="2" ValuePath="10"/>
      <LinkTo ID="4" ValuePath="5"/>
      <LinkTo ID="5" ValuePath="5"/>
    </Membrane>
  </Topology>
  <Initialization>
    <Membrane ID="1">
      <Item NameID="a" Value="0"/>
      <Item NameID="b" Value="0"/>
    </Membrane>
  </Initialization>
  <Rules>
    <Selected>
      <Membrane ID="3">
        <LeftSide>
          <Item NameID="a" Value="1"/>
        </LeftSide>
        <RightSide>
          <Item NameID="a" Value="1"/>
        </RightSide>
        <Additional>
          <Probability Value="0.5"/>
          <DuelID Value="4"/>
          <Polarization Value="true"/>
        </Additional>
      </Membrane>
    </Selected>
    <Default>

```

```

<Membrane Typ="White">
  <LeftSide>
    <Item NameID="a" Value="1" />
  </LeftSide>
  <RightSide>
    <Item NameID="a" Value="1" />
  </RightSide>
</Membrane>
</Default>
</Rules>
</Program>

```

7 Concluding Remarks and Future Work

A P system for traffic flow simulation was defined and an XML scheme designed to store the configuration of such model was presented. A basic variant of P system was described with additional features – stochasticity, time-dependence, selective membrane permeability – described in more detail.

Presented model was designed to simulate the behaviour of vehicles at the intersections and enables elegant simulation of cars behaviour at the intersections, while ignoring detailed description of behaviour at road segments. This robustness is desirable i.e. for traffic light optimization.

On the webpages of the authors is possible to find the application, which we develop and use for experiments. (<http://wh.cs.vsb.cz/voj189>)

In the future, the presented model will be further developed to be able to describe the traffic system in more realistic manner – cars delays, complex intersections and destination preferences are of primal interest.

Acknowledgement. This work was supported by SGS, VŠB – Technical University of Ostrava, Czech Republic, under the grant No. SP2012/151 Large graph analysis and processing. This work was also supported by SoftComp project (CZ.1.07/2.3.00/20.0072). SoftComp project is co-nanced by ESF and Czech state budget.

References

1. Aman, B., Ciobanu, G.: Adding Lifetime to Objects and Membranes in P Systems. *International Journal of Computers Communications and Control* 5(3), 268–279 (2010)
2. Bernardini, F., Manca, V.: Dynamical aspects of P systems. *BioSystems* 70(2), 85–93 (2003)
3. Bernardini, F., Gheorghe, M., Krasnogor, N., Muniyandi, R.C., Perez Jimenez, M.J., Romero-Campero, F.-J.: On P Systems as a Modelling Tool for Biological Systems. In: *Pre-Proc. of the Sixth Workshop on Membrane Computing*, Vienna, Austria, pp. 114–133 (2005)

4. Cavaliere, M.: Evolution-Communication P Systems. In: Proceeding WMC-CdeA 2002 Revised Papers from the International Workshop on Membrane Computing, pp. 134–145 (2003)
5. Cavaliere, M., Sburlan, D.: Time-independent P systems. In: International Workshop on Membrane Computing, WMC5, Milano, Italy, pp. 239–258 (2005)
6. Cazzaniga, P., Pescini, D., Romero-Campero, F.-J., Besozzi, D., Mauri, M.: Stochastic Approaches in P Systems for Simulating Biological Systems. In: Fourth Brainstorming Week on Membrane Computing, Seville, Spain, pp. 145–164 (2006)
7. Hoogendoorn, S.P., Bovy, P.H.L.: State-of-the-art of Vehicular Traffic Flow Modelling. Delft University of Technology, Delft, The Netherlands, pp. 283–303 (2001)
8. Obtulowicz, A., Păun, G.: (In search of) Probabilistic P systems. *Biosystems* 70(2), 107–121 (2003)
9. Păun, G.: Computing with Membranes. Technical report, Turku Center for Computer Science-TUCS. Turku, Finland (1998)
10. Păun, A., Păun, G.: The Power of Communication: P Systems with Symport/Antiport. *New Generation Computation* 20(3), 295–306 (2002)
11. Păun, G.: Introduction to Membrane Computing. In: First Brainstorming Workshop on Uncertainty in Membrane Computing, Palma de Mallorca, Spain, pp. 1–42 (2004)
12. Pescini, D., Besozzi, D., Mauri, G., Zandron, C.: Dynamical probabilistic P systems. *International Journal of Foundations of Computer Science* 17(1), 440–447 (2006)
13. Peeta, S., Ziliaskopoulos, A.K.: Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics* 1, 1233–1265 (2001)
14. Verlan, S., Bernardini, F., Gheorghe, M., Margenstern, M.: Computational Completeness of Tissue P Systems with Conditional Uniport. In: Hooeboom, H.J., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2006. LNCS, vol. 4361, pp. 521–535. Springer, Heidelberg (2006)
15. Verlan, S., Bernardini, F., Gheorghe, M., Margenstern, M.: Generalized communicating P systems. *Theoretical Computer Science* 404(1-2), 170–184 (2008)
16. The P Systems Web Page, <http://ppage.psystems.eu/> (last revision: March 18, 2012)
17. P system - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/P_system (last revision: March 18, 2012)

Using Extended Raster File for Real Time Traffic Information Mining

Michal Radecký, Jan Martinovič, Dušan Fedorčák, Radek Tomis, and Ivo Vondrák

VŠB – Technical University of Ostrava, 17. Listopadu 15,

IT4Innovations,

708 33 Ostrava, Czech Republic

{michal.radecky,jan.martinovic,ivo.vondrak}@vsb.cz,

{dusan.fedorcak,radektomis}@gmail.com

Abstract. Gauging and analyzing the ever-growing strain on today's roadways is an issue that needs to be addressed with urgency. The rate at which technology is developing and the amount of vehicles now equipped with GPS systems are factors that ensure the potential for creating statistics to gauge this strain. When processing data gathered from monitored vehicles, it is necessary to implement procedures that identify specific roadways upon which a given vehicle's movement is recorded. With respects to the volume of this data, a method for indexing these data files becomes a critical issue. Within the following text, we will present a process of collecting and processing data in the FLOREON+ Traffic system and spatial indexing using a raster index that processes queries at a much greater speed than standard indexing.

1 Introduction

The problem with analyzing the increasing strain on today's roadways lies within the fact that monitoring is limited to major intersections and trouble spots, where modern camera systems have become the norm. Publicizing information gathered from these areas leads to faster solutions in crisis situations and aids in the overall elimination of such crises. A great disadvantage in this approach, however, is the fact that a crisis situation needs to be identified in advance in order for it to be resolved in time. A system based on collecting and processing data derived from a vehicle in real time may contribute to the exploitation of this disadvantage - the ability to monitor only vehicles where this equipment is installed. Currently, the amount of vehicles thus equipped is too large for a system of this type to be of any significance or to even be successfully implemented.

Floreon⁺ is a science-research project aimed at building a prototype system for modeling, simulating and monitoring situations caused by unfavorable, natural phenomena using modern computer and Internet technology. The main feature of this project is its focus in the area of crisis management and support. The project has gradually expanded to cover relevant topical fields, such as flash flooding, and other factors pertinent to crisis management and general information services.

Recently, traffic problems have become a recurring issue to be dealt with on a daily basis in various forms and at several levels. Aside from the everyday traffic situations

perceived by drivers, crisis situations, management and solutions must be taken into account. This is precisely why the field of traffic information and management are intricate parts of the Floreon⁺ project. Knowledge of traffic situations and the ability to utilize sophisticated apparatus to effectively process this information are significant factors for an entire line of activities from the perspective of crisis management (e.g. which traffic lanes are clear for emergency unit passage, which areas are flood hazards, etc.). Additional benefits would obviously include the acquisition and processing of everyday traffic situations and activities (intelligent navigation systems for automobiles in metro areas, analyzing high-traffic zones, detecting problems on roadways, etc.)

The Floreon⁺ Traffic system makes calculations (in real time or for potential situations) by utilizing a complete line of information derived from various sources. Data acquired in this manner is then analyzed and adjusted to fit a format that is then usable in the aggregation of this data with connection to time and space. Currently, the main information source remains the vehicle itself. Thanks to our cooperation with companies involved in monitoring car parks, we are able to acquire surface data and individual car speeds in real time. Individual data acquired in this way is anonymous enough to fulfill its purpose. Once received, this information is combined with other data relevant to a given traffic area at a given time. The system actually works with data to determine an automobiles' speed and driving style.

1.1 Navigation Systems

Nowadays, the modern technology, especially information technology, is one of the main parts of automotive industry. There are many new features which bring new driving experiences and encourage driver's skills, e.g. adaptive cruise control, lane assistant or traffic signs reader. However, the on-board navigation is still most common tool for daily personal transportation.

For today's drivers, it is very useful and required to know the route from the one place to another. Unfortunately, the simple knowledge of static route itinerary is not enough. There are many factors which affect the travel time and efficiency of the selected road. These factors are changed during the time and they are also depending on time. So, it is necessary to integrate dynamic models of traffic within the routing approaches to offer real-life navigation functionality. These dynamic models can cover real behaviors of traffic on macro level based on time, knowledge of statistical data related to infrastructure elements, knowledge of real traffic occupancy and utilization, etc. Based on these, it is able to develop much more sophisticated routing algorithms that bring whole new perspective on the on-board navigation systems. But of course, on-board navigation is not only one 'consumer' of such approaches and their results. Also the traffic management or logistics business are covetous users of that.

The crucial layer of these approaches is data layer. The ability to collect relevant data and its preprocessing are the very important parts. It is hard to develop dynamic models without suitable data related to the time and location. The information on streets capacity, its utilization, traffic restrictions and incidents, traffic jams, weather, etc. is necessary to have as a background for dynamic model development and operation. Some of these data (GPS positions with speed value) are possible to collect based on cooperation with car fleet providers, however these data are often geographically and quantitative

limited. The companies that develop and offer navigation solutions are able to solve these problems. For example, the TomTom company offers the IQ Routes technology¹, that deals with standard on-board navigation devices and their users which participate on global traffic state knowledge. This solution puts, by online or offline ways, the driving experience of millions of TomTom users into central knowledge base and calculating routes based on current day time, actual speeds driven on roads compared to speed limits.

