Advances in Intelligent Systems and Computing 578

Marek Kurzynski
Michal Wozniak
Robert Burduk *Editors*

# Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017

Springer

# Advances in Intelligent Systems and Computing

Volume 578

**Series editor**

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

## *About this Series*

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within “Advances in Intelligent Systems and Computing” are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

## *Advisory Board*

More information about this series at http://www.springer.com/series/11156

Marek Kurzynski · Michal Wozniak
Robert Burduk
Editors

# Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017

Springer

*Editors*

Marek Kurzynski
Department of Systems and Computer Networks
Wrocław University of Technology
Wroclaw
Poland

Robert Burduk
Department of Systems and Computer Networks
Wrocław University of Technology
Wroclaw
Poland

Michal Wozniak
Department of Systems and Computer Networks
Wrocław University of Technology
Wroclaw
Poland

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-59161-2 ISBN 978-3-319-59162-9 (eBook)
DOI 10.1007/978-3-319-59162-9

Library of Congress Control Number: 2017940243

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

The goal of the CORES series of conferences is the development of theories, algorithms, and applications of pattern recognition and machine learning methods. These conferences have always served as useful forum where researchers, practitioners, and students working in different areas of pattern recognition can meet to come together and help each other keeping up with this active field of research. This book is a collection of 52 carefully selected works which have been reviewed by the experts from the domain and accepted for presentation during the 10th International Conference on Computer Recognition Systems CORES 2017. We hope that the book can become the valuable source of information on contemporary research trends and most popular areas of application.

Editors would like to express their deep thanks to authors for their valuable submissions and all reviewers for their hard work, and we believe that this book could be a reference tool for scientists who deal with the problems of designing computer pattern recognition systems.

This year we are celebrating the 85th Anniversary of Prof. Juliusz L. Kulikowski from Nalecz Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Sciences, who will be also plenary speaker during the conference. We will have also possibility to attend two outstanding keynote speeches by Prof. Katarzyna Stąpor from Silesian University of Technology, Poland, and Dr. Bartosz Krawczyk from Virginia Commonwealth University, USA.

Although the last, not least, we would like to give special thanks to local organizing team (Kondrad Jackowski, Dariusz Jankowski, Maciej Krysmann, Paweł Trajdos, Andrzej Żołnierek) who did a great job.

We would like also to fully acknowledge support from the Wrocław University of Technology, especially Prof. Andrzej Kasprzak—Chair of Department of Systems and Computer Networks and Prof. Czesław Smutnicki—Dean of Faculty of Electronics which have also supported this event.

We believe that this book could be a great reference tool for scientists who deal with the problems of designing computer pattern recognition systems.

May 2017

Robert Burduk
Marek Kurzynski
Michal Wozniak

# Memories of Professor Co-workers

**Juliusz L. Kulikowski** received MSc. degree in electronic engineering from the Warsaw Technical University in 1955, CandSc. degree from the Moscow Higher School of Technology in 1959, and DSc. degree from the Warsaw Technical University in 1966. Since 1966, he was a scientific worker in several Institutes of the Polish Academy of Sciences. He was a nominated professor in 1973. Since 1981, he is employed in the Institute of Biocybernetics and Biomedical Engineering PAS in Warsaw. He published about 300 papers in information science, signals detection in noise, image processing methods, artificial intelligence, application of computers in medicine as well as 8 books and monographs in these domains. For many years, he was the Editor in Chief of a scientific quarterly "Computer Graphics & Vision", a member of IFIP TC13 on "Human-Computer Interaction" and of IFAC TC on "Stochastic Systems", and a Chairman of the Polish National Committee for cooperation with the Committee of Data for Science and Technology CODATA. He is an ordinary member of the Warsaw Scientific Society.

Professor Juliusz Lech Kulikowski appeared in our life when he started working at the Institute of Automation in 1966 as an associate professor—specialist in frequency modulation technology. He was appointed the head of a department at the institute. Then, we were not yet his employees. He immediately gained a reputation for modest, hardworking, and demanding person. Our first contact with him was participating in a doctoral seminar, which he led next to managing of his team.

We were fascinated by his depth knowledge and at the same time a clear presentation of statistical methods in both theoretical and practical terms. In 1973, he received the title of associate professor, and in 1989 the title of professor. At that time, a team headed by Professor had his first success in the form of Award Scientific Secretary of the Polish Academy of Sciences in the field of digital image processing. Then, the first Polish system of computer image analysis CPO-1/Odra 1204 was developed.

In 1976, Professor left the Institute and started working as a director of the Computer Science Committee chaired by Prime Minister Piotr Jaroszewicz. At that time, under the direction of Professor Kulikowski, "Program for the development of public IT services for the years 1978–1980" was developed and approved by the Minister of Science, Technology, and Higher Education Prof. Sylwester Kaliski. The program included the development of government information systems, manufacturing hardware, and the establishment of a system of government SINTO (System for Scientific Technical and Organizational Information), which was supposed to improve the circulation of scientific information in the country, integrating functional activities of centers of scientific and technical information in the ministries, the unions and the workplaces as well as the activities of archives and libraries.

Then, watching the activity of Professor at such a high state level while he was combining this activity with work in the Institute of Computer Science Polish Academy of Science, we did not think he would ever come back to us (to our Institute). This happened in 1981 when the Institute of Biocybernetics and Biomedical Engineering Polish Academy of Science was established. Professor Kulikowski was appointed the head of the Department of Information Processing Method, where his former and new employees were found. The team employed in the Department counted over 20 people. In the eighties and early nineties, next system for image analysis CPO-1-3, minicomputer system CPO/M6810 with television camera, and finally improved systems of computer vision VIST and SUPERVIST were created.

In 1984, the Institute moved from its headquarter at Twarda street to newly built buildings at Ks. Trojden street. There were new topics of work and new achievements: the design of DIAVENT system—a computer system for analysis of ultrasound images of the heart, based on the original mathematical model of ventricular contractility (1996); the design of an electronic field orientation for the blind ESOT (1999); and the design of computer diagnosis of urine bladder cancer. After started working at the Institute of Biocybernetics and Biomedical Engineering, Professor intensified his scientific activity in theoretical issues, such as the use of discriminant analysis and nonparametric tests for the detection of signals, extended relational algebra, the measure of the quality of information, tripartite graphs, topological logic (relative), and morphological spectra. He worked also on the concept of hyper-relation as a generalization relationship, which is useful to describe and recognize complex images, the use of deontic logic, and the concept of recognizing contents of the images based on the ontology of domain describing their structure using a system of interconnected relationships.

All the time he has been writing and publishing articles, textbooks, and monographs and also acted as an editor. He was always very conscientious, thorough, and well prepared.

He collaborated with the editorial offices, lectured at home and abroad, promoted 20 doctors (including many among us). He was active in the national and international organizations (CODATA, IFIP TC-13 Human-Computer Interaction), in a number of scientific councils of institutes, reviewed doctoral dissertations, postdoctoral research projects, articles in scientific journals . He was always up to date with publications in his areas of scientific interest for what allowed him knowledge of foreign languages (Russian, German, English, and French). In the seventies, he was repeatedly invited as a visiting professor for lectures to the Dipartimento di Informatica Università Degli Studi Di Udine and later to the TU Otto von Guericke University in Magdeburg. It was, among others, in recognition of his high position in the scientific field of algebraic methods of recognition.

He was always willing to help with scientific support, add courage, and serve good advice. His kindness, serenity, and a great sense of humor cause that every day we gather with pleasure at his secretariat even for a moment to feel the friendly atmosphere. He cared diligently for our scientific development, but he did it subtly and without any stress for us. We gratefully and respectfully still admire his diligence, modesty, tolerance, honesty, and his serenity, great sense of humor and common sense, and distance to matters less important. He is always cheerful and with a visible need of bringing joy to others.

Privately, a lover of literature and music, a talented poet.

Annamonika Dulewicz

# Contents

# Recognition of Fuzzy or Incompletely Described Objects

Juliusz L. Kulikowski(✉)

Nalecz Institute of Biocybernetics and Biomedical Engineering,
Polish Academy of Sciences, 4 Ks. Trojdena Street, 02-109 Warsaw, Poland
jkulikowski@ibib.waw.pl

**Abstract.** Typical pattern recognition problem consists in assigning of a given object (result of observation) to one of previously defined similarity classes of objects. The problem has an unique solution if the classes are disjoint; otherwise it may happen that the considered object can be assigned to a class only on a limited certainty level. A more general problem arises if the object being to be recognized has not been described with a full accuracy. The situations of uncertainty consisting in missing some components of objects description and in inaccuracy of some objects' features or parameters description are considered. An approach to the solution of the ill-described objects recognition based on the concepts of relative logic is proposed. This makes the proposed approach closer to a natural human decision making supported by intuition and, as such, useful in the case of uncertainty concerning the input data of the recognition problem.

**Keywords:** Pattern recognition · Limited input data · Input data uncertainty · Information variables · Relative logic · Semi-ordering

## 1 Introduction

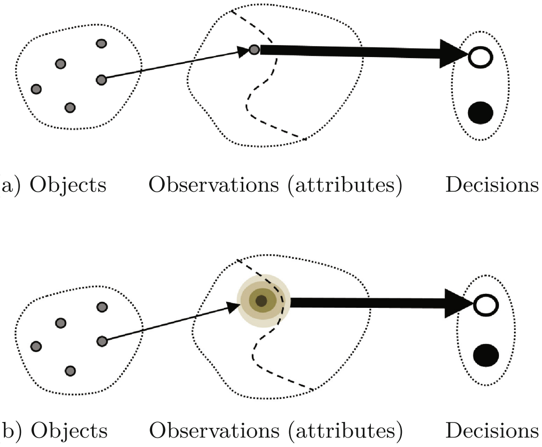
Classical pattern recognition problem is an idealized theoretical model of decision making consisting in assigning to an observed object the name of a class of objects (a pattern) satisfying a predefined criterion of similarity [1,2]. The notion of an *object* may denote in this case a physical or graphical object, an acoustic signal or linguistic expression, a scene, a result of observation etc. Such formulation of the problem admits not only some uncertainty about the membership of the object to a similarity class but also a primary ignorance about the exact borders of the similarity classes. The last can be described analytically or geometrically (this being an exceptional situation), by probability distributions, membership functions or as some environments of enumerable sets of exemplary representatives of the classes. On the other hand, the objects to be recognized are represented by strongly defined attributes, formally represented by points in observation space: vectors of measured parameters or strings of qualitative features.

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_1

However, such pattern recognition models don't correspond well to the situations when the objects being to be recognized are ill-determined. Such situations arise, e.g., in the case of personal identification based on fingerprint, iris or face analysis, etc. if input data have not been sufficiently carefully recorded [3–5]. The widely known pattern recognition methods [6,7] don't suit well to the recognition of ill-defined, fuzzy or incompletely described objects. For example, the $k$-NN method [8] cannot directly be used to the recognition of an object represented by a vector with missing components. Therefore, the problem of *ill-described objects recognition* (I-DOR) as an extension of the typical *pattern recognition* (PR) problems deserves a deeper investigation. A crucial difference between the PR and I-DOR problems consists in representation of the recognized objects not by strongly determined but by *information variables* charged by uncertainty, as illustrated in Fig. 1a, b. The notion of information variables, originally introduced in [9], will be reminded in Sect. 2. It, in particular, suits well to the following I-DOR problems characterization:

(a) *partial visibility or occlusion*: a part of a visual object is invisible or occluded by another visual object;
(b) *fuzziness*: the form of image detail to be recognized has not been exactly described;
(c) *missing attributes*: some object's attributes (features or parameters) have not been observed or measured;
(d) *inexactness*: some object's attributes have not been exactly observed or measured;
(e) *improperness*: some object's attributes have been observed or measured by improper methods, tendentiously changed or became outdated, etc.

Some of the above-mentioned situations in the literature have been described. In particular, the Hough transform is a well know method of reconstruction



(a) Objects Observations (attributes) Decisions

(b) Objects Observations (attributes) Decisions

**Fig. 1.** Difference between (a) the PR and (b) the I-DOR problems.

of partially visible simple geometrical image details [10]. Decision making in the case of missing data have been analyzed in a wider context of data mining methods [11]. However, the I-DOR problems have some specificity that should be taken into consideration when attempts to solve them by methods adopted from other areas are made. The consecutive sections of this paper are devoted to introduction of basic notions (Sect. 2) and to the methods of solution of the above-mentioned I-DOR problems (Sects. 3, 4 and 5). In the last section some conclusions are given.

## 2 Basic Notions

### 2.1 General Notions

Along this paper the following notions will be used. A set of all possible *objects* $\omega$ under consideration constitutes an *object space* denoted by $\Omega$. The objects may be of any fixed physical or abstract nature, however, they are assumed to be mutually distinguishable (for this purpose the symbols $\omega$ will be indexed). The objects are characterized by their observed *attributes* which may be of numerical (real, integer etc.), Boolean or any qualitative nature. The *attribute space* $U$ will thus be defined as a Cartesian product of the sets $U^{(i)}$ of the values of simple attributes:

$$U = \times_{[i]} U^{(i)} \ . \tag{1}$$

It is assumed that in the attribute space $U$ a *similarity* (reflexive, symmetrical and transitive) *relation* $\sigma$ holds and it generates a partition of $U$ into a family $C = \left\{C^{(\kappa)}\right\}, \kappa \in K$, of subsets called *similarity classes*, satisfying the conditions:

(a) $\forall \kappa, \lambda, \kappa \neq \lambda,\ C^{(\kappa)} \cap C^{(\lambda)} = \varnothing$,
(b) $\bigcup_{\kappa \in K} C^{(\kappa)} = U$ .