The future of navigation and other related tools and services is covered by integration of all accessible data sources and its utilization within complex traffic model. On this background, there is able to offer sophisticated

2 Floreon⁺ Traffic

We are currently receiving information on approx. 2,200 vehicles of different types (passenger cars, cargo trucks, lorries, etc.). This data is aggregated into a simple, speed fast profile in real time around the entire Czech Republic. (Coverage is only dependent upon the accessibility of vector data. Acquired data is currently limited to identifying major roadways because an application that covers all streets would not be relevant enough with respect to the amount of moving vehicles and the vastness of this network.) In the near future, we plan to carry out much more detailed analyses of this data in a way that allows us to randomly expand the grid structure of intersections. This expansion will aid in creating a necessary base of information for evaluating operations and applications of graph algorithms.

Currently, the system is receiving anonymous records from approx. 2,200 vehicles. During rush hour, that equates to about 13 records received/second. Taking into account the algorithm being applied, unprecise GPS readings and, specifically, relevant vehicle data (unignited engines, etc.) the success rate for the placement of given information on a specific location is around 30%.

This information is generally highly relevant to the value of real time situations on roadways. Nonetheless, due to the need for high quality and a large quantity of acquired data, this information must be supported with data derived from alternate sources. The most significant data acquired is, therefore, that which is acquired directly from a given infrastructure. Such data may be derived from toll stations, telematic sensors, camera systems, etc. Within the scope of their given locations, these sources may influence aggregated data derived from a specific roadway. The advantage provided by these data sources is the high level of precise information they provide (amount of vehicles passed, etc.). On the other hand, the disadvantages of using these tools as data sources are their predominately localized scope and their high investment and operational costs. However, an appropriate combination of data acquired on-line and data acquired directly from infrastructures allows for the creation of a quality image of the overall traffic situation.

An equally important data source would be that of an agenda system that provides information on traffic accidents, road closures, weather, etc. The potential of this

¹ TomTom: <http://www.tomtom.com>

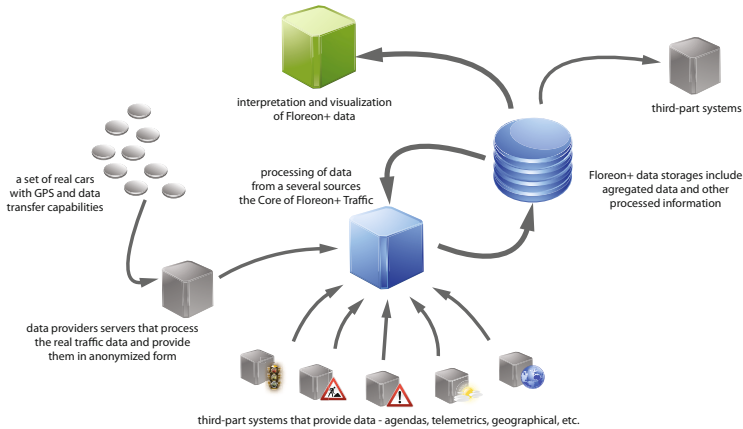


Fig. 1. Floreon⁺ Traffic

alternative as a valuable source is currently being developed in cooperation with operators of the JSDI system (A unified traffic information system in the Czech Republic).

Floreon⁺ Traffic system (scheme in Figure 1) output may be assessed and defined in different ways. Emphasis, however, is placed on its future potential. Aside from providing aggregated data for additional processing, we envision future benefits of this system in the following outputs, applications and expansions:

- Detailed statistical information for individual areas, streets, and roadways.
- Accountability for evaluating roads according to amount, quality and data source influenced by a given road.
- Visualization of entire lines of communication (camera systems, traffic accidents, data quality, etc.) in 2D and 3D imagery.
- Provision of functions and operations on data (searching for optimal roads, etc.).
- Processing and integrating data derived from various data sources (JSDI, telematic equipment, meteo. data, digital images, etc.).
- Analysis of historical data with current situational perspectives (automatic detection of problems on roadways based on comparisons of normal speeds and actual speeds on a given road).
- Application of higher level algorithms (e.g. anthill theory, flood wave algorithms) to create predictions and simulations usable in other systems, primarily from a traffic management perspective.

The applicability of this entire system lies directly in its dependence on data, its quality, quantity and heterogeneousness. The advantage in using data acquired directly from vehicles is in their surface coverage and immediate accessibility, regardless of their lack of relevance. Once this approach is combined with data derived from telematic sensors, for example (limited to local placement, but highly reliable and precise) it will be possible to create a total and real traffic situation model.

2.1 Collecting and Processing Data

As mentioned above, the project we are dealing with is based on collecting and processing data, derived directly from automobiles equipped with GPS functions that are able to send spacial data to a distant server in real time, where it is to be processed. The process then becomes a joint attempt at identifying a roadway where a vehicle is in motion. Using this identification, we place a spacial query above the roadway network. From data prepared in this manner, it becomes possible to effectively set statistics for roadway strain and appropriately visualize their results.

Raw data derived from a vehicle is sent at an amply high rate, i.e. at an average of 1 vehicle/10sec. With an adequate set of monitored vehicles (in our case, we have calculated with 20,000 vehicles), we attain the desired speed of 2,000 recordings/sec. Data derived from vehicles obviously does not flow at a constant rate. This issue, however, may be easily resolved with an appropriately-sized memory. One question remains, however, as to whether it is at all necessary to process raw data in real time. If we perform a rough estimate of memory complexity, say for a two-month interval of a monitored traffic situation, for a recorded surface and a vehicle's speed, we quickly reach hundreds of gigabytes of stored data. With data processed from an identified roadway, upon which their is vehicular movement, we achieve a much lower volume of data (this obviously depends on the strain of traffic on the given network of roads being processed).

Another factor is the speed at which data is processed. Whether we process data in real time or in batches, identifying roadways where there is vehicular movement is both time consuming and technologically demanding. We will also need to factor in several potential difficulties that will likely arise, of which the most significant may be GPS precision. For this, it will be necessary to carefully install error intervals to aid in partially eliminating this problem.

Another issue is the actual size of the roadway data file itself. In order to provide meaningful statistics, it will be necessary to include second class roadways. A file this size for the entire Czech Republic will be quite large and it will have to be queried for every processed recording from every vehicle being monitored. If we use the PostgreSQL tool² as a reference database, with PostGIS³ space expansion, and the commonly used GiST index, the speed a spacial query is sent to the roadway database will still be dependent upon the size of the data file itself. As described above, this data file is quite large and it is almost impossible to reach process speeds greater than 2,000 queries/sec. For this reason we have decided to develop our own method of indexing, based on scanning domains, making time complexity independent of the amount of indexed data.

3 Raster Indexing Spacial Data

The principle of raster indexing, in and of itself, is quite simple. In several aspects, it is very similar to bitmap index. Simply put, a spacial raster index is a very big picture

² PostgreSQL: <http://www.postgresql.org>

³ PostGIS: <http://postgis.refrations.net>

where all indexed recordings are drawn (geometry). Instead of using colors, recordings are "drawn" with an identifier. We can then effectively direct queries to this type of index (we can query the "color" of a specific pixel). The complexity of this type of query will obviously be constant. One issue with raster indexing is the limited amount of whole-number attributes upon which the index is performed. We must be able to clearly name all values of a given attribute in order to be able to use it for indexing. The raster index will then contain a paired key-value, where the keys are all value combinations of an indexed attribute and the value is the recording identifier that answers to a combination of attribute values. The index file itself does not contain a key because it is already possible to encode a key to a position within the index file. A raster index defined in this way has one evident disadvantage: it is not possible to index two recordings with identical index attribute combinations. A solution for this shortcoming is described below.

As far as spacial data is concerned, the whole-number issue presents quite a significant problem. This problem is, however, resolvable. We must set an area border for the raster index we are creating and the resolution with which we will be working. Here, we come up against one troublesome aspect of raster indexing: with resolution comes inaccuracy. When creating an index, it is necessary to consider the application for which the index is being used and, based on this, set the adequate resolution.

Constant complexity, for which point queries are carried out, is attained with a massively increased complex memory. For the project of measuring roadway strain, we have chosen a spacial index resolution of 5m/pixel. Since a raster index is actually a discretely modeled, spacial index data domain, the size of the index created surpasses the size of the data file many times over. Aside from the amount of space used up in the discs storage, this issue does not present any additional complications. We must, however, factor in the disadvantage of occupied space in the discs storage. Since the data file itself does not change, the index file does not change either. If only the data file changes, however, it still is not necessary to go through the entire index file; just going through the space within which the data file appeared is sufficient. The interval in the index file which needs to be renewed is again identifiable with constant complexity. Creating, querying and maintaining a raster index for spacial data is subject to several other rules and conditions, to be described in a later section.

3.1 Raster Index Construction

As described above, we have constructed a spacial index for analyzing roadway strain that features the following properties: the size of the indexed space has been chosen so that the entire Czech Republic is covered and the resolution has been set at 5m/pixel. The structure of the index file, in its basic form, is as follows: the file contains a heading with a configured index (see Table 1) followed by a frame saved in the identifier recording. Every frame contains a matrix of 1024×1024 identifiers (see Figure 2). The frames are saved in the index file in rows.

The indexed data file itself is performed as follows: we gradually go through the space defined by inputting intervals and queries as commonly done with spacial databases (with a resolution of 5m/pixel, that equates to a square with a 5,120m edge) and we draw (raster) the resulting object onto a tile. Tiles are gradually saved in the file.

Table 1. Header of the index file

byte	Value of
0..2	RID
3..6	Resolution (in meters, float)
7..10	Left Margin (SJTSK, float)
11..14	Right Margin (SJTSK, float)
15..18	Number of frames - width (int)
19..22	Number of frames - height (int)

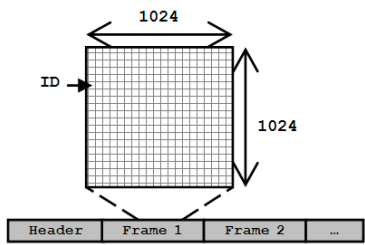


Fig. 2. Structure of the index file

The spacial point query for this type of index then simply lies in the index file based on its assigned coordinate. This results in the identifier object being located on the squares edge, measuring 5m (index resolution), despite its assigned coordinate being in the center of the queried square.

A spacial index constructed in this way, however, has several negative properties. For example, it is impossible to record more objects in one cell, which may present a big problem in the case of roadways. Another problem is the size and effectivity of the saved frame within the index file. Methods for countering these negative properties will be described in the following chapter.

3.2 Raster Index Optimizing

The first problem discussed, saving more files in one cell, can be easily solved by expanding cell recordings. In place of an identifier, we are going to store a paired-sign identifier. Within the one-byte field of signs, we can save information clarifying whether this position is just one object or if there are more objects present. Within the scope of a one-byte field signal, we can save information on whether this position is occupied by either one or more objects. If there is more than one object present, the object identifier is not saved here; its position is saved with the aid of a file where the situation is recorded in a query space (e.g. 3 objects with their identifiers). Another option for expanding cells saved on the frame is very closely related to indexing roadways. One valuable piece of information useful in expanding cells is the position of a queried point for a given roadway (defined as the distance from the start of the roadway, and running parallel to the entire queried point). Knowing this information enables us to

effectively analyze and save data about a strained roadway without having to separate it geometrically into smaller segments. Here, It is important to consider the fact that every expanded cell within the recording increases the-overall index file size. If we had an identifier saved as a 4-byte whole number, expanded by a sign byte and the road position (4 bytes, float), our index file size increases by more than half its original size. This is best possible option for optimizing memory complexity.

By performing simple statistical calculations on an index file, we find that in the case of an indexed roadway (generally all LineString data types), the final index file is more than 90% "empty". The aim of lowering memory complexity will then depend on our ability to avoid saving empty space in the index, if possible. The solution to this problem is to build a quadtree [6] above the matrix frame that is saved in the index file. All nodes in the quadtree will then contain a sign informing us whether or not the frame below it is empty. If not, every leaf contains a frame position in the index file (see Figure 3).

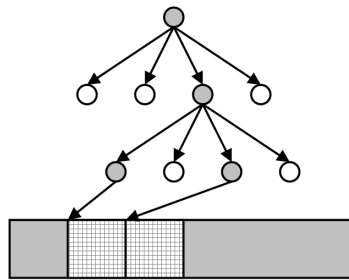


Fig. 3. Quadtree - two gray leaves are pointing at frame position in the index file

Implementing a quadtree actually increases the time complexity of a queried index, but this complexity is dependent on the size of the queried space, not on the amount of indexed data. If we have a raster index the size of n pixels, we will need $\log n$ layers in order to get to only one tree root. A certain disadvantage here may be the square index space and limitations in the size of the line value $2n$.

We do not need go all the way down to the root of the tree, though. We can just set an index from the matrix of the quadtree, which will allow us to partially eliminate the aforementioned shortcoming.

The next and last optimization for the system stems from the input of monitored traffic situations using GPS. GPS alone is able to determine a vehicles position within approx. 5m. Using maps to identify roadways with vehicular movement is also, at times, inaccurate; which is why the resulting overall error is added to the identification of a roadway error. This problem is well-resolved by defining the width of a road before re-rastering its geometries in the index file. Paralleling this example would be like drawing a fatter line over a fine line using a bigger marker. An index created in this manner is able to smooth out smaller errors, even though it may cause further issues, especially when two rows lie very near one another (e.g. a highway that has directions recorded as two differing geometries). This shortcoming may be eliminated by recording perpendicular

distances from the center of the roadway, again, at the cost of increasing the index files size.

4 Result from System FLOREON⁺ Traffic

We are currently receiving information on approx. 2,200 vehicles of different types (passenger cars, cargo trucks, lorries, etc.). This data is aggregated into a simple, speed-fast profile in real time around the entire Czech Republic. The overall coverage (with speed profile visualization) of the Czech Republic by floating cars used in FLOREON⁺ Traffic system is on the Figure 4.

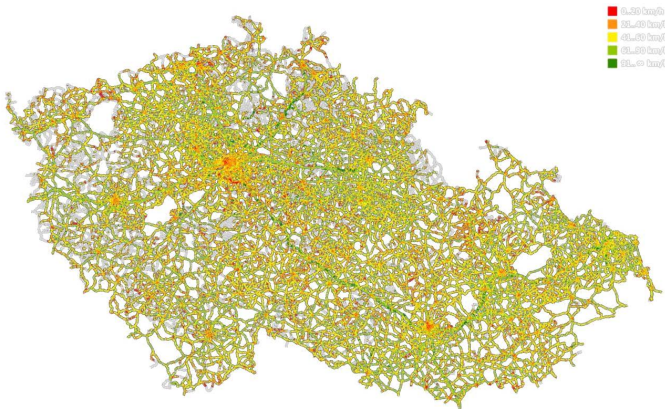


Fig. 4. The overall coverage of FLOREON⁺ data with speed profile visualization

The above mentioned approaches form a background for a whole operation of traffic data processing. It is performed by FLOREON⁺ system on the data storage layer, data mining layer, as well as visualization of results layer. It is important to note that the real traffic data from vehicles is not only source of information important for building traffic model and profiles. Also agenda data (traffic conditions and limitations, etc.), weather data, infrastructure data, live-cam data, etc. are important for that. The integration and combination of a wide set of data sources provide a real power to create effective and usable traffic model. So, this system is responsible for

- receiving the data from its providers (fleet car provider, provider of traffic infrastructure state, weather provider, etc.),
- aggregating the data based on time division and the location of a given records,
- preparing and performing the algorithms working with a wide range of data,
- concentrating all of these to the traffic model on micro, meso or macro levels and also providing simulation based on this model,
- preparing the outputs for following utilizations or integration to other systems.

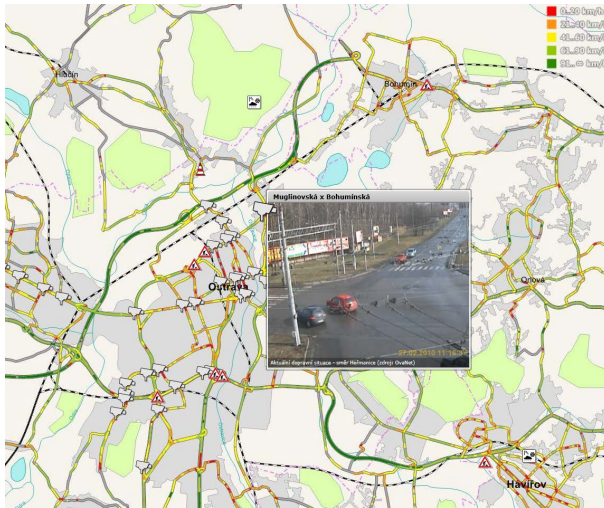


Fig. 5. Map visualization of speed profiles on the infrastructure (Ostrava, Czech Republic). The information on traffic conditions (roadwork, weather, etc.) is also included, as well as real-time camera view.

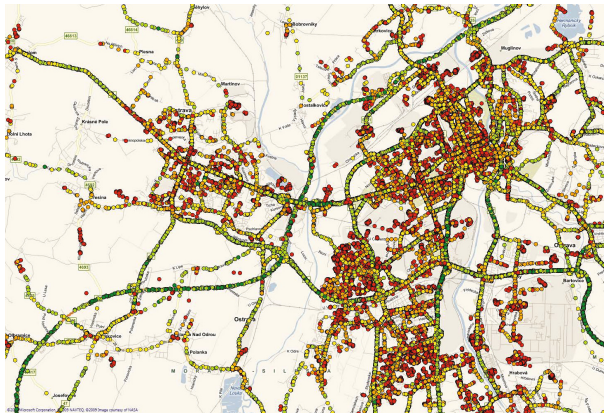


Fig. 6. A different type of floating cars data visualization is shown here. The anonymous locations and speeds are placed on the map and create basis for indexing algorithms. Not all of these 'circles' involve the speed limit of particular road due to infrastructure limitations and GPS inaccuracy.

The Figure 5 and Figure 6 depict a set of graphical outputs based on data processed by the FLOREON⁺ Traffic system. Also, the traffic visualization is a part of functionality of general FLOREON⁺ web interface⁴ which is developed with emphasis to user

⁴ FLOREON⁺: <http://www.floreon.eu>

experiences. The Microsoft Silverlight technology⁵ is used for that. The next figure is a screenshot taken from this our rich internet application.

The main pitfalls of this approach is the primary dependence on the floating car data [9,10], their quality and quantity. Amount of input data can be significantly reduced during the processing. Some data has not sufficiently informative value or are geographically inaccurate. This index based approach, where the individual anonymised records are assigned to a road, is loss (some amount of data is useless due to above mentioned reasons). The elementary knowledge of the motion vector of data sources would be an important contribution to the efficiency of data processing. Thanks to this, it should be possible to process not only one single location, but also the direction of the motion. It allows more complex traffic data processing. Unfortunately, this approach is unfeasible due to law on anonymity and data protection.

5 Conclusion

We presented the method of fast querying to the map of the road network based on the raster index. The method was experimentally and operationally verified by running FLOREON⁺ for three years, receiving information about approx. 2,200 vehicles every day. Obtained and aggregated data will be used for dynamic routing, which means data for routing will be updated accordingly to average situation on a given road section during day. In future work, we will be implementing this to our own version of routing, which is the combination of Highway Hierarchies [7,8] and heuristics [5,4] and also to our approach of road network updating during accidents. We presented earlier in [2]. Next step is preparation of other types of indexes for routing. Currently, we are creating the index for routing algorithms based on OpenStreetMap data⁶, which will also be used for optimized algorithms working on these data. There are sometimes incorrectly stated speed limits in data, so it's useful to replace them with average speeds on roads received by previously described method directly from vehicles.