The set $K$ of tags assigned to the similarity classes is assumed to be finite (at least two-element) and its elements are called *patterns*. Moreover, it is taken into consideration a family

$$F = 2^U \tag{2}$$

of subsets of the attribute space, called an *observation space*. We use the symbol $v$ (indexed, when necessary) for denoting the elements of $F$ (the *observations*). Remark that the observations satisfy the set algebra rules including the operations of sum ($\cup$), intersection ($\cap$) and asymmetrical difference ($\setminus$) of sets [12].

### 2.2 Information Variables

Let us take into consideration an observation space $F$. First, let us remind two basic definitions:

**Definition 1.** A binary: (a) reflexive, (b) symmetrical and (c) transitive relation $\approx$ described in a non-empty set $A$ is called *equivalence* of the elements of $A$.

Equality of the elements (observations) of $F$ is a simple example of their equivalence.

**Definition 2.** A binary relation $\preccurlyeq$ satisfying the conditions of: (a) reflexivity, (b) weak asymmetry ($\alpha' \preccurlyeq \alpha'' \wedge \alpha'' \preccurlyeq \alpha'$ holds if and only if $\alpha' \approx \alpha''$) and (c) transitivity is called a *semi-ordering* of the elements of $A$.

The expression $\alpha' \preccurlyeq \alpha''$ should be read as: $\alpha'$ *precedes* $\alpha''$ or $\alpha'$ *is followed by* $\alpha''$. The well-known notion of a *subset* $\subseteq$ (in the set theory sense [12]) introduces, in particular, a semi-ordering of the elements of any family $B$ of sets. Really, for any sets $\beta', \beta'' \in B$ if $\beta' \subseteq \beta''$ then it also can be put $\beta' \preccurlyeq \beta''$. We call a *natural semi-ordering* of the elements of $B$ the semi-ordering following from the notion of subset. This, in particular, can be applied to the observations in $F$. Let us remark that Definition 2 admits existence in $A$ some pairs of *incomparable* elements, for which is neither $\alpha' \preccurlyeq \alpha''$ nor $\alpha'' \preccurlyeq \alpha'$. Natural semi-ordering satisfies the following *mini-maximization* properties: for any $\beta', \beta'' \in B$ if $\beta' \preccurlyeq \beta''$ then

$$(\beta' \cup \beta'') \approx \beta'', \tag{3}$$

$$(\beta' \cap \beta'') \approx \beta' \ . \tag{4}$$

This property, when necessary, can be extended on other types of semiordering.

**Definition 3.** If $F$ is an observation space described on the basis of attribute space $U$ then we call *information variable* a pair:

$$Z_U = [F, \precsim] \tag{5}$$

where $\precsim$ is a semi-ordering relation described in $F$ satisfying the following conditions:

(a) it contains a natural semi-ordering of $F$ as its sub-relation,
(b) it satisfies the mini-maximization properties.

The symbol $\precsim$ can be read as "*is not more certain that*" while $\approx$ in this context can be interpreted as "*is equally certain as*".

**Definition 4.** Let $S = \{v^{(1)}, v^{(2)}, \ldots, v^{(k)}\}$ be a sub-family of observations of an information variable $Z_U$. Any observation $v^{(i)} \in S$ such that no other observation $v^{(n)} \in S$, if not equivalent to $v^{(i)}$, satisfies the condition:

(a) $v^{(i)} \precsim v^{(n)}$, is called to be of *maximal certainty in* $S$,
(b) $v^{(n)} \precsim v^{(i)}$, is called to be of *minimal certainty in* $S$.

It follows from Definition 4 that: (a) $U$ is of maximal certainty in $F$, (b) empty observation $\theta$ is of minimal certainty in $F$. It also follows from Definition 3 that if $F$ is a family of all measurable subsets of $U$ and $\mu$ is a probability measure described on $U$ then $\mu$ generates a semi-ordering in $F$ according to the principle:

$$v^{(i)} \precsim v^{(j)} \Longleftrightarrow \mu\left(v^{(i)}\right) \leq \mu\left(v^{(j)}\right) \ . \tag{6}$$

In similar way, it can be shown that observations described by fuzzy sets (with a membership function generating the semi-ordering relation [13]) or by rough sets (where only three classes of uncertainty are taken into consideration [14]) can also be considered as particular cases of information variables [9].

### 2.3 Relative Logic

The relation of semi-ordering used above to define the information variables is also a basis of logical systems called *topological* [15] or *relative* [16], which quantitative logical values of statements replace by their relative logical assessment. The correspondence between topological logic and other types of logical systems as well as a concept of *logical space* were described in [9]. Below, for our purposes, the concept of relative logic will be presented in a slightly simplified form.

**Definition 5.** An ordered string of elements:

$$\Lambda = [\Sigma, S, \Sigma^*, \approx, \precapprox] \tag{7}$$

where:

- $\Sigma$ denotes a set of simple, linguistically correct, assertive sentences (statements),
- $S$ is a set of logical operators ($\vee$, $\wedge$, $\neg$, $\Rightarrow$),
- $\Sigma^*$ is a set of composite sentences based on $\Sigma$ and $S$,
- $\approx$ is a relation of logical equivalence and
- $\precapprox$ is a relation of weak logical semi-ordering described in $\Sigma^*$, and satisfying the mini-maximization properties: $\forall q', q'' \in \Sigma^*$ such that $q' \precapprox q''$ it is

$$(q' \vee q'') \approx q'', \tag{8}$$

$$(q' \wedge q'') \approx q' \tag{9}$$

is called a *relative logical system* (RLS).

The symbol $\approx$ in this context can be read as "*is (logically) equivalent to*", while the symbol $\precapprox$ as "*is (logically) not more true than*". Let us remark that in RLS the concepts of "*absolute false*" or "*absolute true*" do not exist.

*Example 1.* Let X be an information variable described on a parameter space $U = [35 \div 42]$. We take into consideration the following statements concerning observations of $X$:

$$q^{(1)} : \text{“}x \textit{ is near } 36.6°C\text{”}, q^{(2)} : \text{“}x \textit{ is higher than } 36°C\text{”},$$
$$q^{(3)} : \text{“}x \textit{ is less than } 35.5°C\text{”},$$

Then the following logical assessment of simple and composite statements in RLS are possible:

$$q^{(1)} \precapprox q^{(2)}, q^{(1)} \approx q^{(2)}, \left(q^{(1)} \vee q^{(2)}\right) \approx q^{(2)}, \left(q^{(1)} \wedge q^{(2)}\right) \approx q^{(1)},$$

but also it may happen: $q^{(3)} \precapprox q^{(2)}$, $\left(q^{(2)} \wedge q^{(3)}\right) \approx q^{(2)}$, etc. On the other hand, it may happen that $q^{(2)}$ and $q^{(3)}$ occur mutually incomparable; such situation can be denoted by $q^{(2)}$ ? $q^{(3)}$.

### 2.4 I-DOR Problems Formulation

If $Z_U$ is an *information variable* described on the basis of the attribute space $U$ then on the basis of the notions introduced in Sects. 2.1, 2.2 and 2.3 two basic projections can be defined:

- *object description*: $\Delta : \Omega \rightarrow Z_U$,
- *object recognition*: $\Phi : Z_U \rightarrow K$.

Various types of I-DOR problems follow from different types of information variables characterizing the observations. Moreover, logical validation of the decisions specifying the object description can be based on RLS concepts.

## 3 Recognition of Objects with Missing Attributes

Missing attribute takes place if information variable $Z_U$ describing a considered object contains components charged with maximal uncertainty. This means that the family $F$ of observations of the corresponding attribute are initially semi-ordered only in a natural way. On the other hand, an object recognition (calculating $\Phi$) algorithm may need complete input data to be entered (an alternative situation is here not considered). The problem of missing attributes' values handling in decision making is widely described in the literature [11] and some of the proposed solution methods to I-DOR problems can be applied. Roughly speaking, the methods consist in missing data replacement by their assumed values. However, the problem is, what values deserve to be "assumed". This can be illustrated on the case of PR based on the *Nearest Neighbor* method [7,8]. The situation is illustrated in Fig. 2: two classes of objects described by pairs of data values $(x_1, x_2)$ are presented by white and black dots. A new object is described by an information value $\xi = (\xi_1, \xi_2)$ whose component $\xi_1$ is given, while $\xi_2$ is missing. Therefore, $\xi$ is presented by a dotted line containing all possible values of the component $\xi_2$. In a more general case the number of similarity classes (patterns) to be recognized may be higher than two. Let $C^{(1)}$, $C^{(2)}$, ..., $C^{(K)}$ denote some given sets of training objects representing the similarity classes. If they are represented by vectors in a $n$-dimensional parameter space $U$ then $\xi$, whose one component (say, $\xi_2$) is missing, will be represented by a line parallel to the $x_2$-axis in $n$-dimensional space. Three possible approaches to fill the missing component can be mentioned:

- Choose the $\xi_2$ value on $\xi$ as close as possible to the nearest to $\xi$ training object $o$; include $\xi$ to the similarity class $C^*$ to which $o$ belongs.
- Put $\xi_2$ equal the mean of the $x_2$ components of all training objects representing the given similarity classes $C^{(1)}$, $C^{(2)}$, ..., $C^{(K)}$.
- Put $\xi_2$ maximizing the object-class similarity $\sigma(\xi, C^*)$ among all values of $\xi_2 \in \xi$ and all $C^{(K)}$.

First approach is simply to be realized, however it may fail if more than one training object are the closest to $\xi$. Second approach is also simply to realization,

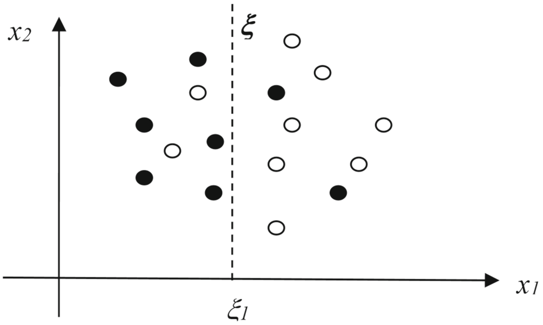


**Fig. 2.** Missing component $\xi_2$ in an I-DOR problem.

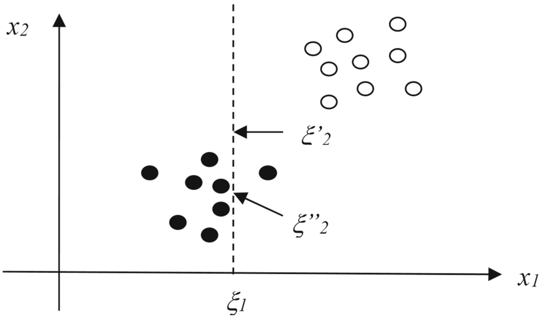


**Fig. 3.** Two, in different way chosen values of missing data component.

however, in some cases it leads to incorrect decisions. Such situation is illustrated in Fig. 3. The value $\xi_2'$, chosen as the mean of $x_2$-components of all objects seems to be not well adjusted to the given situation. On the other hand, the value $\xi_2''$, chosen as the closer one to the class of black objects, more likely characterizes the missing $\xi_2$ component.

## 4 Recognition of Objects Described with Interval-Type Uncertainty

A more general case of uncertainty corresponds to a situation when datacomponents take values within a cuboid $\Delta$ described by the intervals:

$$\xi_i \in [\xi_{i,\min}, \xi_{i,\max}], i = 1, 2, \ldots, I.$$

Once again, the information variable describing the object is of maximal uncertainty within $\Delta$. Two types of I-DOR problems then will be taken in to consideration:

- Similarity classes in objects' parameter space $U$ are exactly given,
- Similarity classes are represented by sets of training data.

The first situation is illustrated in Fig. 4. Primarily, it can be assumed that a central point of $\Delta$ is the most representative and, consequently, the information variable $\xi$ describing the object should belong to the $II$ similarity class. However, taking into account that all data-values within $\Delta$ are equally admissible, it is also possible to admit that $\xi$ should be assigned to the similarity class $C^*$ that covers the largest part of $\Delta$. In this case $\Delta$ should be replaced by a centroid 0 of the intersection $\Delta \cap C^*$. The case of similarity classes represented by training objects needs a slightly modified approach, based on a general concept of similarity measure [1,2] used to define a *measure of coherence* $\eta$ between an observation $u$ and a set $\Delta$ of observations.

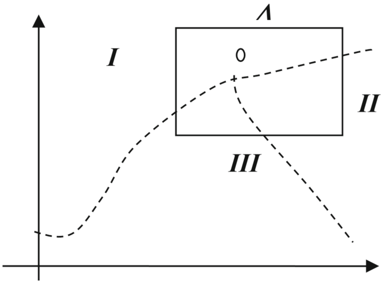


**Fig. 4.** Interval data-value choosing among similarity classes.

**Definition 6.** Let $U$ be an attribute space in which a similarity measure $\sigma$ has been defined, let $u \in U$ be an observation and let $\Delta \subseteq U$ be a subset of observations. We call *measure of coherence* between $u$ and $\Delta$ the parameter

$$\eta\,(u, \Delta) = \max_{(\xi \in \Delta)} \{\sigma\,(u, \xi)\} \ . \tag{10}$$

Solution of the I-DOR problem in the case of interval uncertainty of object description and similarity classes given by training sets then can be based on a modification of the *k-most-similar-objects* method taking the form:

$\xi \in C^*$ if $C^*$ contains most of $k$ observations $u$ such that $\eta\,(u, \Delta)$ is maximal.