Acknowledgment. This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

References

1. Barger, H.: Evaluation of a cellular phone-based system for measurements of traffic speeds and travel times: A case study from Israel. *Transportation Research Part C: Emerging Technologies* 15, 380–391 (2007)
2. Kromer, P., Martinovic, J., Radecký, M., Tomis, R., Snasel, V.: Ant Colony Inspired Algorithm for Adaptive Traffic Routing. In: *NaBIC 2011*, Salamanca, Spain (2011)
3. Gajdos, P., Radecký, P., Martinovic, J., et al.: Floreon plus system: Web applications with 3D visualization support. In: *NDT 2009*, Ostrava, Czech Republic (2009)
4. Goldberg, A.V., Harrelson, C.: Computing the shortest path: A* meets graph theory. Technical report, MSR-TR-2004-24 (2004)

⁵ Silverlight: <http://www.silverlight.net>

⁶ OpenStreetMap: <http://www.openstreetmap.org>

5. Hart, P.E., Nilsson, N.J., Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* 4, 100–107 (1968)
6. Samet, H.: *Foundations of Multidimensional and Metric Data Structures*. The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling. Morgan Kaufmann Publishers Inc., San Francisco (2005)
7. Sanders, P., Schultes, D.: Highway Hierarchies Hasten Exact Shortest Path Queries. In: Brodal, G.S., Leonardi, S. (eds.) *ESA 2005*. LNCS, vol. 3669, pp. 568–579. Springer, Heidelberg (2005)
8. Sanders, P., Schultes, D.: Engineering Highway Hierarchies. In: Azar, Y., Erlebach, T. (eds.) *ESA 2006*. LNCS, vol. 4168, pp. 804–816. Springer, Heidelberg (2006)
9. Schäfer, R.-P., Thiessenhusen, K.-U., Wagner, P.: A traffic information system by means of real-time floating-car data. In: *ITS World Congress*, pp. 1–8 (2002)
10. Sohr, A., Brockfeld, E., Krieg, S.: Quality of Floating Car Data. In: *12th World Conference for Transportation Research*, pp. 1–7. World Conference on Transport Research Society, Lisabon (2010)

A Comparative Study of Risk Assessment Methods, MEHARI & CRAMM with a New Formal Model of Risk Assessment (FoMRA) in Information Systems

Imed El Fray

West Pomeranian University of Technology, Szczecin
Faculty of Computer Science, Szczecin, Poland
ielfray@wi.zut.edu.pl

Abstract. In this article, we present a comparative study of a developed new formal mathematical model of risk assessment (FoMRA) with expert methods of risk assessment in the information systems (IS). Proposed analysis verified the correctness of theoretical assumptions of developed model. In the paper, the examples of computations illustrating the application of FoMRA and known and accepted throughout the world methods of risk assessment: MEHARI and CRAMM were presented and related to a specific unit of the public administration operating in Poland.

Keywords: risk assessment in information systems, risk assessment methods, MEHARI, CRAMM, FoMRA, comparative analysis of risk assessment methods.

1 Introduction

Continuous technological innovations and competition among existing and entering into the market organizations (firms) enable customer's access to a wider range of services and products delivered by the ICT systems [1,2,3]. A rapid development of the IT systems and growing acceptance of the Internet as a medium (channel) of products and services distribution, carries both benefits and risks [4,5]. A particular risk arises from the possibility of unauthorized disclosure, modification or removal of a larger amount of a significant information without leaving traces of an unauthorized access [6,7,8]. A particular attention should be paid nowadays to ensure an appropriate, understandable as a secure, access to such a type of systems [9,10].

The choice of methods to ensure the security of the IT systems in a given organization should be relevant to the type of risk. A transparent and proactive approach to the analysis and risk management may not only minimize risk but also allows achieving a competitive supremacy of the organization [11,12].

Among the methods of risk assessment, a particular attention is paid to the methods, which can be represented by means of the mathematical models. One of such models has been developed by us and described in detail in [13]. The advantage of this formal mathematical model of risk assessment (FoMRA) on the background of existing models (shown below) is that, it enables the performing of risk assessment

of the information system of the organization according to the ISO/IEC standards and OECD recommendations.

In this paper, a comparative analysis will be performed in order to demonstrate the correctness of the FoMRA theoretical assumptions on the background of the expert methods such as the CRAMM and MEHARI, accepted and applied by many professionals throughout the world.

2 Description of the Risk Evaluation Methods

Nowadays, amongst about 200 available methods of risk assessment and risk management [14], only few have found the acceptance of the market, including COBRA [15], COBIT [16], OCTAVE [17], CRAMM [18] and MEHARI [19]. The majority of these methods is based on know-how solutions developed by the independent or governmental organizations of different countries, and is assigned for the application in the governmental systems and public service organizations. These methods are not supported by proofs based on formal mathematical models, but they are only the collections of good practices within the IT Governance [20].

One of the first formalized methods of risk assessment for the information systems, approved as the government standard in the USA, is the Courtney method [21]. It considers the risk of information systems in terms of confidentiality, integrity and availability.

The Courtney method, being a standard in the USA, was developed by Fisher and others [22,23], but Parker was the first one, who eliminated the weak points of this method [24]. These weak points, as reported recently in [25,13], are related to the “human factor” which influences the risk of the incident. Parker, who applied mathematical knowledge and the experience of the IT experts, has proposed the risk analysis model containing five phases as described in [24].

The model above is an improved model proposed by Courtney. Most of elaborated quantitative, qualitative methods (graph-based, static and dynamic, relational and Markov) uses some or the majority of the assumptions of a standard model proposed by Parker [26,27,28,29,30]. These methods differ, however, in the approaches to the identification and classification of the assets, vulnerabilities and risks, the risk value assessment, the choice of countermeasures, etc. This fact makes the most of the methods presented above to move in quite different directions, even if the final goal seems to be the same [14]. In most cases, it is thus impossible to directly compare results generated by two different methods, and an indirect comparison is hard and time-consuming, even if conversion mechanisms are obtainable. The important questions thus become whether these more or less widely used methods are competent? whether they can properly describe any IT system? and whether they effectively ensure declared compliance with standards?

Some answers to these questions were provided in the paper [13] we propose a formal model for risk assessment (FoMRA) based on the experience of experts who created the MEHARI method and it complies with the requirements and standards' guidelines for the security of the information systems [simplified model is presented in Section 2.1].

In this paper, the FoMRA model described in [13], will be a subject of the comparative analysis aiming, as mentioned above, to demonstrate the correctness of the theoretical assumptions of the model against the well-known and widely used methods of risk assessment. The experimental results of the risk analysis should confirm whether the FoMRA is meaningful, and if it truly describes any information system, and whether these results are comparable with the results obtained from other methods.

2.1 Simplified Formal Model of Risk Analysis (FoMRA)

The defined mathematical structures of the standard formal model of risk assessment (FoMRA), were used to precisely define a graph for calculating risk values [13], and to define an algorithm of its construction. Briefly, let A be a set of some assets¹:

$$A = \{a_i : i = 1, \dots, n_A\} \quad (1)$$

Additionally, let us consider the following finite sets:

$$V = \{v_j : j = 1, \dots, n_V\} \text{ - a set of vulnerability classes,} \quad (2)$$

$$T = \{t_k : k = 1, \dots, n_T\} \text{ - a set of threat classes,} \quad (3)$$

$$S = \{s_l : l = 1, \dots, n_S\} \text{ - a set of risk scenarios,} \quad (4)$$

$$DP = \{dp_s : s = 1, \dots, n_{DP}\} \text{ - a set of measures reducing a potentiality,} \quad (5)$$

$$DI = \{di_t : t = 1, \dots, n_{DI}\} \text{ - a set of measures reducing an impact.} \quad (6)$$

The above sets define classes of system assets, vulnerabilities concerning threats, classes of threats for assets, risk scenarios and measures reducing potentialities and impacts of threats resulting from assets losses.

Let us assume that there is a given and ordered set \mathfrak{R} of n -values for the arguments in IS system (according to [13]), corresponding to sets A , V , T , DP , DI , and M , W , where M and W are subsequent arrays and values reducing threats, risks and consequences of risk.

In this set $\mathfrak{R} = [r_{\min}, r_{\max}] \subset N$, where N is the set of natural numbers, additional auxiliary functions are defined:

- $value_A : A \rightarrow \mathfrak{R}^* \times \mathfrak{R}^* \times \mathfrak{R}^*$ - which assigns to a given asset $a \in A$ the values of three basic security parameters: *CIA* (*Confidentiality, Integrity, Availability*)
- $value_V : V \rightarrow \mathfrak{R}^* \times \mathfrak{R}^* \times \mathfrak{R}^*$ - which assigns the values of three parameters depending on *AEV* (*Accident, Error, Voluntary*) to a given natural vulnerability $v \in V$ (a so-called “natural exposure”, independent from the security measures used)

¹ The numbers $n_A, n_V, n_S, n_T, n_{DP}, n_{DI}$ further used are some relevantly great natural numbers.

- For $value_A$, $value_V$ set $\mathfrak{R}^* = \mathfrak{R} + \{null\}$, where $null \in \mathfrak{R}^*$ is the neutral value, what means that such a function's argument value does not have the defined feature, and no value can therefore be assigned to it.
- $value_T : T \rightarrow \mathfrak{R}$ - assigning a given threat $t \in T$ to t value,
- $value_{DP} : DP \times S \times N \rightarrow \mathfrak{R}$ - assigning given measures reducing the threat potentiality $d_p \in DP$ to d_p value.

Additionally, in order to determine the risk values $W^{s,a}$ for any risk scenario s assigned to an asset a the following arrays are predefined:

- an array of potentiality reduction $M_{pot}^{s, n \times n \times n}$, which makes a declared value of measure-reducing potentialities $W_{pot}^s[i, j, k] \in \mathfrak{R}$ dependent on $CM_{s,j}$ (for particular $j = dp_1, \dots, dp_{n_{dp}}$) and vulnerability value $value_V(v)$,
- an array of impact $M_{imp}^{s, n \times n \times n}$, which makes a declared value of measure-reducing impacts $W_{imp}^s[i, j, k] \in \mathfrak{R}$ dependent on $CM_{s,j}$ (for particular $j = di_1, \dots, di_{n_{di}}$),
- an array of impact reduction $M_{imp}^{s,a, n \times n}$, which makes a declared value of measures-reducing impact $W_{imp}^{s,a}[i, j] \in \mathfrak{R}$ dependent on the value W_{imp}^s determined from the array $M_{imp}^{s, n \times n \times n}$ and the value of an asset $value_A(a)$,
- an array of risk $M^{s,a, n \times n}$, which makes a declared value of risk $W^{s,a}[i, j] \in \mathfrak{R}$ dependent on the value W_{pot}^s determined from the array $M_{pot}^{s, n \times n \times n}$ and value $W_{imp}^{s,a}$ determined from the array $M_{imp}^{s,a, n \times n}$.

$CM_{s,j} \in \mathfrak{R}$ is a weighted value of the measure reducing the potentiality and impact of some threat. The following formula (7) shows how to calculate values for potentiality and impact actions:

$$CM_{s,j} = \left[(r_{\max} - r_{\min}) \cdot \frac{\sum R_i \times P_i}{\sum P_i} + r_{\min} + 0.5 \right] \quad (7)$$

where:

- $j \in DP \cup DI$ – represents implemented measure/countermeasure,
- $\lfloor x \rfloor$ - indicates the rounding down of the result x to the number belonging to the set \mathfrak{R} (to the infimum of x in this set),
- R_i – is an answer to an audit question (the value 1 or 0),
- $P_i = value_X(j, s, no(R_i))$ – is a value assigned to an i -th question, where $X = DP$ or DI , which depends on the defined measure type j scenario s and a number of the question $no(R_i)$ associated with the answer R_i .

The values of defined arrays M_{pot}^s , M_{imp}^s , $M_{imp}^{s,a}$ and $M^{s,a}$ should depend on the criticality of business processes in a given organization, and should not be "rigidly" taken, as proposed in most methods such as CRAMM and MEHARI. The criticality of the process depends on vulnerability values $value_v(v)$, assets $value_A(a)$ and the effectiveness of the implemented measures reducing potentialities $DP(dp_s : s = 1, \dots, n_{DP})$ and impacts $DI(dp_t : t = 1, \dots, n_{DI})$.

For these specific arrays, the following sets of arrays are also determined:

$$M_{pot} = \bigcup_s : \exists_{dp \in DP} (s, dp) \in \overline{DP} \{M_{pot}^s\} \quad (8)$$

$$M_{imp} = \bigcup_s : \exists_{di \in DI} (s, di) \in \overline{DI} \{M_{imp}^s\} \quad (9)$$

$$M_{imp}^a = \bigcup_s : \exists_{di \in DI} (s, di) \in \overline{DI} \{M_{imp}^{s,a}\} \quad (10)$$

$$M = \bigcup_{(a,s) \in A \times S} \{M^{s,a}\} \quad (11)$$

The above brief description of the FoMRA complies with the fundamental requirements of the ISO standard: ISO/IEC 27005:2011 - Security techniques - Information security risk management. The FoMRA allows to conduct completely a comparative analysis with other methods, as shown below.

2.2 Description of Methods Used in Comparative Analysis

The choice of the CRAMM and MEHARI methods for comparative analysis has been made to demonstrate the correctness of the theoretical assumptions of the FoMRA [13]. These methods are widely accepted by the risk analysis and security management experts.

The risk value determined by the CRAMM method is dependent on assets value, threats and vulnerabilities of the system (Fig 1). By applying this method, a list of countermeasures aiming at reducing risks in information security is created.

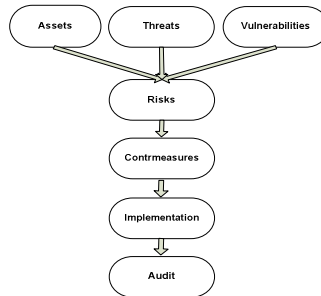


Fig. 1. Graphical representation of the CRAMM risk assessment method

To determine risk value, CRAMM method applies heptavalent array of risk (Tab.1).

where:

- 1, 2 : negligible risk,
- 3, 4 : tolerable risk,
- 5,6 : inadmissible risk,
- 7 : intolerable risk.
- Depending on the risk value, CRAMM enables the selection of countermeasures from 70 groups for a given scenario, making it a dedicated method for large organizations and enterprises. In the case of the SME sector, the choice of countermeasures may not be optimal, since risk does not match the scale of the risk of failure due to the scale of the enterprise.

Table 1. The array of risk value according the CRAMM method

Threats	VL	VL	VL	L	L	L	M	M	M	H	H	H	VH	VH	VH
Vuln.	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H
Assets/Value	1	1	1	1	1	1	1	1	2	1	2	2	2	2	3
	2	1	1	2	1	2	2	2	3	2	3	3	3	3	4
	3	1	2	2	2	2	2	3	3	3	3	4	3	4	4
	4	2	2	3	2	3	3	3	4	3	4	4	3	4	4
	5	2	3	3	3	3	4	3	4	4	4	4	4	4	5
	6	3	3	4	3	4	4	4	4	5	4	5	5	5	6
	7	3	4	4	4	4	5	4	5	5	5	5	6	5	6
	8	4	4	5	4	5	5	5	6	5	6	6	6	6	7
	9	4	5	5	5	5	6	5	6	6	6	7	6	7	7
	10	5	5	6	5	6	6	6	6	6	7	7	7	7	7

V.L – Very Low, L – Low, M –Medium, H – High, V.H – Very High.

According to various reviews [32,33], CRAMM as a commercial tool, should be used only by the experienced users, since it generates too much information, it is inflexible and slow. The full analysis can take months, instead of several days.

Unlike CRAMM, the MEHARI method is available as a Know-How knowledge base (in the Excel File) related to threats, vulnerabilities and threat scenarios assessment.

The MEHARI method is based on the knowledge of assets, vulnerabilities and threats identification and classification, and the assessment of risk levels (Fig. 2).

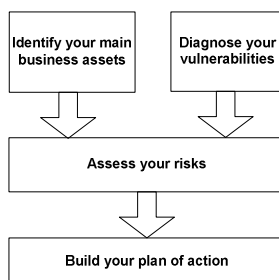


Fig. 2. Graphical representation of the MEHARI risk assessment method

To determine risk value, the MEHARI method applies tetravalent array of a risk (Tab. 2).

Table 2. The array of risk value according to the MEHARI method

Impact				
4	2	3	4	4
3	2	3	3	4
2	1	2	2	3
1	1	1	1	2
	1	2	3	4
Potentiality				

where:

- 1 : negligible risk,
- 2 : tolerable risk,
- 3 : inadmissible risk,
- 4 : intolerable risk.

The risk value in the MEHARI method depends on the potentiality and impact values for each of the identified risk scenarios. According to various authors [32,33], the MEHARI method is flexible and dedicated to small and large organizations. The disadvantage of this method is the lack of data bases of countermeasures which are reducing risks in information security systems. Moreover, new upgrade issued in the year 2012, delivering new knowledge base of vulnerabilities, threats, etc., made this method even more complicated and a little more time-consuming.

Tables 1 and 2 determine the risk values estimation for each type of the organization for each of the methods mentioned above . They indicate the risk that a given organization takes into account as: (i) intolerable (risk demanding the immediate implementation of countermeasures, despite of the organization budget and security plans), (ii) inadmissible (risk must be eliminated or minimized sooner or later, according to the established organization budget and security plans), or (iii) tolerable (low or insignificant risks - depending on the organization's security policy).

3 Conditions and Results of the Experiment

To perform a comparative analysis demonstrating the correctness of the theoretical assumptions of the FoMRA [12], we have to:

- establish an uniform scale of risk value array (CRAMM, MEHARI, FoMRA),
- use the same assets, vulnerabilities, threat/risk scenarios for various methods of risk analysis.

It was assumed for the analysis that the scale of risk value is in the range $<1 - 4>$. The assumed scale is not dictated by any requirements, only for ease operation of the quantitative records (for the MEHARI method and FoMRA), dissimilar to quantitative-qualitative as CRAMM.

It was also assumed that risk values (MEHARI and FoMRA *versus* CRAMM) should be interpreted after transformation as follows:

- for risk value: $1 = (1,2)$; $2 = (3,4)$; $3 = (5,6)$; $4 = 7$ - and they correspond to the set of values given in (CRAMM and MEHARI). The scale of risk values according to the FoMRA is flexible [17] and can be matched to any of the above methods.

Additionally, the following classification system was used:

- for assets: 1 - less important, 2 - important 3 - very important, 4 - critical
- vulnerability/threat: 1 = Very Low/Low, 2 = Medium, 3 = High, 4 = Very High

The data derived from the analysis of the IT security of the administrative unit operating in Poland were used to perform the comparative analysis.

FoMRA was used to perform a pre-audit [13] which allowed to identify confidentiality (C), integrity (I), and availability (A). Table 3 shows an example of the identified CIA parameters.

Table 3. The results of the pre-audit of resources and vulnerabilities for a given organization

A	Assets	Value _s (a)			V	Vulnerability	Value _v (v)		
		C	I	A			A	E	V
a ₁	Data files or data bases accessed by applications	4	4	3	v ₁	Distorted data entry or fiddling of data	null	null	3
a ₅	Written or printed information and data kept by users and personal archives	2	2	2	v ₃	Intentional erasure (direct or indirect), theft or destruction of program or data containers	null	null	3
a ₆	Main systems, servers hosting applications and their peripheral equipments, shared file servers	null	null	4	v ₅	IT or telecom equipment breakdown	2	null	null
a ₉	Application software, package or middleware (executable code)	null	null	2	v ₁₅	Bug in application program	null	4	null
...

For each of the threat scenarios (s_1, s_2, \dots, s_n) the risk value $W^{s,a}$ was calculated. To calculate the $W^{s,a}$ value, the results from the audit questionnaire were used. The audit concerned the implemented dissuasive measures, preventing from potential threats (measures defined in formula 5) and protective, preventive as well as palliative measures, depending on the threat type (measures reducing threat, formula 6). The questionnaire results taken from the MEHARI knowledge base [19] have been used to perform audit. Similar questionnaires are also available from the OCTAVE [17], EBIOS [34], CRAMM [18], etc. One should note that the content of the audit questionnaires may not perfectly match between the analyzed methods (certain audit questionnaires from the MEHARI contain more details and higher number of implemented countermeasures as compared to the CRAMM audit questionnaires and *vice versa*, for the same hazard risk scenarios).

The situation may happen when one or more of the audit questionnaires will be covered in one method, while not in the other one (e.g., audit questionnaires related to the recovery measures in the MEHARI method do not have the coverage in the CRAMM method, because such measures are not taken into account there). This situation can ultimately affect the outcome of risk assessment (e.g. risk values).