## 5 Recognition of Objects with Fuzzy Attributes

Let us assume that a recognized object is described by information variable given by any type of a membership or probability density function. The similarity classes then correspond to the areas where a certain membership function is dominating over any other membership function. PR based on Bayesian rule

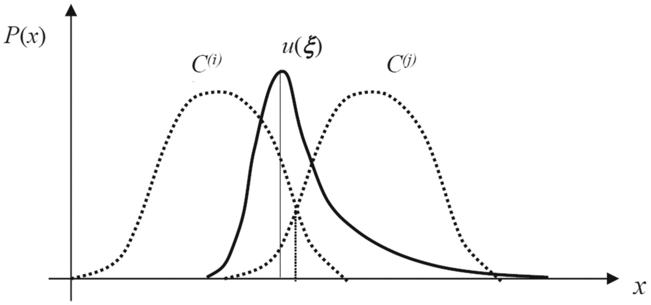


**Fig. 5.** Recognized object described by a membership function $u(\xi)$.

is an example of such situation. Here, the case of an object represented by a membership function, illustrated in Fig. 5, is considered. In this case inclusion of $\xi$ to a similarity class cannot be done by a comparison of the values of the membership functions calculated for $\xi$. Alternative methods may consist in:

- Taking the point $\xi_{\max}$ of maximum of the membership function $u(\xi)$ as a single representative of the considered object,
- Taking the centre of gravity $\xi_{\text{centr}}$ of $u(\xi)$ as a representative of the considered object,
- Including $\xi$ to a similarity class whose domain in higher degree covers the domain of $u(\xi)$,

etc. However, the first two proposals do not take into consideration all information about the form of $u(\xi)$. Third proposal may be ineffective if the domains of the membership functions are extended over all admissible area of the attribute parameter. Instead, the following approach can be proposed.

**Definition 7.** Let $u(\xi)$ and $w(\xi)$ be two described in the parameter space $U$ membership functions. We call their *coincidence factor* $\varphi$ the scalar product:

$$\varphi(u, w) = \int_U u(\xi) \cdot w(\xi)\, d\xi \,. \tag{11}$$

Therefore, in this case the I-DOR problem can be solved according to the rule:

$\xi \in C^*$ if $C^*$ is the class, described by $w^*(\xi)$ for which $\varphi(u, w^*)$ is maximal.

## 6 Conclusions

Typical PR problems assume that the objects being to be recognized are strongly described by the values of their attributes. The I-DOR problems assume that the attributes are available only as instances of some information variables charged by uncertainty. The situations of a total ignorance of a parameter value, of the value being located within an interval or being given by a membership or a probability density function can be described as information variables. The paper tries to show that the concepts based on relative logic provide general tools which make possible description of various types of the I-DOR problems and also suggest some methods of their solution.

## References

1. Kulikowski, J.L., Przytulska, M.: Pattern recognition based on similarity in linear semi-ordered spaces. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011. LNCS (LNAI), vol. 6678, pp. 22–29. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21219-2_3
2. Kulikowski, J.L.: Object recognition based on comparative similarity assessment. In: Burduk, R., Jackowski, K., Kurzyński, M., Woźniak, M., Żołnierek, A. (eds.) Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015. AISC, vol. 403, pp. 243–253. Springer, Cham (2016). doi:10.1007/978-3-319-26227-7_23
3. Abate, A.F., Nappi, M., Riccio, D., Sabatino, G.: 2D and 3D face re cognition: a survey. Pattern Recogn. Lett. **28**(14), 1885–1906 (2007)
4. Choraś, R.S.: Iris recognition. In: Kurzynski, M., Wozniak, M. (eds.) Computer Recognition Systems 3. AISC, vol. 57, pp. 593–600. Springer, Heidelberg (2009). doi:10.1007/978-3-540-93905-4_69
5. Kubanek, M.: Automatic methods for determining the characteristic points in face image. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010. LNCS (LNAI), vol. 6113, pp. 523–530. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13208-7_65
6. Russ, J.C.: The Image Processing Handbook, 6th edn. CRC Press, Boca Raton (2011)
7. Pratt, W.K.: Digital Image Processing. Wiley, New York (2007)
8. Pitas, I.: Digital Image Processing Algorithms and Applications. A Wiley Interscience Publication, New York (2013)
9. Kulikowski J.L. Topological logic and fuzzy sets - a comparison of methods (in Polish). In: Chojcan, J., Łeski, J. (eds.): Zbiory rozmyte i ich zastosowania, pp. 195–516. Wyd. Pol. Śl., Gliwice (2001)
10. Hough, P.V.C.: Machine analysis of buble chamber picture. In: Proceedings of International Conference of High Energy Accelerators and Instrumentation, CERN (1959)
11. Grzymala-Busse, W.J.: Handling missing attribute values. In: Maimon, O., Rokach, L. (eds.) The Data Mining and Knowledge Discovery Handbook, p. 3557 (2005)
12. Rudeanu, S.: Sets and Ordered Structures, pp. 1–10. Bentham Science Publishers, Oak Park (2012)
13. Pedrycz, W.: Fuzzy Control and Fuzzy Systems. Wiley–Interscience, New York (1993)
14. Pawlak, Z.: Rough Sets–Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
15. Hempel, C.G.: A purely topological form of non-Aristotelian logic. J. Symb. Logic **2**(3) (1937)

16. Kulikowski, J.L.: Computer-aided relative assessment of complex effects of decisions. In: S. Hippe, Z., L. Kulikowski, J., Mroczek, T., Wtorek, J. (eds.) Issues and Challenges in Artificial Intelligence. SCI, vol. 559, pp. 101–113. Springer, Cham (2014). doi:10.1007/978-3-319-06883-1_9
17. Kulikowski, J.L.: Rozpoznawanie obrazów (in Polish). In: Chmielewski, L., et al. (ed.) Obrazowanie biomedyczne. Biocybernetyka i Inżynieria Biomedyczna 2000, pp. 193–237. AOW Exit, Warsaw (2003)

# Evaluating and Comparing Classifiers: Review, Some Recommendations and Limitations

Katarzyna Stąpor(✉)

Institute of Computer Science, Silesian Technical University, Gliwice, Poland
katarzyna.stapor@polsl.pl

**Abstract.** Performance evaluation of supervised classification learning method related to its prediction ability on independent data is very important in machine learning. It is also almost unthinkable to carry out any research work without the comparison of the new, proposed classifier with other already existing ones. This paper aims to review the most important aspects of the classifier evaluation process including the choice of evaluating metrics (scores) as well as the statistical comparison of classifiers. Critical view, recommendations and limitations of the reviewed methods are presented. The article provides a quick guide to understand the complexity of the classifier evaluation process and tries to warn the reader about the wrong habits.

**Keywords:** Supervised classification · Classifier evaluation · Performance metrics · Statistical classifier comparison

## 1 Introduction

In a supervised classification problem one aims to learn a classifier from a dataset $U = \{(x^{(1)}, t^{(1)}), \ldots, (x^{(n)}, t^{(n)})\}$ of $n$ labeled data instances and each instance $x^{(i)}$ is characterized by $d$ predictive variables/features, $X = (X_1, \ldots, X_d)$, and a class $T$ to which it belongs. This dataset is obtained from a physical process described by an unknown probability distribution $f(X, T)$. Then, the learned classifier, after *evaluating* its *quality* (usually on test dataset), can be used to classify new samples, i.e. to obtain their unknown class labels. We do not make here a distinction between a *classifier* (being a function that maps an input feature space to a set of class labels) and a *classification learning algorithm* which is a general methodology that can be used, given a specific dataset, to learn a specific classifier. Theoretical background on supervised classification problem as well as the whole description of classifier construction process can be found in many books on machine learning and pattern recognition (see for example [2,8,31,33,34,44,47,49]). Usually, the problem of evaluating a new classifier is tackled by using the *score* that try to summarize the specific conditions of interest. *Classification error* and *accuracy* are widely used scores in the classification problems. In practice, classification error must be *estimated* from all the available samples.

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_2

The *k-fold cross-validation*, for example, is on of the most frequently used such estimation methods. Then, questions are whether such a new, proposed classifier (or enhancement of the existing one) yields an improved score over the competitor classifier (or classifiers) or the state of the art. It is almost impossible now to do any research work without an experimental section where the score of a new classifier is tested and compared with the scores of the existing ones. This last step also requires the *selection of datasets* on which the compared classifiers are learned and evaluated. The purpose of dataset selection step should not be to demonstrate classifiers superiority to another in all cases, but rather to identify its areas of strengths with respect to domain characteristics. This paper is focused only on a supervised classification problem as defined in the beginning. Other types of classification such as *classification from data streams* or *multi-label classification* are not addressed here, since they may impose specific conditions to the calculation of the score (for the most important reference in evaluating (static) data streams, see for example [15]). The whole evaluation process of a classifier should include the following steps [41]:

1. choosing an evaluation metric (i.e. a score) according to the properties of a classifier,
2. deciding the score estimation method to be used,
3. checking whether the assumptions made by (1) and (2) are fulfilled,
4. running the evaluation method and interpret the results with respect to the domain,
5. compare a new classifier with the existing ones selected according to the different criteria, for example problem dependent; this step requires *selection of datasets*.

The main purpose of this paper is to provide the reader with a better understanding about the overall classifier evaluation process. As there is no fixed, concrete recipe for the classifier evaluation procedure, we believe that this paper will facilitate the researcher in the machine learning area to decide which alternative to choose for each specific case. The paper is set up as follows. In Sect. 2 we describe measures of classifier quality while in Sect. 3, a short overview of their estimation methods. Section 4 focuses on statistical methods for classifier quality comparison. Finally, in Sect. 5 we conclude giving some recommendations.

## 2 Measures of Classifier Quality

Usually the problem of evaluating a new classifier (i.e. measuring its quality) is tackled by using the *score* that try to summarize the specific conditions of interest when evaluating a classifier. There may be many scores according to how we aim to quantify classifiers behavior. In this section, we only present some of the most extended scores. Typical scores for measuring the performance of a classifier are *accuracy* and *classification error*, which for a two-class problem can be easily derived from a $2 \times 2$ confusion matrix as that given in Table 1. These scores can be computed as:

$$Acc = (TP + TN)/(TP + FN + TN + FP)$$

$$Err = (FP + FN)/(TP + FN + TN + FP)$$

Sometimes, accuracy and classification error are selected without considering in depth whether it is the most appropriate score to measure the quality of a classifier for the classification problem at hand. When both class labels are relevant and the proportion of data samples for each class is very similar, these scores are a good choice. Unfortunately, equally class proportions are quite rare in real problems. This situation is known as the *imbalance problem* [29,45]. Empirical evidence shows that accuracy and error rate are biased with respect to data imbalance: the use of these scores might produce misleading conclusions since they do not take into account misclassification costs, the results are strongly biased to favor the majority class, and are sensitive to class skews. In some application domains, we may be interested in how our classifier classifies only a part of the data. Examples of such measures are: *True positive rate (Recall or Sensitivity):* TPrate = TP/(TP+FN), *True negative rate (Specificity):* TNrate = TN/(TN + FP), *False positive rate:* FPrate = FP/(TN + FP), *False negative rate:* FNrate = FN/(TP + FN), Precision = TP/(TP + FP). Shortcomings of the accuracy or error rate have motivated search for new measures which aim to obtain a trade-off between the evaluation of the classification ability on both positive and negative data samples. Some straightforward examples of such alternative scores are: the *harmonic mean* between *Recall* and *Precision* values: *F-measure* $= 2 \times \mathrm{TPrate} \times \mathrm{Precision}/(\mathrm{TPrate} + \mathrm{Precision})$, and the *geometric mean* of accuracies measured separately on each class: *G-mean* $= \sqrt{\mathrm{TPrate} \times \mathrm{TNrate}}$ [3]. Harmonic and geometric means are symmetric functions that give the same relevance to both components. There are other proposals that try to enhance one of the two components of the mean. For instance, index of balaned accuracy [18], the *adjusted geometric mean* [1], the *optimized precision OP* from [37] computed as: $\mathrm{OP} = \mathrm{Acc} - (|\mathrm{TNrate} - \mathrm{TPrate}|/(\mathrm{TNrate} + \mathrm{TPrate}))$, and F-score [30]:

$$F\text{-}score = \frac{(\beta^2 + 1)\mathrm{Precision} \times \mathrm{TPrate}}{\beta^2 \times \mathrm{Precision} + \mathrm{TPrate}}$$

A parameter $\beta$ can be tuned to obtain different trade-offs between both components. When a classifier classifies an instance into a wrong class group, a loss is incurred. *Cost-sensitive learning* [10] aims to minimize this loss incurred by the classifier. The above introduced scores use the 0/1 *loss function*, i.e. they treat all the different types of misclassification as equally severe. The *cost matrix* can be used if the severity of misclassifications can be quantified in terms of costs. Unfortunately, in real applications, specific costs are difficult to obtain. In such situations, however, the described above scores may be useful since they may also be used to set more relevance into the costliest misclassification: minimizing the cost may be equivalent to optimal trade-off between *Recall* and *Specificity* [7]. When the classification costs cannot be accessed, another most widely-used techniques for the evaluation of classifiers is the *ROC curve* [4,11], which is a

**Table 1.** Confusion matrix for a two-class problem

| | Predicted positive | Predicted negative |
|---|---|---|
| Positive class | True Positive (TP) | False Negative (FN) |
| Negative class | False Positive (FP) | True Negative (TN) |

graphical representation of Recall versus *FPrate (1-Specificity)*. The information about classification performance in the ROC curve can be summarized into a score known as AUC (Area under the ROC curve) which is more insensitive to skewness in class distribution since it is a trade-off between Recall and Specificity [43]. However, recent studies have shown that AUC is a fundamentally incoherent measure since it treats the costs of misclassification differently for each classifier. This is undesirable because the cost must be a property of the problem, not of the classification method. In [21,22], the *H measure* is proposed as an alternative to AUC. While all of the scores described above in this section are appropriate for two-class imbalanced learning problems, some of them can be modified to accommodate the multi-class imbalanced learning problems [23]. For example [46] extends the G-mean definition to the geometric mean of Recall values of every class. Similarly, in [12] they defined *mean F-measure* for multi-class imbalance problem. The major advantage of this measure is that it is insensitive to class distribution and error costs. However, it is now an open question if such extended scores for multi-class classification problem are appropriate on scenarios where there exist multiple minority and multiple majority classes [40]. In ([20] they proposed the *M measure*, a generalization approach that aggregates all pairs of classes based on the inherent characteristics of the AUC. In this paper, we focus on the scores since they are popular way to measure classification quality. But these measures do not capture all the information about the quality of classification methods some graphical methods may do. However, the use of quantitative measures of quality makes the comparison among the classifiers easier (for more information on graphical methods see for example [9,30,36]). The presented list of scores is by no means exhaustive. The described scores are focused only on the evaluating the performance of a classifier. However, there are other important aspects of classification such as robustness to noise, scalability, stability under data shifts, etc. which are not addressed here.

## 3 Quality Estimation Methods

Various methods are commonly used to estimate classification error and the other described classifier scores (the review of estimation methods can also be found in the mentioned literature on machine learning). *Holdout method* of estimation of classification error divides randomly the available dataset into independent training and testing subsets which are then used for learning and evaluating a classifier. This method gives a pessimistically biased error estimate (calculated as a ratio of misclassified test samples to a size of test subset), moreover it depends on a particular partitioning of a dataset. These limitations are overcome with a family of *resampling methods*: cross validation (random sub-sampling, k-fold cross-validation, leave-one-out) and bootstrap. *Random subsampling* performs $k$ random data splits of the entire dataset into training and testing subsets. For each data split, we retrain a classifier and then estimate error with test samples. The true error estimate is the average of separate errors obtained from $k$ splits. The *k-fold cross-validation* creates a $k$ fold partition of the entire dataset once: Then, for each of k experiments, it uses $(k-1)$ folds for training and a different fold for testing. The classification error is estimated as the average of separate errors obtained from $k$ experiments. It is approximately unbiased, although at the expense of an increase in the variance of the estimate. *Leave-one-out* is the degenerate case of $k$-fold cross-validation where $k$ is chosen as the total number of samples. This results in the unbiased error estimate, but have large variance. In the *bootstrap* estimation, we randomly select with replacement the samples and use this set for training. The remaining samples that were not selected for training are used for testing. We repeat this procedure $k$ times. The error is estimated as the average error on test samples from $k$ procedures. The benefit of this method is its ability to obtain accurate measures of both bias and variance of classification error estimate.

## 4 Statistical Comparison of Classifiers

The comparison of the scores obtained by two or more classifiers in a set of problems is a central task in machine learning, so it is almost impossible to do any research work without an experimental section where the score of a new classifier is tested and compared with the scores of the existing ones. When the differences are very clear (e.g., when the classifier is the best in all the problems considered), the direct comparison of the scores may be enough. But in most situations, a direct comparison may be misleading and not enough to draw sound conclusions. In such situations, the statistical assessment of the scores such as hypothesis testing is required. Statistical tests arise with the aim of giving answers to the above mentioned questions, providing more precise assessments of the obtained scores by analyzing them to decide whether the observed differences between the classifiers are real or random. However, although the statistical tests have been established as a basic part of classifier comparison task, they are not a definitive tool, we have to be aware about their limitations and misuses. The statistical tests for comparing classifiers are usually bound to a specific estimation method of classifier score. Therefore, the selection of a statistical test is also conditioned by this estimation method. For the comparison of two classifiers on one dataset, the situation which is very common in machine learning problems, the *corrected resampled t test* has been suggested in the literature [35]. This test is associated with a repeated estimation method (for example holdout): in $i$-th of the $m$ iterations, a random data partition is conducted and the values for the scores $A_{k1}^{(i)}$ and $A_{k2}^{(i)}$ of compared classifiers $k1$ and $k2$, are obtained. The statistic is:

$$t = \frac{\overline{A}}{\sqrt{\left(\frac{1}{m} + \frac{N_{\text{test}}}{N_{\text{train}}}\right) \cdot \sum_{i=1}^{m} \frac{(A^{(i)} - \overline{A})^2}{m-1}}}$$

where $\overline{A} = \frac{1}{m}\sum_{i=1}^{m} A^{(i)}$, $A^{(i)} = (A_{k1}^{(i)} - A_{k2}^{(i)})$, $N_{\text{test}}$, $N_{\text{train}}$ are the number of samples in the test and train partitions. The second parametric test that can be used in this scenario whose behavior, however, has not been studied as for previous, is the *corrected t test for repeated cross-validation* [3]. These tests assume the data follow the normal distribution which should be first checked using the suitable normality test. A non-parametric alternative for comparing two classifiers that is suggested in the literature is *McNemars test* [26]. For the comparison of two classifiers on multiple datasets the *Wilcoxon signed-ranks test* [26] is widely recommended. It ranks the differences $d_i = A_{k1}^{(i)} - A_{k2}^{(i)}$ between scores of two classifiers $k1$ and $k2$ obtained on $i$-th of $N$ datasets, ignoring the signs. The test statistic of this test is:

$$T = \min(R^+, R^-)$$

where:

$$R^+ = \sum_{d_i > 0} \text{rank}(d_i) + \frac{1}{2}\sum_{d_i = 0} \text{rank}(d_i), R^- = \sum_{d_i < 0} \text{rank}(d_i) + \frac{1}{2}\sum_{d_i = 0} \text{rank}(d_i)$$

are the sums of ranks on which the $k2$ classifier outperforms $k1$, respectively. Ranks $d_i = 0$ are split evenly among the sums. Other test that can be used is the sign test, but it is much weaker than the Wilcoxon signed-ranks test. Comparison among multiple classifiers on multiple datasets arise in machine learning when a new proposed classifier is compared with the state of the art. For this situation, the general recommended methodology is as follows [5,6,16,39,41]. First, we apply an omnibus test to detect if at least one of the classifiers performs different than the others. *Friedman nonparametric test* [14] with *Iman-Davenport extension* [28] is probably the most popular omnibus test. It is a good choice when comparing more than five different classifiers. Let $R_{ij}$ be the rank of the $j$-th of $K$ classifiers on the $i$-th of $N$ data sets and

$$R_j = \frac{1}{N}\sum_{i=1}^{N} R_{ij}$$

is the mean rank of $j$-th classifier. The test compares the mean ranks of the classifiers and is based on the test statistic:

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1) - \chi_F^2} \quad \chi_F^2 = \frac{12N}{K(K+1)}\left[\sum_{j=1}^{K} R_j^2 - \frac{K(K+1)^2}{4}\right]$$

which follows a $F$ distribution with $(K-1)$ and $(K-1)(N-1)$ degrees of freedom. For the comparison of five or less different classifiers, *Friedman aligned* *ranks* [17] or the *Quade* test [25,38] are the more powerful alternatives. Second, if we find such a significant difference, then we apply a pair-wise test with the corresponding post-hoc correction for multiple comparisons. For the described above Friedman test, comparing the $r$-th and $s$-th classifiers is based on the mean ranks and has the form:

$$z = \frac{R_r - R_s}{\sqrt{\frac{K(K+1)}{6N}}}$$

The $z$ value is used to find the corresponding probability from the table of normal distribution, which is then compared with an appropriate significance level $\alpha$. As performing pair-wise comparisons is associated with a set or family of hypotheses, the value of $\alpha$ must be adjusted for controlling the family-wise error [42]. There are multiple proposals in the literature to adjust the significance level $\alpha$: Holm [27], Hochberg [24], Finner [13]. The results of pair-wise comparisons, often, give not disjoint groups of classifiers. In order to identify disjoint, homogenous groups, in [19] they apply special cluster analysis approach. Their method results in dividing $K$ classifiers into groups in such a way that classifiers belonging to the same group do not significantly differ with respect to the chosen distance.

## 5 Recommendations and Conclusions

This paper covers the basic steps of classifier evaluation process, focusing mainly on the evaluation metrics and conditions for their proper usage as well as the statistical comparison of classifiers. The evaluation of classification performance is very important to the construction and selection of classifiers. The vast majority of the published articles use the *accuracy* (or *classification error*) as the score in the classifier evaluation process. But these two scores may be appropriate only when the datasets are balanced and the misclassification costs are the same for false positives and false negatives. In the case of skew datasets, which is rather typical situation, the accuracy/error rate is questionable and other scores such as *Recall*, *Specificity*, *Precision*, *Optimized Precision*, *F-score*, *geometric* or *harmonic means*, $H$ or $M$ measures are more appropriate. The comparison of two classifiers on a single dataset is generally unsafe due to the lack of independence between the obtained score values. Thus, the *corrected* versions of the *resampled t test* or *t test* for *repeated cross-validation* are more appropriate. *McNemars test*, being non-parametric, does not make the assumption about distribution of the scores (like the two previous tests) but it does not directly measure the variability due to the choice of the training set nor the internal randomness of the learning algorithm. When comparing two classifiers on multiple datasets (especially from different sources), the measured scores are hardly commensurable. Therefore, the *Wilcoxon signed-rank test* is more appropriate. Regarding the comparison of multiple classifiers on multiple datasets, if the number of classifiers involved is higher than five, the use of the *Friedman test with Iman and Davenport extension* is recommended. When this number is low, four or five, *Friedman aligned ranks* and the *Quade test* are more useful. If the null hypothesis has been rejected, we should proceed with a *post-hoc test* to check the statistical differences between pairs of classifiers. The last but not least conclusion follows from no free lunch theorem [48] which states that for any two classifiers, there are as many classification problems for which the first classifier performs better than the second as vice versa. Thus, it does not make sense to demonstrate that one classifier is, on average, better than the others. Instead, we should focus our attention on exploring the conditions of the classification problems which make our classifier to perform better or worse than others. We must carefully choose the datasets to be included in the evaluation process to reflect the specific conditions, for example class imbalance, classification cost, dataset size, application domain, etc. In other words, the choice of the datasets should be guided in order to identify specific conditions that make a classifier to perform better than others. Summarizing, this review tries to provide the reader with a better understanding about the overall process of comparison in order to decide which alternative to choose for each specific case. We believe, that this review can improve the way in which researchers and practitioners in machine learning contrast the results achieved in their experimental studies using statistical methods.

## References

1. Batuvita, R., Palade, V.: A new performance measure for class imbalance learning: application to bioinformatics problem. In: Proceedings of 26th International Conference Machine Learning and Applications, pp. 545–550 (2009)
2. Bishop, C.: Pattern Recognition and Machine Learning. Springer, New York (2006)
3. Bouckaert, R.: Estimating replicability of classifier learning experiments. In: Proceedings of the 21st Conference on ICML. AAAI Press (2004)
4. Bradley, P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn. **30**, 1145–1159 (1997)
5. Dietterich, T.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. **10**, 1895–1924 (1998)
6. Demsar, J.: Statistical comparison of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
7. Dmochowski, J., et al.: Maximum likelihood in cost-sensitive learning: model specification, approximation and upper bounds. J. Mach. Learn. Res. **11**, 3313–3332 (2010)
8. Duda, R., Hart, P., Stork, D.: Pattern Classification and Scene Analysis. Wiley, New York (2000)
9. Drummond, C., Holte, R.: Cost curves: an improved method for visualizing classifier performance. Mach. Learn. **65**(1), 95–130 (2006)
10. Elkan, C.: The foundation of cost-sensitive learning. In: Proceedings of 4th International Conference Artificial Intelligence, vol. 17, pp. 973–978 (2001)
11. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**(8), 861–874 (2006)
12. Ferri, C., et al.: An experimental comparison of performance measures for classification. Pattern Recogn. Lett. **30**(1), 27–38 (2009)
13. Finner, H.: On a monotonicity problem in step-down multiple test procedures. J. Am. Stat. Assoc. **88**, 920–923 (1993)

14. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. Ann. Math. Stat. **11**, 86–92 (1940)
15. Gama J., et. al.: On evaluating stream learning algorithms. Mach. Learn., pp. 1–30 (2013)
16. Garcia, S., Herrera, F.: An extension on statistical comparison of classifiers over multiple datasets for all pair-wise comparisons. J. Mach. Learn. Res. **9**(12), 2677–2694 (2008)
17. Garcia, S., Fernandez, A., Lutengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in the computational intelligence and data mining: experimental analysis of power. Inf. Sci. **180**(10), 2044–2064 (2010)
18. García, V., Mollineda, R.A., Sánchez, J.S.: Index of balanced accuracy: a performance measure for skewed class distributions. In: Araujo, H., Mendonça, A.M., Pinho, A.J., Torres, M.I. (eds.) IbPRIA 2009. LNCS, vol. 5524, pp. 441–448. Springer, Heidelberg (2009). doi:10.1007/978-3-642-02172-5_57
19. Górecki, T., Krzyśko, M.: Regression methods for combining multiple classifiers. Commun. Stat. Simul. Comput. **44**, 739–755 (2015)
20. Hand, D., Till, R.: A simple generalization of the area under the ROC curve for multiple class classification problems. Mach. Learn. **45**, 171–186 (2001)
21. Hand, D.: Measuring classifier performance: a coherent alternative to the area under the ROC curve. Mach. Learn. **77**, 103–123 (2009)
22. Hand, D., Anagnostopoulos, C.: A better beta for the H measure of classification performance. Pattern Recogn. Lett. **40**, 41–46 (2014)
23. He, H., Garcia, E.: Learning from imbalanced data. IEEE Trans Data Knowl. Eng. **21**(9), 1263–1284 (2009)
24. Hochberg, Y.: A sharper Bonferroni procedure for multiple tests of significance. Biometrika **75**, 800–802 (1988)
25. Hodges, J.L., Lehmann, E.L.: Ranks methods for combination of independent experiments in analysis of variance. Ann. Math. Stat. **33**, 482–487 (1962)
26. Hollander, M., Wolfe, D.: Nonparametric Statistical Methods. Wiley, New York (2013)
27. Holm, S.: A simple sequentially rejective multiple test procedure. Scand. J. Stat. **6**, 65–70 (1979)
28. Iman, R., Davenport, J.: Approximations of the critical region of the Friedman statistic. Comput. Stat. **9**(6), 571–595 (1980)
29. Japkowicz, N., Stephen, N.: The class imbalance problem: a systematic study. Intell. Data Anal. **6**(5), 40–49 (2002)
30. Japkowicz, N., Shah, M.: Evaluating learning algorithms: a classification perspective. Cambridge University Press, Cambridge (2011)
31. Krzyśko, M., Wołyński, W., Górecki, T., Skorzybut, M.: Learning Systems. In: WNT, Warszawa (2008) (in Polish)
32. Kubat, M., Matwin, S.: Adressing the curse of imbalanced training sets: one-sided selection. In: Proceedings of the 14th ICML, pp. 179–186 (1997)
33. Kurzyński, M.: Pattern Recognition. Statistical Approach. Wrocław University Technology Press, Wrocław (1997) (in Polish)
34. Malina, W., Śmiatacz, M.: Pattern Recognition. EXIT Press, Warszawa (2010) (in Polish)
35. Nadeau, C., Bengio, Y.: Inference for the generalization error. Mach. Learn. **52**(3), 239–281 (2003)
36. Prati, R., et al.: A survey on graphical methods for classification predictive performance evaluation. IEEE Trans. Knowl. Data Eng. **23**(11), 1601–1618 (2011)