Table 4 shows a section of the questionnaire for the audit of dissuasive measures against theft of archives in an office for asset a₅ - written or printed information and data kept by users and personal archives, susceptibility v₃ - intentional erasure (direct or indirect), theft or destruction of a program or data containers and the threat t₂ - loss of data files or documents: theft of data media.

Table 4. Section of the audit questionnaire related to the dissuasive measures of the potential attackers against theft of archival documents

Monitoring of protected office areas	Response (0/1)	Value _{pi}
Is there a complementary video surveillance system, complete and coherent, for protected office areas, able to detect movement and abnormal behaviour?	1	4
In the case of an alarm, does the surveillance team have the possibility of sending out an intervention team without delay to verify the cause of the alarm and to take appropriate action?	1	2
Has the security team sufficient resources to cover the eventuality of multiple alarms set off intentionally?	0	4
Is video surveillance material recorded and kept for a long period?	1	1
Is the intrusion detection system itself under surveillance (alarm in the case of shutdown, video auto-surveillance etc.)?	1	2
Are procedures for surveillance and intervention in the case of abnormal behaviour audited regularly?	1	2

According to formula 7, we calculate the weighted value of measures $CM_{s,j}$ ($CM_{s,j}=1$ means that the measure is ineffective and $CM_{s,j} = 4$ means that it is very effective) for each identified scenario s . The following example shows the calculation of $CM_{s,j} = dp_1$ for dissuasive measures taken from table 4 (scenario s_{15} -value in bold):

$$CM_{s_g, j = dp_1} = \left[(4 - 1) \times \frac{(1 \cdot 4 + 1 \cdot 2 + 0 \cdot 4 + 1 \cdot 1 + 1 \cdot 2 + 1 \cdot 2)}{15} + 1 \right] = 3$$

where:

1/0 - means Yes / No

Table 5 shows the calculation of the weighted values of the $CM_{s,j}$ for exemplary scenarios of threats. As can be seen from the table, some of $CM_{s,j}$ measures have value equal to 1. This value may be the result of calculation as above or can be taken arbitrarily in the absence of such measures (scenario s_4 - bold values). For example, in order to prevent copying the application data files (s_4) by a potential hacker, we can only use the protective measures against copying and/or measure-reducing impacts of copying. The use of effective dissuasive measures against potential hacker (to discourage him from performing an attack) is, however, minimal or impossible.

Table 5. An example of the calculated weighted values of $CM_{s,j}$ for exemplary threat scenarios

N°	Scenario -S	Parameters' & Value _A (a)		Parameters' & Value _v (v)		$CM_{s,j=dp1} = value_{DP}(dp_1)$	$CM_{s,j=dp2} = value_{DP}(dp_2)$	$CM_{s,j=di1} = value_{DI}(di_1)$	$CM_{s,j=di2} = value_{DI}(di_2)$	$CM_{s,j=di3} = value_{DI}(di_3)$
		CIA	value	AEV	value					
S ₄	Repeated copy of application data files, by a hacker connecting from outside to an open port for network remote maintenance	C	a ₁ =4	V	v ₂ =3	1	3	4	1	1
S ₁₅	Loss of data files or documents: theft of archives in an office	A	a ₅ =2	V	v ₃ =3	3	2	1	2	3
...

Further procedure is the $W_{pot}^s, W_{imp}^s, W_{imp}^{s,a}$ calculation and $W^{s,a}$ mentioned above.

For this purpose, the standard values of the risk arrays $M_{pot}^s, M_{imp}^s, M_{imp}^{s,a}$ and $M^{s,a}$ described in detail in [13] were considered. Tables 6,7 and 8 illustrate all the necessary data to calculate the risk values for the selected threats in an exemplary administrative unit. Tables cover all the CIA safety parameters and types of actions using vulnerabilities AEV within FoMRA. The table also includes the results of analysis performed according to the MEHARI and CRAMM methods. During the risk

assessment with the use of CRAMM and MEHARI methods, the system of resources, vulnerabilities and risks classification was used. The scale of risk values was set according to the requirements of both methods. The audit questionnaires from each of the methods were used during the analysis in accordance to the requirements described above (with the same resources, vulnerabilities, and threat / risk scenarios).

The results obtained with the use of the FoMRA for 14 out of the 21 scenarios presented in Table 7, Table 8 and Figure 5 are comparable with those obtained using CRAMM (Fig. 3). In turn, 16 out of the 21 scenarios are comparable with those obtained using MEHARI (Fig. 4). Analyzing the results derived from the CRAMM and MEHARI and given in Table 8, we received comparable results, for which 14 out of the 21 scenarios overlaps.

As can be seen from Table 7 and Table 8, from the 21 scenarios representing approximately 10% of all threat scenarios [19], 12 overlap (they give the same results for the three methods). This result contradicts the statement made by the authors [14] that in most cases it is impossible to compare directly the results generated by two different methods.

Referring to the statement on audit questionnaires related to the recovery measures from the MEHARI method having no coverage in CRAMM method, we performed the analysis of the results from Table 7 for the weighted value, $CM_{s,j} = di_2$.

Table 6. The identified and classified resources and vulnerabilities for each threat scenario

S	Scenarios	Assets			vulnerability		
		CIA	a	value _a (a)	AEV	v	value _v (v)
s ₁	Deliberate erroneous data input by a staff member usurping an authorized user's identity	I	a ₁	4	M	v ₁	3
s ₂	Deliberate substitution of data media, by an unauthorized person	I	a ₁	4	M	v ₁	3
s ₃	Theft of application data media during production, by a person authorized to handle the media	C	a ₁	4	M	v ₂	3
s ₄	Repeated copy of application data files, by a hacker connecting from outside to an open port for network remote maintenance	C	a ₁	4	M	v ₂	3
s ₅	Access to a system and copy of application data files, by a staff member using a security breach left open after a maintenance operation	C	a ₁	4	M	v ₂	3
s ₆	Accident during data processing -- Alteration of sensitive data	I	a ₁	4	A	v ₁₁	3
s ₇	Accidental loss of business related data (sensitive data) due to obsolescence or pollution	D	a ₁	3	A	v ₁₂	2
s ₈	Data integrity distortion during transmission on the WAN/LAN network, by a (remote) hacker	I	a ₃	4	M	v ₁	3
s ₉	Erroneous message sent by a staff member usurping the identity of another person, with a forged signature	I	a ₃	4	M	v ₁	3
s ₁₀	Diversion of sensitive information by a system administrator, using access to user data not erased after use	C	a ₃	2	M	v ₆	3
s ₁₁	Interception of sensitive information transferred over the LAN by a network administrator modifying a network equipment	C	a ₃	2	M	v ₆	3
s ₁₂	Interception of sensitive information transferred between a nomadic user and the internal network, by listening to the exchanges	C	a ₃	2	M	v ₆	3
s ₁₃	Interception of sensitive information: eavesdropping electromagnetic emission	C	a ₃	2	M	v ₆	3
s ₁₄	Short circuit resulting in a fire with important damages on WAN/LAN network equipment	D	a ₄	4	A	v ₁₃	2
s ₁₅	Loss of data files or documents: theft of archives in an office	D	a ₅	2	M	v ₃	3
s ₁₆	Malicious alteration of the expected functionalities of an application due to a logic bomb or back door laid by operation staff	I	a ₉	2	M	v ₇	3
s ₁₇	Deliberate modification of an application by the maintenance	I	a ₉	2	M	v ₇	3
s ₁₈	Unintentional degradation of performances for applications after software maintenance operation	D	a ₉	3	E	v ₂₀	2
s ₁₉	Extended network configurations erased or polluted by a non operational staff member	D	a ₁₁	3	M	v ₉	2
s ₂₀	Departure of strategic personnel	D	a ₁₃	2	E	v ₁₆	3
s ₂₁	Remote attack of a third organization by internal personnel using authorized connections to the organization	D	a ₁₅	2	M	v ₁₇	3

The analysis showed that among the 13 scenarios examined by the FoMRA, where the value of the $CM_{s,j} = di_2 > 1$ (recovery measure - this means that the audited organization possesses the insurance covering some cost of property, assets, etc. damage), seven scenarios were identified (5 - Integrity, 2 - Availability) which do not match the results derived from CRAMM (Tab. 8).

Taking into account the method of $W^{s,a}$ evaluation, as described in details in [13], it was observed that changes in $CM_{s,j} = di_2$ for ($s_1, s_2, s_6, s_7, s_8, s_{14}, s_{16}$) scenarios from Table 7, impose values changes in assigned arrays (Table 9 - $CM_{s,j} = di_2 = 1$ dark gray color, $CM_{s,j} = di_2 > 1$ gray).