37. Ranavana, R., Palade, V.: Optimized precision: a new measure for classifier performance evaluation. In: Proceedings of the 23rd IEEE International Conference on Evolutionary Computation, pp. 2254–2261 (2006)
38. Quade, D.: Using weighted rankings in the analysis of complete blocks with additive block effects. J. Am. Stat. Assoc. **74**, 680–683 (1979)
39. Salzberg, S.: On comparing classifiers: pitfalls to avoid and recommended approach. Data Min. Knowl. Disc. **1**, 317–328 (1997)
40. Sánchez-Crisostomo, J.P., Alejo, R., López-González, E., Valdovinos, R.M., Pacheco-Sánchez, J.H.: Empirical analysis of assessments metrics for multi-class imbalance learning on the back-propagation context. In: Tan, Y., Shi, Y., Coello, C.A.C. (eds.) ICSI 2014. LNCS, vol. 8795, pp. 17–23. Springer, Cham (2014). doi:10.1007/978-3-319-11897-0_3
41. Santafe, G., et al.: Dealing with the evaluation of supervised classification algorithms. Artif. Intell. Rev. **44**, 467–508 (2015)
42. Shaffer, J.P.: Multiple hypothesis testing. Annu. Rev. Psychol. **46**, 561–584 (1995)
43. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Inf. Proc. Manag. **45**, 427–437 (2009)
44. Stąpor, K.: Classification methods in computer vision. In: PWN, Warszawa (2011) (in Polish)
45. Sun, Y., et al.: Classification of imbalanced data: a review. Int. J. Pattern Recogn. Artif. Intell. **23**(4), 687–719 (2009)
46. Sun, Y., et. al.: Boosting for learning multiple classes with imbalanced class distribution. In: Proceedings of International Conference on Data Mining, pp. 592–602 (2006)
47. Tadeusiewicz, R., Flasiński, M.: Pattern recognition. In: PWN, Warszawa (1991) (in Polish)
48. Wolpert, D.: The lack of a priori distinctions between learning algorithms. Neural Comput. **8**(7), 1341–1390 (1996)
49. Woźniak, M.: Hybrid classifiers. Methods of Data, Knowledge and Classifier Combination. SCI, vol. 519, Springer, Heidelberg (2014)

# Multi-aspect Assessment and Classification of Porous Materials Designed for Tissue Engineering

Małgorzata Przytulska and Juliusz L. Kulikowski(✉)

Nalecz Institute of Biocybernetics and Biomedical Engineering,
Polish Academy of Sciences, 4 Ks. Trojdena Street, 02-109 Warsaw, Poland
{mprzytulska,jkulikowski}@ibib.waw.pl

**Abstract.** The paper presents an approach to classification of porous materials used in biomedicine based on computer-aided analysis of scanning electron microscope images of sections of the examined material. Due to various size and high irregularity of forms of the pores visible in the images selected morphological parameters are used to the description of the samples of the porous material. The space of morphological parameters is automatically divided into porosity classes which are a step to establish the classes of porous material quality, based on suggestions of experts. An approach to verification of the morphological parameters utility to discriminate the materials according to their porosity is also proposed.

**Keywords:** Porous materials · Tissue engineering · Image analysis · Morphological parameters

## 1 Introduction

Typical pattern recognition problems deal with similarity classes of relatively regular, well defined objects. However, in some cases not only the classes of objects but also the objects themselves are not strongly defined as having extremely irregular form. In particular, in the area of technology of materials production for bioreactors, water filters, blood detoxification by dialysis, tissue engineering aimed at repairing of damaged or formation of new organs etc. porous materials in the form of membranes or scaffolds play a substantial role [1–3]. Such materials are usually based on ceramics, polymers or on their composites [4–6]. Porosity of such materials is one of the most important properties influencing their practical usefulness. However, geometrical form of pores cannot be easily described. On the other hand, numerical evaluation of pores is of high importance while in practical use of a porous material different role by different types of pores is played. In the case of porous polymer materials used as scaffolds in biological tissues production large pores constitute a three-dimensional microenvironment for cells migration and seeding while small pores provide to them the nutritive agents

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_3

and remove the metabolic waste products. The difference between the small and large pores is visible in an electron microscope image (1000× magnified) in Fig. 1. The variety and irregularity of the form of pores makes the porous materials classification based on simple geometrical concepts rather impossible. However, the porosity of materials can be examined by other, more sophisticated methods. It was shown in [7] that the methods based on computer-aided analysis of scanning electron microscope (SEM) images of the examined material sections combined with statistical methods may be fairly effective. Such methods give insight into the morphological structure of pores: their size, density, form, interconnectivity etc. The problem is how to establish the relations between measurable morphological parameters of the examined material and its being of interest biophysical properties. Till now, the mechanisms of biological cells' propagation, settling and growing in a porous environment have not been exactly investigated and described. Also the influence of large pores' size and geometrical form upon the effectiveness of tissue planting needs to be investigated. Therefore, the problem of selection of easily measurable numerical parameters as criteria of classification of porous materials from the point of view of their utility in biological cells cultures needs to be investigated. An attempt to solution of this problem aimed at classification of large pores in polyvinylpyrrolidone membranes designed for the construction of scaffolds for biological cell cultures is presented in the paper. The first step to do it was choosing measurable parameters adequate for pores' form description and classification. Special attention has been paid to the pores larger than 80 $\mu$m diameter as well as to the detection of existence of narrow (2-6 $\mu$m diameter) channels making possible cells propagation between the pores. The methods based on computer-aided image processing occurred to be an adequate tool for solution of the problem. The paper is organized as follows. The materials and image processing based methods used in this investigation are described in Sect. 2. The results of computer experiments based on the proposed method and aimed at classification of pores and at discrimination of different types of porous materials are presented in Sect. 3. Final conclusions are given in Sect. 4.

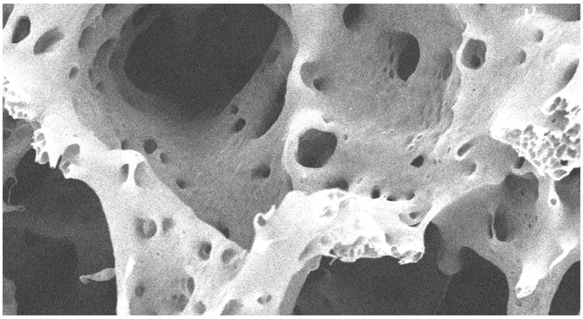
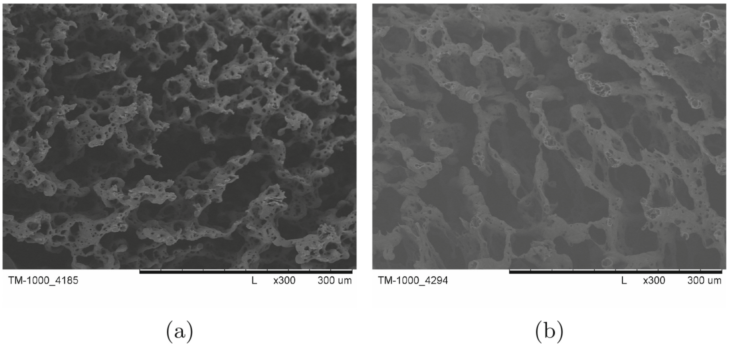


**Fig. 1.** Electron microscope image of porous material with visible two (small and large) types of pores.

# 2 Materials and Methods

## 2.1 Materials

There were investigated the images from a scanning electron microscope (SEM) type Hitashi TM1000 with 15 kV acceleration voltage. The obtained by inverse-phase method images taken with 300× and 1000× magnitude presented specimens of polyvinylpyrrolidone membranes. The images of lower, 300× magnification, as presenting more objets, occur to be more suitable for statistical analysis of pores Two types of membranes produced by alternative methods: type I of the PLLA:PVP 1:1 rate and type II PLLA:PVP:Pluronic® 100:100:25 rate were taken into consideration. Figure 2 presents original SEM images of the two types of membranes. Two types of pores in the images can be observed: small pores visible as black dots and large pores visible as dark areas of irregular form. Stochastic nature of the details observable in the images is also evident. Parameters characterizing both types of pores should be separately analyzed. However, in order to analyze the properties of large pores the images should be subjected to a preliminary image enhancement procedure consisting of image calibration (presentation in unified pixel/metric distances scale ratios), noise reduction, equalization of the histograms of luminance, filtering (by morphological closure) enhancing the separation of pores. Finally, the enhanced images ready for analysis take the form shown in Fig. 3.

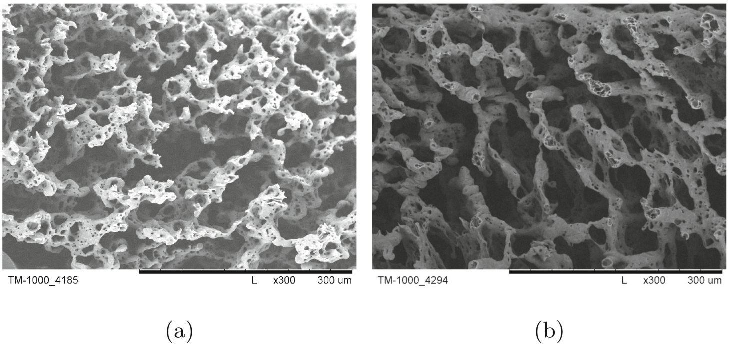


(a) (b)

**Fig. 2.** Original SEM images of membranes: (a) type I, (b) type II.

## 2.2 Methods

In this project standard Image Pro Plus image processing procedures were used. Enhanced images were subjected to the operations of contouring and separation of pores before being subjected to further image analysis operations. Due to the fact that the depth of pores is in the images expressed by local darkness level, contouring of pores can be performed by using a standard watershed method [8]. Fixing the "water level" on selected depths makes possible insight into the structure of connections between the pores. For better distinction of pores their rough



(a)

(b)

**Fig. 3.** Enhanced SEM images of membranes prepared for numerical analysis: (a) type I, (b) type II.

contours were smoothed by morphological closing operation [9]. For the analysis and classification of pores the following attributes were taken into consideration:

- object's (pore's section) area, proportional to the number of pixels whose value (brightness) does not exceed a threshold level;
- object's polygon area, calculated as area of a polygon outlining the object;
- holes, the number of holes inside a single object;
- maximal and minimal radii, defined as distances between the centroid (centre of mass) and the boundary of the object;
- radius ratio, the ratio of (maximal radius)/(minimal radius);
- roundness, the ratio of the square of object's perimeter to object's area normalized by this of a circle ($4\pi$); it takes vale 1 for a circle and >1 for other shapes of objects (remark that this, so called in literature attribute, describes rather a deflection from circular form);
- fractal dimension of object's perimeter.

Two different methods of pore's area evaluation make possible detection of objects containing external narrow stripes: their area is neglected in the polygon area measurement. However, existence of such stripes is of low importance in biological tissue production. The radius ratio, roundness and fractal dimension of the objects have been chosen as parameters characterizing various aspects of the irregularity of object's form. We denote by $U$ the set of attributes selected for pores description. The number of objects (pores) selected and analyzed in single images varied in the series of examined images between 75 and 45. In each image for each attribute the following parameters were calculated:

- its maximal and minimal value in all objects in the image,
- the range between the maximal and minimal parameter value in the image,
- number of objects corresponding to the maximal and minimal value,
- arithmetical mean value of the parameter in the image,
- standard deviation of the parameter in the image.

We denote by $V$ the set of types of parameters used to characterize the attributes. Therefore, the Cartesian product:

$$W = U \times V \tag{1}$$

constitutes a *parameter space* used to the description of pores. In particular, for porous materials assessment the space $W$ into a family of *quality classes* should be partitioned. The way to do it leads through a preliminary partition of $W$ into a family of *morphological classes* and then – by assigning to them some *quality weights*. The last step needs an intervention of experts assessing various (biophysical, medical, technological etc.) aspects of the examined materials.

## 3 Experiments

### 3.1 Morphological Classification

In two typical SEM images presenting the membranes type I and type II the parameters specified in Sect. 2.2 were measured. Figure 4 presents the next step of images from Figs. 2a and 3a preparation to pores segmentation and measurement of their parameters. Example of numerical experimental data obtained in a selected pair of images of type I and type II membranes are presented below by Tables 1 and 2. For selected parameters the numbers of classes can be established by a preliminary examination of the histograms of the parameter values. It was taken into account the fact that the number of classes should not be too large in order to avoid occurring empty classes. Selected histograms are shown in Fig. 5. Finally, by taking into account the histograms of morphological parameters and the accuracy of objects classification the following numbers of classes have been established: 3 area classes and by 4 classes for roundness and fractal dimension. This means that a 3-dimensional parameter space $W$ has been established and it has been divided into $3 \times 4 \times 4 = 48$ morphological classes. The classification of objects performed automatically by the Image Pro Plus procedure. The program neglected objects contiguous to image frame, because their (possibly) cancelled parts are not available for measurements. The result of ordering, according to their decreasing area, of the objects shown in Fig. 4b is shown in Fig. 6. The effects of ordering the objects selected in the same image evidently depend on the criterion of ordering, as it is visible in Fig. 7a, b and c (remark that the objects are here presented in different scale and position). The classifying program assigns identifying numbers to the objects and, optionally, it presents in various colors the objects included into different classes. For each ordering of objects performed according to a given criterion (parameter) and for each class of objects the following morphological parameters have been calculated:

1. class #,
2. number of objects in the given class,
3. % of objects in the class,
4. minimal value of the classifying parameter,

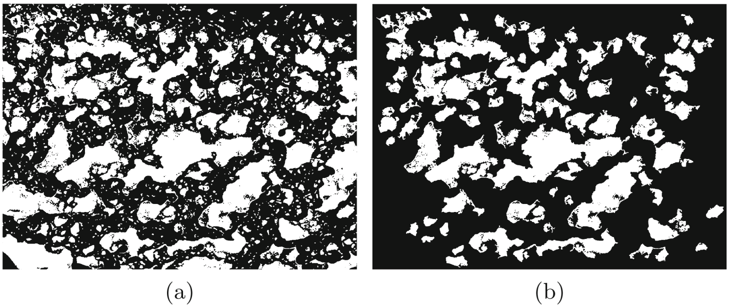


**Fig. 4.** SEM image prepared to selection and morphological analysis of pores: (a) binarized image, (b) image after removing undersize and cancelled objects

**Table 1.** Measured parameters of pores in membrane I

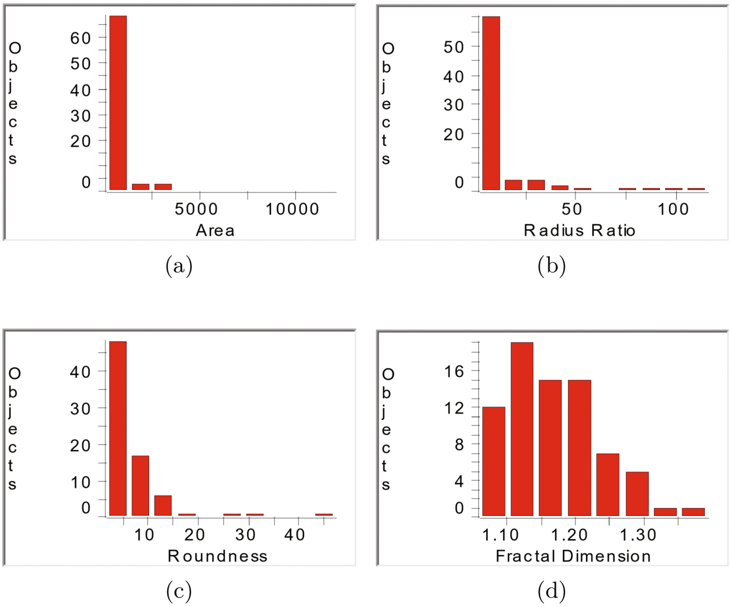
| Attribute parameter | Area | Rad. max | Rad. min | Rad. ratio | Round | Fract. dim |
|---|---|---|---|---|---|---|
| Minimal | 81.27 | 7.21 | 0.20 | 1.74 | 1.58 | 1.06 |
| Maximal | 12133.42 | 161.04 | 16.69 | 115.87 | 47.35 | 1.39 |
| Range | 12052.15 | 153.83 | 16.49 | 114.13 | 45.77 | 0.33 |
| Mean | 665.76 | 21.84 | 3.90 | 13.79 | 45.77 | 1.18 |
| St. dev. | 1510.62 | 22.12 | 2.88 | 23.91 | 7.32 | 0.08 |
| Samples # | 75 | 75 | 75 | 75 | 75 | 75 |

**Table 2.** Measured parameters of pores in membrane II

| Attribute parameter | Area | Rad. max | Rad. min | Rad. ratio | Round | Fract. dim |
|---|---|---|---|---|---|---|
| Minimal | 87.97 | 7.49 | 0.74 | 1.49 | 1.30 | 1.06 |
| Maximal | 5250.37 | 75.09 | 16.8 | 23.84 | 18.53 | 1.34 |
| Range | 5162.40 | 67.60 | 16.00 | 22.35 | 17.23 | 0.28 |
| Mean | 701.14 | 20.47 | 5.11 | 6.11 | 3.86 | 1.15 |
| St. dev. | 1124.76 | 15.65 | 3.97 | 5.30 | 3.17 | 0.06 |
| Samples # | 45 | 45 | 45 | 45 | 45 | 45 |

5. maximal value of the classifying parameter,
6. mean value of the classifying parameter.

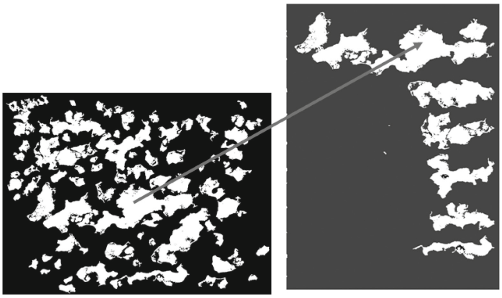
Example of calculated morphological parameters of area classes (Table 3), roundness classes (Table 4) and fractal dimension classes (Table 5) are given below. The calculated, based on observations, parameters make possible fixing intervals for classification of objects according to their morphological parameters. The intervals are fixed by choosing separatying points between the maximal parameter



(a) (b)

(c) (d)

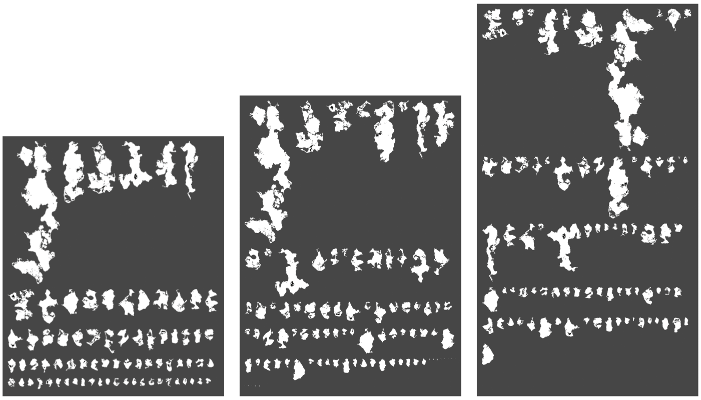
**Fig. 5.** Histograms of selected parameters of objects observed in type I membrane: (a) area, (b) radius ratio, (c) roundness, (d) fractal dimension.



(a) (b)

**Fig. 6.** Automatic ordering of objects according to their decreasing area: (a) repeated image from Fig. 4b, (b) selected and ordered several first largest objects

value of a former class and the minimal parameter value of the next class, as it is shown in Table 6. The fixed intervals of morphological parameters' values establish a partition of pores into classes. In practice, the morphological parameters of pores should be evaluated on the basis not of one (as, for illustration of the method, has been shown above) but of a series of images presenting various samples of the material under examination. The fixed intervals of morphological



(a) (b) (c)

**Fig. 7.** Effects of automatic ordering of objects selected in the same image according to different criteria: (a) object area, (b) roundness, (c) fractal dimension

parameters establish a partition of parameter space $W$ into non-overlapping *morphological classes*:

$$W = \bigcup_{[\alpha,\beta,\ldots,\varepsilon]} C_{\alpha\beta\ldots\varepsilon} \tag{2}$$

where $C_{\alpha\beta\ldots\varepsilon}$ denotes a Cartesian product of, correspondingly, the $\alpha$-, $\beta$-, $\ldots$, $\varepsilon$- th morphological class of the first, second, etc. up to the last (in a given order taken) morphological parameter. However, the morphological classes are not identical to the desired classes of quality of the porous material under examination. Untill the relations between the morphological parameters and the biophysical properties of porous materials are fully explained the correspondence between the morphological and quality classes cannot be established but on the basis of suggestions of experts. This should be aimed at introduction of a semi-ordering (reflexive, antisymmetrical and transitive) relation into the family of morphological classes, according to the quality assigned to them by the experts.

**Table 3.** Parameters of area classes of selected objects

| Parameter class | # objects | % objects | Mean area $[\mu\text{m}^2]$ |
|---|---|---|---|
| 1 | 48 | 64.00 | 267.58 |
| 2 | 26 | 34.67 | 959.80 |
| 3 | 1 | 1.33 | 12133.42 |

**Table 4.** Parameters of roundness classes of selected objects

| Parameter class | # objects | % objects | Min. par. value | Max. par. value | Mean par. value |
|---|---|---|---|---|---|
| 1 | 57 | 55.88 | 1.000 | 4.183 | 2.064 |
| 2 | 43 | 42.16 | 4.312 | 19.838 | 8.311 |
| 3 | 1 | 0.98 | 30.556 | 30.556 | 30.556 |
| 4 | 1 | 0.98 | 46.102 | 46.102 | 46.102 |

**Table 5.** Parameters of fractal dimension classes of selected objects

| Parameter class | # objects | % objects | Min. par. value | Max. par. value | Mean par. value |
|---|---|---|---|---|---|
| 1 | 38 | 48.101 | 1.066 | 1.162 | 1.118 |
| 2 | 26 | 32.911 | 1.166 | 1.239 | 1.201 |
| 3 | 14 | 17.722 | 1.247 | 1.319 | 1.279 |
| 4 | 1 | 1.266 | 1.388 | 1.388 | 1.388 |

**Table 6.** Intervals of morphological classification of objects

| Class parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Area $[\mu\mathrm{m}^2]$ | 0÷300 | 301÷1000 | > 1000 | – |
| Roundness | 1.00÷4.20 | 4.21÷20.00 | 20.01÷40.00 | > 40.00 |
| Fractal dim | 1.000÷1.165 | 1.166÷1.240 | 1.241÷1.380 | 1.381÷2.000 |

### 3.2 Evaluation of Discriminating Power of Morphological Parameters

Despite the classification of porous materials from the point of view of their quality a problem of detection of a difference between morphological properties of any two (or more) samples of the materials under examination is of interest. This, in particular, suits to a control of a technological material production process stability. For this purpose, the discriminating power of selected morphological parameters was evaluated. The parameters were measured in two samples of porous material produced by alternative methods. For any parameter $\gamma$ whose values measured in the two samples of material are denoted by $\gamma_1$ and $\gamma_2$ the discriminating power of $\gamma$ denoted by $d_\gamma$ has been defined by the formula

$$d_\gamma = 2\frac{|\gamma_1 - \gamma_2|}{\gamma_1 + \gamma_2}. \tag{3}$$

The vales of $d_\gamma$ for selected morphological parameters are given in Table 7. It can be remarked that some parameters, like: *maximal number of holes per object*, *range of radii ratios* or *mean roundness* are particularly predestined to be used to discriminate the porous materials.

**Table 7.** Discriminating power of selected morphological parameters

| Parameter/Statistics | Area | Holes /obj | Rad. max | Rad. ratio | Round | Fract. dim |
|---|---|---|---|---|---|---|
| Minimal | 0.0790 | 0.0000 | 0.0381 | 0.1548 | 0.1944 | 0.0000 |
| Maximal | 0.7919 | 1.5357 | 0.7280 | 1.3174 | 0.8749 | 0.0366 |
| Range | 0.8004 | 1.5357 | 0.7788 | 1.3450 | 0.9060 | 0.1639 |
| Mean | 0.0517 | 0.7232 | 0.0648 | 0.7719 | 1.7333 | 0.0258 |

## 4 Conclusions

Recognition of objects of highly irregular form needs using adequately chosen attributes for their characterization, classification and establishing classes of similarity. Porous materials used in technology and/or biomedical engineering need to be classified according to the type of pores playing various roles in their practical using. The size of pores may vary in large intervals, their topological features may be sophisticated and their geometrical form may be highly irregular. Modern computer aided image processing methods provide effective tools for quantitative analysis of such objects. However, for this purpose adequate morphological parameters should be chosen. The automatically measured parameters are, in fact, instances of random variables of unknown probability distribution. On the other hand, for various reasons the pdf-s of these variables cannot be exactly described but by histograms of some limited numbers of observations. That is why classification of such objects cannot be reached by solution of any strongly defined analytical problem. For this purpose rather a heuristic approach based on some elements of statistical reasoning can be used. The paper presents such approach to pores classification based on their presentation by elements of a space of morphological parameters. The, introduced here, parameter space $W$ is neither a linear vector space nor a metric space. However, multi-aspect similarity measures in this space can be defined and the problem of types of porous materials recognition can be solved.

**Acknowledgement.** The authors would like to thank prof. Andrzej Chwojnowski and his P.T. Collaborators from the Nalecz Institute of Biocybernetics and Biomedical Engineering of PAS in Warsaw for providing us with examples of SEM images of porous materials used in this work.