Table 7. Calculated risk values using FoMRA for selected scenarios on the example of administrative unit operating in Poland

S	Risk value FoMRA								
	$CM_{s,indp1}$	$CM_{s,indp2}$	W_{pot}	$CM_{s,indl1}$	$CM_{s,indl2}$	$CM_{s,indl3}$	W_{imp}	$W^{s,a}_{imp}$	$W^{s,a}$
s_1	1	2	3	2	2	1	2	2	2
s_2	4	2	2	2	2	1	2	2	2
s_3	3	1	3	1	1	1	4	4	4
s_4	1	3	2	4	1	1	3	3	3
s_5	3	3	2	2	1	1	3	3	3
s_6	1	3	2	3	2	4	2	2	2
s_7	1	2	2	3	2	3	2	2	2
s_8	1	2	3	3	2	1	2	2	2
s_9	1	2	3	1	2	1	4	4	4
s_{10}	2	3	2	2	1	1	3	2	2
s_{11}	2	2	3	3	1	1	3	2	2
s_{12}	1	3	2	1	1	1	4	2	2
s_{13}	1	2	3	1	1	1	4	2	2
s_{14}	1	3	2	3	3	3	2	2	2
s_{15}	3	2	2	1	2	3	3	2	2
s_{16}	2	4	1	2	2	1	2	2	1
s_{17}	3	3	2	1	2	1	4	2	2
s_{18}	1	2	2	1	2	3	3	3	3
s_{19}	1	3	2	2	2	3	3	3	3
s_{20}	1	1	3	1	2	4	3	2	2
s_{21}	3	2	2	2	1	1	3	2	2

Table 8. Calculated risk values using CRAMM and MEHARI for selected scenarios on the example of administrative unit operating in Poland

S	Risk value CRAMM					Risk value MEHARI		
	$value_s(a)$	$value_v(v)$	$value_T(t)$	$W_{CRAMM <1,7>}$	$W_{CRAMM <1,4>}$	W_{pot}	W_{imp}	W_{MEHARI}
s_1	10	H	M	6	3	3	4	4
s_2	9	M	M	6	3	2	4	3
s_3	9	M	V.H	7	4	3	4	4
s_4	9	M	M	6	3	2	3	3
s_5	9	L	H	6	3	2	3	3
s_6	9	M	M	6	3	2	3	3
s_7	9	M	M	5	3	2	2	2
s_8	10	H	M	6	3	3	2	2
s_9	10	H	V.H	7	4	3	4	4
s_{10}	3	L	H	3	2	2	2	2
s_{11}	4	M	M	3	2	3	2	2
s_{12}	4	M	V.H	4	2	2	3	3
s_{13}	4	H	V.H	4	2	3	2	2
s_{14}	10	L	M	6	3	2	2	2
s_{15}	3	L	H	3	2	2	2	2
s_{16}	4	L	M	3	2	1	2	1
s_{17}	4	L	M	3	2	2	2	2
s_{18}	8	H	H	6	3	2	2	2
s_{19}	6	M	H	5	3	2	3	3
s_{20}	5	H	H	4	2	3	2	2
s_{21}	4	M	H	4	2	2	2	2

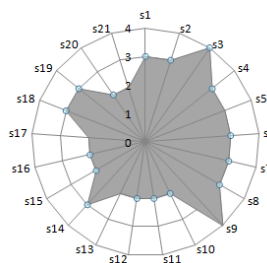


Fig. 3. Risk value assessed from CRAMM method

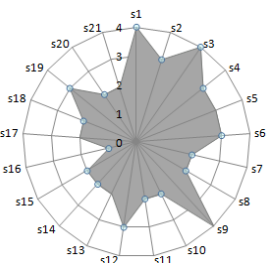


Fig. 4. Risk value assessed from MEHARI method

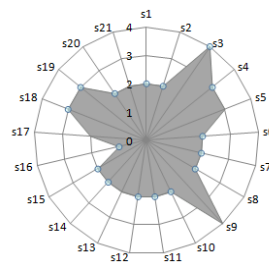


Fig. 5. Risk value assessed from FoMRA method

The values, W_{pot}^s , W_{imp}^s , $W_{imp}^{s,a}$ and $W^{s,a}$ are read from diagonals of M_{pot}^s , M_{imp}^s , $M_{imp}^{s,a}$, $M^{s,a}$ arrays.

Table 9. Impact Arrays M_{imp}^s in relation to CIA parameters

Integrity

CM _{s,j=di1} - protective =4					
CM _{s,j=di2} -recovery	4	1	1	1	1
	3	1	1	1	1
	2	2	2	2	2
	1	3	3	3	3
	1	2	3	4	
CM _{s,j=di3} - palliative					

CM _{s,j=di1} - protective =3					
CM _{s,j=di2} -recovery	4	1	1	1	1
	3	1	1	1	1
	2	2	2	2	2
	1	3	3	3	3
	1	2	3	4	
CM _{s,j=di3} - palliative					

CM _{s,j=di1} - protective =2					
CM _{s,j=di2} -recovery	4	2	2	1	1
	3	2	2	1	1
	2	2	2	2	2
	1	3	3	3	3
	1	2	3	4	
CM _{s,j=di3} - palliative					

CM _{s,j=di1} - protective =1					
CM _{s,j=di2} -recovery	4	4	3	2	1
	3	4	3	2	1
	2	4	3	2	2
	1	4	3	3	3
	1	2	3	4	
CM _{s,j=di3} - palliative					

Availability

CM _{s,j=di1} - protective =4					
CM _{s,j=di2} -recovery	4	3	3	2	1
	3	3	3	2	1
	2	3	3	2	2
	1	3	3	3	3
	1	2	3	4	
CM _{s,j=di3} - palliative					

CM _{s,j=di1} - protective =3					
CM _{s,j=di2} -recovery	4	3	3	2	1
	3	3	3	2	1
	2	3	3	2	2
	1	3	3	3	3
	1	2	3	4	
CM _{s,j=di3} - palliative					

CM _{s,j=di1} - protective =2					
CM _{s,j=di2} -recovery	4	3	3	2	1
	3	3	3	2	1
	2	3	3	2	2
	1	3	3	3	3
	1	2	3	4	
CM _{s,j=di3} - palliative					

CM _{s,j=di1} - protective =1					
CM _{s,j=di2} -recovery	4	4	3	2	2
	3	4	3	3	2
	2	4	3	3	3
	1	4	3	3	3
	1	2	3	4	
CM _{s,j=di3} - palliative					

Given the additional asset values (a_1, a_3, a_4, a_9) attributed to scenarios ($s_1, s_2, s_6, s_7, s_8, s_{14}, s_{16}$) from Table 6, it was noticed that for assets (a_1, a_3, a_4), classified as very important or critical, the change in the value W_{imp}^s (including $CM_{s,j} = di_2 > 1$, gray color, and $CM_{s,j} = di_2 = 1$ dark gray color) derived from M_{imp}^s array, affect the changes of values in Table 10. All other assets assigned to scenarios (in our case, $a_9 \rightarrow s_{16}$), classified as minor or major ones, do not affect value changes. This situation may be related to asset value (classified as significant), which is comparable to or lower than the security cost.

Finally, it can be concluded that by considering recovery measures for losses related to some assets revealed when determining the risk value, $W^{s,a}$ from Tab. 11, for 6 out of the 7 scenarios derived from the FoMRA, a significant effect on the resulting difference in risk values between this model and a CRAMM method was observed.

To unambiguously confirm the above statement, an explanation concerning the lack of changes in risk value reduction, $CM_{s,j} = di_2 > 1$ in 6 among 13 scenarios ($s_9, s_{15}, s_{17}, s_{18}, s_{19}, s_{20}$), needs to be found.

Table 10. Array reducing impact

$Value_{s_i}(a)$				
4	1	2	3	4
3	1	2	3	3
2	1	2	2	2
1	1	1	1	2
	1	2	3	4
W_{imp}^s				

Table 11. Array of risk value

$W_{imp}^{s,a}$				
4	2	3	4	4
3	2	3	3	3
2	1	2	2	4
1	1	1	1	2
	1	2	3	4
W_{pot}^s				

For scenarios (s_9, s_{17}) associated with the Integrity parameter (Tab.12), at the absence of the protective measures, changes in the weighted value, $CM_{s,j} = di_2$ do not affect the obtained value, W_{imp}^s from M_{imp}^s array (dark gray color for the $CM_{s,j} = di_2 = 1$, gray color for $CM_{s,j} = di_2 > 1$).

Table 12. Impact Arrays M_{imp}^s in relation to the CIA parameters**Integrity**

$CM_{s_i=di_2}$ recovery	$CM_{s_i=di_1} - \text{protective}=4$				
	4	1	1	1	1
	3	1	1	1	1
	2	2	2	2	2
	1	3	3	3	3
		1	2	3	4
$CM_{s_i=di_3} - \text{palliative}$					

$CM_{s_i=di_2}$ recovery	$CM_{s_i=di_1} - \text{protective}=3$				
	4	1	1	1	1
	3	1	1	1	1
	2	2	2	2	2
	1	3	3	3	3
		1	2	3	4
$CM_{s_i=di_3} - \text{palliative}$					

$CM_{s_i=di_2}$ recovery	$CM_{s_i=di_1} - \text{protective}=2$				
	4	2	2	1	1
	3	2	2	1	1
	2	2	2	2	2
	1	3	3	3	3
		1	2	3	4
$CM_{s_i=di_3} - \text{palliative}$					

$CM_{s_i=di_2}$ recovery	$CM_{s_i=di_1} - \text{protective}=1$				
	4	4	3	2	1
	3	4	3	2	1
	2	4	3	2	2
	1	4	3	3	3
		1	2	3	4
$CM_{s_i=di_3} - \text{palliative}$					

Availability

$CM_{s_i=di_2}$ recovery	$CM_{s_i=di_1} - \text{protective}=4$				
	4	3	3	2	1
	3	3	3	2	1
	2	3	3	2	2
	1	3	3	3	3
		1	2	3	4
$CM_{s_i=di_3} - \text{palliative}$					

$CM_{s_i=di_2}$ recovery	$CM_{s_i=di_1} - \text{protective}=3$				
	4	3	3	2	1
	3	3	3	2	1
	2	3	3	2	2
	1	3	3	3	3
		1	2	3	4
$CM_{s_i=di_3} - \text{palliative}$					

$CM_{s_i=di_2}$ recovery	$CM_{s_i=di_1} - \text{protective}=2$				
	4	3	3	2	1
	3	3	3	2	1
	2	3	3	3	2
	1	3	3	3	3
		1	2	3	4
$CM_{s_i=di_3} - \text{palliative}$					

$CM_{s_i=di_2}$ recovery	$CM_{s_i=di_1} - \text{protective}=1$				
	4	4	3	2	2
	3	4	3	3	2
	2	4	3	3	3
	1	4	3	3	3
		1	2	3	4
$CM_{s_i=di_3} - \text{palliative}$					

Other scenarios ($s_{15}, s_{18}, s_{19}, s_{20}$) associated with the Availability parameter (Table 12) show also no difference for derived values W_{imp}^s from the array ($CM_{s,j} = di_2 = 1$ dark gray, and $CM_{s,j} = di_2 = 2$ gray).

As can be seen from the Table 11, noticeable differences in values, W_{imp}^s appear for $CM_{s,j} = di_2 > 2$. In a situation where there are no differences in W_{imp}^s values for the weighted value, $CM_{s,j} = di_2 = 1$ and the calculated $CM_{s,j} = di_2 = 2$, the value $W_{imp}^{s,a}$ and $W^{s,a}$ determined from the $M_{imp}^{s,a}$ and $M^{s,a}$ arrays, will be the same for the identified and classified assets, etc.