## References

1. Thongmaka, N., Sridangb, P., Puetpaiboona, U., He'rane, M., Lesagee, G., Grasmicke, A.: Performances of a submerged anaerobic membrane bioreactor (AnMBR) for latex serum treatment. Desalination Water Treat. **57**, 20694–20706 (2016)
2. Johira, M.A.H., Shimb, W.G., Pradhana, M., Vigneswarana, S., Kandasamya, J.: Benefit of adding adsorbent in submerged membrane microfiltration treatment of wastewater. Desalination Water Treat. **57**, 20683–20693 (2016)

3. Sueoka, A.: Present status of apheresis technologies: part 2. membrane plasma fractionator. Ther. Apheresis **1**(2), 135–146 (1997)
4. Tanaka, Y., Yamaoka, H., Nishizawa, S., Nagata, S., Ogasawara, T., Asawa, Y., et al.: The optimization of porous polymeric scaffolds for chondrocyte/atelocollagen based tissue-engineered cartilage. Biomaterials **31**, 4506–4516 (2010)
5. Płończak, M., Czubak, J., Hoser, G., Chwojnowski, A., Kawiak, J., Dudziński, K., Czumińska, K.: Repair of articular cartilage defects with cultured chondrocytes on polysulphonic membranes experimental studies in rabbits. Biocyb. Biomed. Eng. **28**, 87–93 (2008)
6. Chen, C.C., Wang, W.C., Ding, S.J.: In vitro physiochemical properties of a biomimetic gelatin/chitosan oligosaccharide/calcium silicate cement. J. Biomed. Mater. Res. B Appl. Biomater. **95**, 456–465 (2010)
7. Chwojnowski, A., Przytulska, M., Wierzbicka, D., Kulikowski, J.L., Wojciechowski, C.: Membrane's porosity evaluation by computer-aided analysis of SEM images, a preliminary study. Biocyb. Biomed. Eng. **32**(4), 65–75 (2012)
8. Russ, J.C.: The Image Processing Handbook, 6th edn. CRC Press, Boca Raton (2011)
9. Pitas, I.: Digital Image Processing Algorithms and Applications. A Wiley-Interscience Publication, New York (2013)

# Enhancing English-Japanese Translation Using Syntactic Pattern Recognition Methods

Thomas McMahon and B. John Oommen(✉)

School of Computer Science, Carleton University, Ottawa, Canada
thomasmcmahon@cmail.carleton.ca, oommen@scs.carleton.ca

**Abstract.** In this paper, we present a novel approach to Machine Translation (MT) using syntactic Pattern Recognition (PR) methods. Our aim is to evaluate the possibility of using syntactic PR techniques in this mature field, and to identify any potential benefits that can be gleaned by such an approach. To make use of syntactic PR techniques, we propose a system that performs string-matching to pair English sentence *structures* to Japanese (The specific languages, namely English and Japanese, were chosen because their sentence structures are completely dissimilar. This, however, proves the point that such syntactic methods will be applicable for other pairs of languages too.) *structures* – as opposed to matching strings in and of themselves, and to thus facilitate translation between the languages. In order to process the sentence structures of either language as a string, we have created a representation that replaces the tokens of a sentence with their respective Part-of-Speech tags. Further, to perform the actual string-matching operation, we make use of the OptPR algorithm, a syntactic award-winning PR scheme that has been proven to achieve optimal accuracy, and that also attains the *information theoretic bound*. Through our experiments, we show that our implementation obtains superior results to that of a standard statistical MT system on our data set. Our results provide the additional guarantee of generating a known sentence structure in the target language. With further research, this system could be expanded to have a more complete coverage of the languages worked with. The incorporation of such PR techniques in MT, in general, and the OptPR algorithm, in particular, are both pioneering.

**Keywords:** Machine Translation (MT) · Syntactic Methods in MT

## 1 Introduction

This paper is concerned with the field of research known as Machine Translation (MT). This involves the task of a computer translating the text written in one language into that of another, without the need of human intervention. Specifically, we are interested in English-Japanese MT, where English text is automatically converted into its equivalent text in the Japanese language [8]. MT is a sub-domain of Natural Language Processing, a field of research concerned with transforming the text of natural languages into a form that can be

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_4

understood and manipulated by machines. The term “Natural Language” refers to languages that evolved in human society, as opposed to artificially constructed languages, or explicitly-defined languages that lack evolutionary development. English-Japanese translation is a particularly difficult language-pairing, as there is no known common ancestry between the two languages, resulting in fundamentally distinct sentence structures [2]. One approach to handling the translation of two such structurally-distinct languages is to create a method of transforming the *structure* of one language so that it matches the the *structure* of other – which is far from being a trivial exercise. One potential way of accomplishing this is to compare the structure of an English sentence to every possible Japanese sentence structure, and to somehow be able to tell which pairing is the best match. The issue then is one of determining which pairing is ideal. There is a class of algorithms that are capable of matching structures in this way, i.e., those that invoke Syntactic Pattern Recognition (PR) techniques. Making use of such algorithms could potentially help in improving translation between two languages as distant as English and Japanese. This is the focus of this paper.

### 1.1 Objective

The goal of this paper is to show the viability of using syntactic PR methods in the field of MT. The potential benefits of using such an approach are great, particularly in being able to ensure that the output of the translation system is, at least, structurally sound. Before we proceed, it is pertinent for us to highlight the scope of the paper. This paper does not present a complete MT system – embarking on the latter would be the task of a “lifetime” for a small team. Rather, our goal is to demonstrate the applicability of syntactic PR tools, that have not been previously used in MT. In that sense, we are proposing the use of these tools in MT components that could potentially be *used* in more comprehensive MT systems. Further, since our aim is to demonstrate a *prima facie* case for these tools, our component essentially takes as its input an English sentence with a rather simple form, i.e., it contains a Subject Phrase, a Verb and an Object Phrase, and these are labelled with Part-of-Speech (POS) tags. It matches this against a collection of potential Japanese sentence structures containing analogous POS tags.

### 1.2 Contributions of This Paper

In this paper, we have opted to embark on a novel approach to MT by incorporating syntactic PR methods, and these are combined with our own representation for sentence structures. The contributions of this paper are thus as follows:

1. String Representation of Sentence Structures: We make use of a unique representation of sentences that corresponds to their *structures* rather than their meanings. Instead of an exhaustive list of sentences, these sentence structures are stored so as to replace the approximations of a language model.
2. Phrase-Level Template Parsing: This is a novel scheme that we will employ.

3. Word Alignment Utilization: Here, we re-purpose existing statistical MT components for our syntactic PR system.
4. Machine Translation using the OptPR Algorithm: By making use of our string-based representation and word alignment models, we choose to use the above-mentioned syntactic PR algorithm (the OptPR algorithm [9]) to effectively match English structures to Japanese structures contained in a set list. By doing this, we can perform accurate translation with the guarantee that the output will have a known Japanese sentence structure.

# 2 Survey of the Field

## 2.1 Machine Translation

Machine Translation[1] is the process by which a machine automatically converts text from one natural language into another. The language in which the text originated from is referred to as the "Source Language" (SL), and the language of the output text is called the "Target Language" (TL). The primary goal of a MT system is to preserve the intended meaning of the source text by preventing any loss of information. There are numerous instances where a TL cannot fully represent the meaning of the source text, and so loss of information is inevitable. Ideally, situations of this type should be the only instances where information loss is observed within the system. That being said, various challenges associated with working with a natural language, cause loss at other junctures. The secondary goal of a translation system is to correctly translate the input text into the TL, with no *grammatical* errors. Depending on the differences between the source and target languages, this too can be a very difficult task.

### 2.1.1 Example Based Machine Translation (EBMT)

At its core, an EBMT is a MT system that uses a parallel corpus as its primary translation utility. Beyond this, the boundaries of what constitutes an EBMT are vague, as many systems borrow techniques typically found in Transfer or Statistical Machine Translation (SMT) systems [4]. The use of a parallel corpus of example sentences is not sufficient enough to distinguish EBMT systems from other types of systems. This is because SMT uses the same concept to build its language model, and a Rule-Based MT system could use such a corpus to derive the rules. However, EBMT uses the examples themselves as the *primary* knowledge base, as opposed to any data derived from it. In other words, an EBMT system uses the corpus at runtime, while an SMT model uses one to build the translation model *prior* to the actual translation. This topic is covered in greater detail in [7] and in the thesis of the first author [6].

[1] Due to space constraints, this survey is necessarily brief.

## 2.2 Syntactic PR

Pattern Recognition (PR), in general terms, is the process of taking input data and labelling it in some way, based on its presentation. The simplest approach is to manually define a set of rules using which one can compare the input data against the training data and decide its label. However, attempting to do so against real data can become a large undertaking, both in the knowledge required within the domain to define well-formed rules, as well as the number of rules needed to sufficiently cover the problem. Most systems, thus, use techniques that are able to automatically build a model with which PR can be performed. In that sense, the type of model built depends greatly on the PR algorithm used.

### 2.2.1 Syntactic Pattern Recognition

The basic idea behind Syntactic PR is that the pattern being processed is represented by a variable-sized set of *nominal* (as opposed to *numerical*) features, to avoid losing any structural information from the source. This allows for multiple types of representations, including hierarchical, relational graphs, and string-based representations. As mentioned previously, this paper uses Syntactic PR methods to assist in MT. This is the distinguishing feature of our paradigm. The approach that we invoke is different from the ones typically used in the literature, because a string-based representation is used rather than a heavily-structured one. In this regard, we utilize research performed by Oommen *et al.* [9] to develop the Information Theoretic Optimal Pattern Recognition (OptPR) algorithm to achieve this.

### 2.2.2 The OptPR Model

The OptPR algorithm has two major modules: a dictionary of strings (or sequences), and a model of a channel that arbitrarily adds noise to a string based on three distinct distributions. It operates by assuming that the input pattern to be recognized is one of the entries in the dictionary, garbled after passing through the noisy channel. Using the channel's model, the likelihood of the noisy pattern being a certain entry in the dictionary can be calculated, so that the dictionary entry which is most likely to be the original string, can be determined. The first component, the dictionary, is a set of example patterns against which the input has to be matched. If the PR problem involves matching noisy English words with ungarbled, actual English words, the dictionary would consist of the actual words. Both the original and noisy version of the strings must consist of strings from the corresponding alphabets. The OptPR model achieves significantly high results in string-based syntactic PR as presented in the thesis version of this paper's research [6], as well as in the original paper that pioneered the OptPR model [9]. In the subsequent sections, we briefly present the various contributions of this research.

## 3 Contribution I: Template Sentences

A Template Sentence (TS) is a representation of a sentence that replaces some or all of the tokens that it contains with placeholders. The placeholders represent, in some way, a generic form of the word or phrase that it replaces. In this case, the placeholders that are used are the syntactic word classes of the original tokens. TSs are used to represent the parallel corpus in both building the translation model as well as in creating the dictionary to be used in the translation algorithm. The main purposes of using TSs in translation are to reduce the data needed to build a translation model as well as in creating a translation system that focuses on forming syntactically correct sentences in the TL. By constraining the system so that the source text is matched with a TS that occurred in the parallel corpus, the system guarantees that the output of translation will correspond to a sentence structure that has, indeed, occurred in the TL. TSs can be used to perform various levels of translation between two languages. The simplest example of this would be that of having a single word in a sentence being replaced by a placeholder in the two respective languages. In such a case, the word being added to the sentence in the SL only needs to be translated and placed into the matching sentence in the TL. Multi-word translations can also be performed using this same process, but some sort of mapping operation is needed so that the input words in the source text can be correctly placed in the TL sentence. A major issue in using TSs in translation is the need to "invent" such a mapping function between the two languages. This is a problem similar to that of SMT, where the translation model is built using a parallel corpus without word-to-word mappings. The same algorithms used in SMT can be used for TS translation, except that instead of using the raw parallel corpus to build the translation model, every entry in the corpus is replaced by its TS equivalent.

## 4 Contribution II: Enhancements Using Syntactic PR

### 4.1 Word Alignment Utilization

Word Alignment tools, specifically GIZA++, are used to produce the probability distributions that are required for the OptPR algorithm to function properly. The input that must be supplied are two files containing the raw sentences of the respective languages, with equivalent sentences of either language occurring at the same line number. The pre-processing stage generates co-occurrence files, counting the number of times two tokens of the respective language occurred within the same sentence. Additionally, the number of times a token appears in the entire corpus is counted, so that one can calculate how often a token appears together with another *against* how often it appears by itself. The details of how the GIZA output is used to compute the distributions needed for the OptPR algorithm are clearly described in [6,7].

### 4.2 MT Using OptPR Algorithm

As the reader should appreciate, the main component of this paper is the usage of syntactic PR algorithms to assist in MT, and in particular, the OptPR algorithm. This algorithm has been used to perform string matching between garbled and ungarbled English words, and our goal is to utilize the same functionality for a different purpose. Along with the three probability distributions covered earlier, the algorithm requires a dictionary and an alphabet. The dictionary is the set of strings that the input is matched against, and the alphabet contains the elements that compose both the strings within the dictionary and the input strings. The phenomenon of modeling using a "noisy channel" has, of course, been used in the prior SMT art. However, the channel used in in previous SMT systems operates somewhat differently from the Syntactic PR channel model. We shall now clarify how a noisy channel is invoked in SMT systems. Consider Fig. 1 which depicts a generalized noisy channel model typically used for SMT [5]. The basic idea behind the SMT noisy channel approach is that it assumes that the SL sentence being translated, $f$, originally belonged to the TL. In order to obtain this TL sentence $e$, it invokes a language model to generate sentences. This is done because it is not feasible to collect every potential sentence belonging to the TL. The original TL sentence is assumed to have been transformed into the input by it having passed through the translation model. Finally, the original sentence in the TL is estimated using a decoder, yielding a translation $\hat{e}$. We now describe the noisy channel used by the OptPR algorithm. First, a random number of insertions is chosen using the distribution $G$. Once a number is obtained, the positions in which the insertions will occur are chosen, with each potential position in $U$ being equally likely. When the positions are selected, they are filled by randomly selecting an element in $A$ using the distribution $Q$. Afterwards, the original elements of $U$ are randomly substituted/deleted using the distribution $S$. Once this is complete the input null symbols, $\lambda$, are dropped and the string $Y$ is attained. The image below shows the basic process of how $Y$ is obtained from $U$. With regard to functionality, the translation model of the SMT noisy channel can be seen to be similar to that of the OptPR noisy channel. The essential purpose of the translation model is to estimate the combined probability of a sentence $e$ becoming a sentence $f$ given every potential alignment between the two sentences. On the other hand, the OptPR noisy channel estimates the total probability of a string $U$ becoming a string $Y$ by means of all potential sequences of elementary edit operations. These edit operations are performed by invoking the three distinct distributions depicted in the figure describing the OptPR noisy channel. This, combined with the fact that a dictionary is used in place of a language model, allows us to optimally and "totally" search the problem space during the decoding phase. For more details on how the OptPR algorithm is used by us, we refer the reader to [6,7]. Due to space constraints, it is impossible to explain it here (Fig. 2).