Taking into account the derived results, it can be unambiguously stated that by considering the recovery measures of resulting losses from threats, an effect in differences of risk values between the FoMRA a CRAMM methods is noticed. The rationale for the resulting difference is coming out from a different structure of both methods. According to the literature [32], CRAMM and MEHARI methods are designed to analyze active risk (preventive measures are planned as a reaction to the possible risks before they occur). The proposed FoMRA, which is partially based on the

MEHARI method (with the same requirements for $W^{s,a}$ assessment [13]) includes recovery measures, which leads to statement that both, FoMRA and MEHARI method are dedicated for partial analysis of reactive risk (some preventive measures are applied "post factum", after the occurrence and identification of a risk and as a reaction to it).

4 Summary

A comparative analysis of a new model against the well-known and widely used methods of risk assessment was discussed. The obtained experimental results confirm the correctness of theoretical assumptions of the FoMRA model. Comparative analysis of the model gave almost identical results of risk values, assuming lack of the recovery measures in the FoMRA, oppositely to CRAMM methods. Considering CRAMM as the most well-known, accepted and used method for risk assessment in various IT systems (source materials for establishment ISO/IEC 27002 standard [31]), it can be concluded that the proposed FoMRA is meaningful, it is not difficult and laborious and can describe really well any information system, as it was shown on the example of an administrative unit operating in Poland. It can also be adapted to any organization. Further research is primarily focused on the FoMRA development towards its adaptation to any method, not only CRAMM or MEHARI. Another, equally important issue is the possibility of avoiding cost and time-consuming analyzes in the FoMRA, which must be performed after the introduction of any changes in the system.

References

1. Datta, A.: Information Technology Capability, Knowledge Assets and Firm Innovation: A Theoretical Framework for Conceptualizing the Role of Information Technology in Firm Innovation. *International Journal of Strategic Information Technology and Applications* 2, 9–26 (2011)
2. Raduan, C.R., Jegak, U., Haslinda, A., Alimin, I.I.: A Conceptual Framework of the Relationship Between Organizational Resources, Capabilities, Systems, Competitive Advantage and Performance. *Research Journal of International Studies* 12, 45–58 (2009)
3. Van Kleef, J.A.G., Roome, N.J.: Developing capabilities and competence for sustainable business management as innovation: a research agenda. *Journal of Cleaner Production* 15, 38–51 (2007)
4. Bhatnagar, A., Ghose, S.: Segmenting consumers based on the benefits and risks of Internet shopping. *Journal of Business Research* 57, 1352–1360 (2004)
5. Byeong-Joon, M.: Consumer adoption of the internet as an information search and product purchase channel: some research hypotheses. *Int. J. Internet Marketing and Advertising* 1, 104–118 (2004)
6. Bumsuk, J., Ingoo, H., Sangjae, L.: Security threats to Internet: a Korean multi-industry investigation. *Information & Management* 38, 487–498 (2001)
7. Posthumus, S., Solms, R.: A framework for the governance of information security. *Computers & Security* 23, 638–646 (2004)
8. Baker, W.H., Wallace, L.: Is Information Security Under Control?: Investigating Quality in Information Security Management. *IEEE Security & Privacy* 5, 36–44 (2007)
9. Yeh, Q.-J., Chang, A.J.-T.: Threats and countermeasures for information system security: A cross-industry study. *Information & Management* 44, 480–491 (2007)

10. Ezingear, J.N., Bowen, S.M.: Triggers of change in information security management practices. *Journal of General Management* 32, 53–72 (2007)
11. Whitman, M.E., Mattord, H.: *Principles of Information Security*, 3rd edn. Course technology, Boston (2009)
12. Mellado, D., Blanco, C., Sánchez, L.E., Medina, E.F.: A systematic review of security requirements engineering. *Computer Standards & Interfaces* 32, 153–165 (2010)
13. El Fray, I., Kurkowski, M., Pejas, J., Mackow, W.: A New Mathematical Model for Analytical Risk Assessment and Prediction in IT Systems. *Control and Cybernetics* 41, 1–28 (2012)
14. Mayer, N., Humbert, J.P.: *La gestion des risques pour les systèmes d'information*. MISC-Éditions Diamond 24, 1–7 (2006)
15. Consultative Objective and Bi-functional Risk Analysis (COBRA): C&A Security Risk Analysis Group, UK (1991)
16. Control Objectives for Information and related Technology (COBIT). Information Systems Audit and Control Association, US (2007)
17. Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE): Carnegie Mellon University, US (2006)
18. Risk Analysis and Management Method (CRAMM): Central Computing and Telecommunications Agency, United Kingdom (1987)
19. Méthode Harmonisée d'Analyse de Risques (MEHARI): Club de la Sécurité de l'Information Français, France (2010)
20. Moeller, R.: *IT Audit, Control, and Security*. John Wiley & Sons, Inc., Hoboken (2010)
21. Guideline for Automatic Data Processing Risk Analysis: Federal Information Processing Standard - FIPS 65. National Bureau of Standard, US (1997)
22. Dray, J.: Computer Security and Crime: Implications for Policy and Action. *Information Technology & People* 4, 297–313 (1988)
23. Fisher, T.: ROI in social media: A look at the arguments. *Journal of Database Marketing & Customer Strategy Management* 16, 189–195 (2009)
24. Parker, D.B.: *Computer Security Management*. Reston Publishing Co., Reston (1991)
25. Rainer, R.K., Snyder, C.A., Carr, H.H.: Risk Analysis for Information Technology. *Journal of Management Information Systems Archive* 8, 129–147 (1991)
26. Ferdous, R., Khan, F.I., Veitch, B., Amyotte, P.R.: Methodology for Computer-Aided Fault Tree Analysis. *Process Safety and Environmental Protection* 85, 70–80 (2007)
27. Andrews, J.D., Ridley, L.M.: Application of the cause-consequence diagram method to static systems. *Reliability Engineering & System Safety* 75, 47–58 (2002)
28. Bartlett, M., Hurdle, E.E., Kelly, E.M.: Integrated system fault diagnostics utilising digraph and fault tree-based approaches. *Reliability Engineering & System Safety* 94, 1107–1115 (2009)
29. Jacoub, S.M., Ammar, H.H.: A methodology for architectural-level reliability risk analysis. *IEEE Transaction on Software Engineering* 28, 529–547 (2002)
30. Technical manual - Reliability/availability of electrical & mechanical systems for command, control, communications, computer, intelligence, surveillance and reconnaissance. Department of the U.S. Army, US (2007)
31. Information technology – Security techniques – Code of practice for information security management. ISO/IEC 27002 (2007)
32. Inventory of Risk Management/Risk Assessment Methods. European Network and information Security Agency (March 2012), http://rm-inv.enisa.europa.eu/methods_tools
33. Braun, G.: *Information Security Risk Analysis and Decision Modelling*. BWI-paper Vrije Universiteit De Boelelaan HV Amsterdam, pp. 1–27 (2002)
34. Expression des Besoins et Identification des Objectifs de Sécurité (EBIOS): Direction Centrale de la Sécurité des Systèmes d'Information, France (2010)

Author Index

- Abdelkrim, Mohamed Naceur 395
Adamec, Ondrej 236
Albarelli, Andrea 168
Augustynek, Martin 236
- Babskova, Alisa 101
Bagchi, Aditya 111
Ben-Abdallah, Hanène 224
Bergamasco, Filippo 168
Bilal, Muhammad 68
Bodei, Chiara 1
Boucetta, Rahma 395
- Chaki, Nabendu 80, 266, 351
Chaki, Rituparna 361
Chakraborty, Manali 351
Chattopadhyay, Matangini 314
Chattopadhyay, Samiran 314
Chmielewski, Andrzej 242
Cho, Young Im 30
Cortesi, Agostino 266
Cyganek, Bogusław 180
- Dasgupta, Subhasis 111
Deb, Novarun 80
Degano, Pierpaolo 1
Doroz, Rafal 158
Dráždilová, Pavla 101, 278
Dutta, Anjan 266
Dvorský, Jiří 405
Dzieńkowski, Bartłomiej Józef 290
- El Fray, Imed 428
- Fedorčák, Dušan 416
Ferrari, Gian-Luigi 1
- Gajdoš, Petr 92
Galletta, Letterio 1
- Hammami, Mohamed 224
Homenda, Wladyslaw 326, 338, 382
Homsirikamol, Ekawat 56
Horák, Zdeněk 302
Hyla, Tomasz 41
- Janoška, Zbyněk 405
Jastrzebska, Agnieszka 382
- Klimes, Petr 215
Kocyan, Tomáš 278
Korpas, David 215
Kozera, Ryszard 146
Krömer, Pavel 302
Kuciapski, Michal 374
Kudělka, Miloš 302
- Lahiri, Ayan 314
Lucińska, Małgorzata 254
- Marcinkowski, Bartosz 374
Markowska-Kaczmar, Urszula 290
Martinovič, Jan 101, 278, 416
Masood, Rahat 68
Mezzetti, Gianluca 1
Micanik, David 236
Minks, Štěpán 101
Miształ, Krzysztof 135
Mliki, Hazar 224
Moravec, Pavel 92
Morawiecki, Paweł 56
Mukherjee, Saswati 314
- Orczyk, Tomasz 191
- Panasiuk, Piotr 203
Pejaś, Jerzy 41
Penhaker, Marek 215
Pindor, Jakub 215
Platoš, Jan 302
Poniszewska-Maranda, Aneta 123
Porwik, Piotr 191
- Radecký, Michal 416
Rogawski, Marcin 56
Rogowski, Marcin 203
Rybník, Mariusz 203, 326
- Saeed, Khalid 135, 203
Saha, Soumyabrata 361
Sen, Soumya 266
Shibli, Muhammad Awais 68
Sitarek, Tomasz 338

Slaninová, Kateřina 101, 278

Snášel, Václav 278, 302

Srebrny, Marian 56

Tabor, Jacek 135

Tchórzewski, Jacek 146

Tomis, Radek 416

Torsello, Andrea 168

Um-e-Ghazia 68

Vojáček, Lukáš 405

Vondrák, Ivo 416

Wierzchoń, Sławomir T. 242, 254

Wrobel, Krzysztof 158