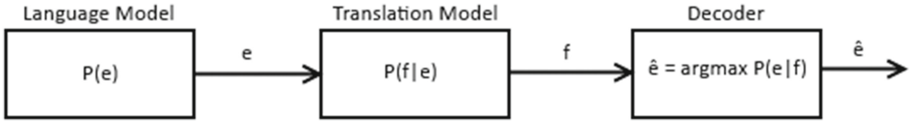


**Fig. 1.** The language model generates a TL sentence, $e$, that is transmitted through the channel's translation model into the SL sentence, $f$. The original sentence $e$ is then estimated with a decoder function obtaining $\hat{e}$.

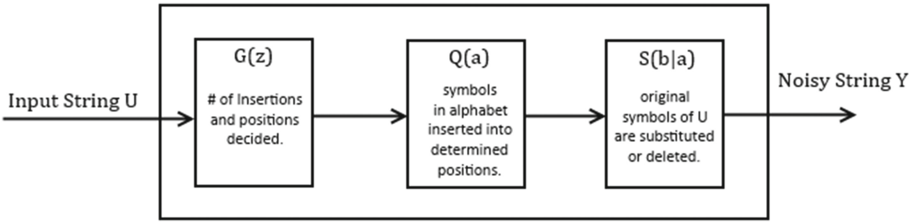

Input String U
G(z)
# of Insertions and positions decided.
Q(a)
symbols in alphabet inserted into determined positions.
S(b|a)
original symbols of U are substituted or deleted.
Noisy String Y

**Fig. 2.** The three distributions of the OptPR noisy channel are used to generate a noisy string $Y$ from $U$.

# 5 Experimental Setup and Results

## 5.1 Data Acquisition and Processing

The corpus used in our experimentation is the Tanaka corpus. Initially, there were a great deal of errors and repeated examples in it, but the collection has since been revised and corrected, reducing the number of sentence pairs to just over 150,000. This revised set, currently maintained by the Tatoeba group [10], was the one that we used. In order to help ensure that the data used during experimentation is valid, additional filtering steps are used to validate the sentence pairs, at the expense of reducing the size of the data set. The two filters used are syntactic filtering and content agreement filtering explained in [6,7]. Based on these filters, two data sets are used in our experiments: one with merely syntactic filtering applied containing 2807 sentence pairs, and the other with both syntactic and content agreement filtering applied containing 834 sentence pairs. The set containing 834 instances will be used as the training data for our primary evaluations, but the less restricted set will also be analyzed to capture how well the system performs with potentially noisier data.

### 5.1.1 Grammar Used in Experimentation

Our current *prima facie* approach works within a limited scope of the complete English and Japanese languages, placing restrictions on the grammar that this system can process. On the English side, we limit the acceptable sentences to be of the standard subject-verb-object structure. To achieve this, we first obtain a constituent parse of the sentence being processed, obtained using the OpenNLP toolkit [1]. We then ensure that the three child nodes of the sentence node are (in the specified sequence) "NP" (Noun Phrase), "VP" (Verb Phrase), and a period. Thereafter, we check that the VP discovered has two children, consisting of a verb (or another VP) and a NP child. As for the Japanese language, we verify that the sentence consists of a verb, as well as subject and object markers[2].

### 5.2 Baseline Experiment

In order to properly evaluate the power of the system proposed by this paper, it is important to have results of existing translation systems against which it can be compared. Due to the unique representation of TSs used in this paper, no existing data is available that our results can be directly compared against. Instead, we have opted to construct[3] an SMT using standard tools with the same data used in our own experiments. Three major packages have been used in constructing the SMT used to create the baseline results, the Moses toolkit, the IRST Language Modeling library, and the GIZA++ Word Alignment Tool.

### 5.3 Experimental Results and Analysis

We have conducted numerous experiments using a variety of settings, but it is not possible to present all of them here. More details of additional experiments, settings, results and comparisons are found in [6,7]. For this paper, the comparison of our experimental results and the baseline are shown in Fig. 3. Overall, our system outperforms the baseline SMT in terms of both producing sentences which are syntactically-correct, and in achieving string matching with the correct TS. One factor that could contribute to this is that a standard SMT is designed to work with large amounts of data to create sophisticated models. Our system is able to make better use of the smaller amount of data available in our data-set. Finally, a major drawback of SMTs is their ability to create sentences that are close to ones that would appear in the TL but which could have minor grammatical errors. With the usage of the OptPR algorithm this is much less likely to happen, as the output must have originated from a preset "dictionary" – the target TSs in the training data. The reason for an SMT generating syntactically-incorrect sentences is due to the fact that it has to rely on a language model to approximate the probabilities of a sentence existing in the language. Our system does not make use of such a language model, as the OptPR algorithm instead compares the input against a set of data. Additional experimental results and more detailed analyses are included in in [6,7], where the former is the thesis on which this paper is based [6].

[2] As such, our current experiments do not deal with certain major issues that affect MT, such as structural ambiguity [3], which we suggest as a future research avenue.

[3] We mention, in passing, that this was no small endeavor!

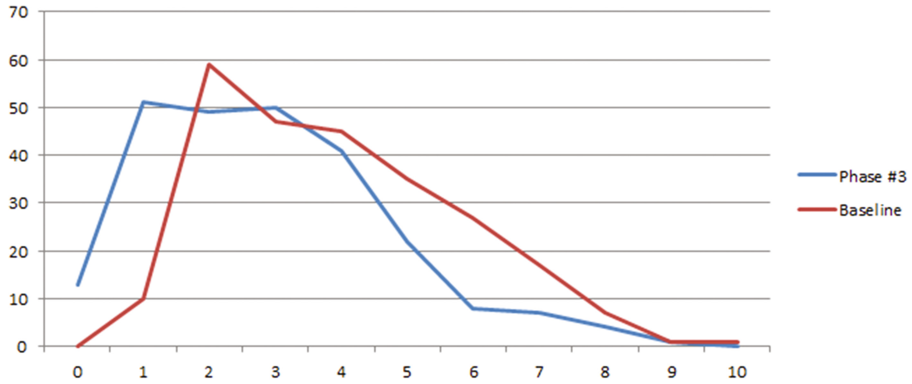


**Fig. 3.** Comparison graph between the baseline SMT results and the results of Phase-3. While the results of Phase-1 show that the OptPR algorithm can produce comparable results to the baseline with the benefit of syntactic correctness, Phase-3 shows that with further processing our system is capable of surpassing the baseline results while still maintaining that correctness.

# 6 Conclusion

## 6.1 String Representation of Sentence Structures

As mentioned previously, the optimal way by which we can ensure that a sentence belongs to a given language is to locate it within a list of every potential sentence for that language. As a proxy for this ideal method, we have chosen to perform essentially the same operation, but instead, to compare it against a list of potential sentence *structures*. To achieve this, we have created a unique representation of a sentence structure, referred to in this paper, as a "Template Sentence". This representation extracts the essential meaning from a sentence, replacing every token with its Part-of-Speech equivalent. The TSs extracted from our training corpus were used to build both our system and a standard statistical MT system. In spite of it removing the meaning from the training data, both systems were able to translate the test sentence structures fairly accurately. Additionally, our system was able to translate the test data with many perfect and close-to-perfect matches, showing that our list of sentence structures attains coverage of the target language beyond that of our training data. Because of this finding, we believe that our system is capable of functioning without a standard language model.

## 6.2 Machine Translation Using the OptPR Algorithm

The core of our research involved incorporating and assessing the viability of using syntactic PR techniques to the field of MT. The specific algorithm we have utilized is the OptPR algorithm, an algorithm that matches an input string against a set of strings [9], using probabilistic models to determine the most likely candidate. By making use of the template sentence representation and the word alignment models, we were able to construct the list and probability models required for this algorithm to function. Our system, using the OptPR algorithm without any post-processing enhancements, achieves results comparable to that of the baseline system. The major benefit of using the OptPR algorithm is that it guarantees that the chosen sentence structure has appeared in the training data. The baseline system, on the other hand, is more capable of generating sentence structures that are close to the expected result, but are themselves not grammatically correct. The fact that our system does not rely on a language model, but instead uses a rigid list, serves to limit the potential sentence structures generated by our system to those contained in the training data. However, as our system achieves comparable or even superior results over the baseline system, we have observed that the structures discovered in the training data sufficiently cover the structures contained in our test cases. Further, our system prevents the over-generation of grammatically-incorrect structures, while it is not dramatically affected by a potential lack of coverage over the complete set of sentence structures permissible for the target language.

## References

1. Baldridge, J.: The OpenNLP Project (2005). http://opennlp.apache.org/index.html
2. Fujinami, T., Nanz, C.: The 101 Translation Problems between Japanese and German/English. University Stuttgart (1997)
3. Hindle, D., Rooth, M., Ambiguity, S., Relations, L.: Structural ambiguity and lexical relations. In: Computational Linguistics - Special Issue on Using Large Corpora, Cambridge, USA, vol. 19, 103–120 (1993)
4. Hutchins, J.: Towards a definition of example-based machine translation. In: Proceedings of Workshop on Example-Based Machine Translation, Phuket, Thailand, pp. 63–70, September 2005
5. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
6. McMahon, T.: Enhancing Machine Translation for English-Japanese Using Syntactic Pattern Recognition Methods. MCS Thesis, School of Computer Science, Carleton University, Ottawa, Canada, May 2015
7. McMahon, T., Oommen, B.J.: On the Use of Syntactic Pattern Recognition Methods to Enhance English-Japanese Machine Translation. Unabridged version of this paper. Submitted for Publication
8. Nagao, M.: A framework of a mechanical translation between Japanese and English by analogy principle. In: Proceedings of the International NATO Symposium on Artificial and Human Intelligence, New York, USA, pp. 173–180 (1984)
9. Oommen, B.J., Kashyap, R.L.: A formal theory for optimal and information theoretic syntactic pattern recognition. Pattern Recogn. **31**, 1159–1177 (1998)
10. Ho Tatoeba, T., et al.: Collecting Example Sentences (2014). http://tatoeba.org/eng/home

# Novel Results on Random Walk-Jump Chains That Possess *Tree*-Based Transitions

Anis Yazidi[2] and B. John Oommen[1](✉)

[1] Department of Computer Science,
Oslo and Akershus University College of Applied Sciences, Oslo, Norway
oommen@scs.carteton.ca
[2] School of Computer Science, Carleton University, Ottawa, Canada

**Abstract.** The most difficult task in analyzing and appraising algorithms in Artificial Intelligence (AI) involves their formal mathematical analysis. In general, such an analysis is intractable because of the size of the search space and the fact that the transitions between the states within this space can be very intricate. That is why AI algorithms are, for the most part, evaluated *empirically* and *experimentally*, i.e., by simulations. However, whenever such an analysis is undertaken, it usually involves an analysis of the underlying stochastic process. In this connection, the most common tools used involve Random Walks (RWs), which is a field that has been extensively studied for more than a century [6]. These walks have traditionally been on a *line*, and the generalizations for two and three dimensions, have been by extending the random steps to the corresponding neighboring positions in one or many of the dimensions. The analysis of RWs on a *tree* have received little attention, even though it is an important topic since a tree is a counter-part space representation of a line whenever there is some ordering on the nodes on the line.

Nevertheless, RWs on a tree entail moving to *non-neighbor* states in the space, which makes the analysis involved, and in many cases, impossible. This is precisely what we achieve in this rather pioneering paper. The applications of this paper are numerous. Indeed, the RW on the tree that this paper models, is a type of generalization of dichotomous search with faulty feedback about the direction of the search, rendering the real-life application of the model to be pertinent. To resolve this, we advocate the concept of "backtracking" transitions in order to efficiently explore the search space. Interestingly, it is precisely these "backtracking" transitions that naturally render the chain to be "time reversible". By doing this, we are able to bridge the gap between deterministic dichotomous search and its faulty version, explained, in detail, in [21].

**Keywords:** Time reversibility · Controlled random walk · Random walk with jumps · Dichotomous search · Learning systems

J. Oommen—*Chancellor's Professor*; *Fellow: IEEE* and *Fellow: IAPR*. This author is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway. The work of this author was partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada.

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_5

## 1 Introduction

Academics and practitioners are usually amazed by the fact that verification in the field of Artificial Intelligence (AI) is so empirical and simulation-based. The reason for this is, quite simply, because the most difficult task in evaluating AI algorithms involves their formal mathematical analysis. The fact of the matter is that the state space in which these algorithms operate is prohibitively large. Further, the transitions between the states within this space can be very intricate, especially if the AI paradigm that is invoked requires the algorithm to migrate to non-neighbor states of the space. This renders a mathematical analysis of the AI algorithm to be intractable. Thus, AI is for the most part, a science where quantitative metrics of evaluation are *empirical* and *experimental.* However, there are some exceptions. These occur whenever the transitions can be modeled using the underlying stochastic process. In this connection, the most common tools used involve Random Walks (RWs).

The theory of RWs and its applications have gained an "exponential" amount of research interest since the early part of the last century. From the recorded literature, we see that the pioneering treatment of a uni-dimensional RW was due to Pearson in [16]. The RW is, usually, defined as a trajectory involving a series of successive random steps, which are, quite naturally, modeled using Markov Chains (MCs). MCs are probabilistic structures that possess the so-called "Markov property" – which, informally speaking, implies that the next "state" of the walk depends on the current state and not on the entire past states (or history). The latter property is also referred to as the "lack of memory" property, which imparts to the structure practical consequential implications since it permits the modeler to predict how the chain will behave in the immediate and distant future, and to thus quantify its behavior.

**Applications of RWs:** It would be no exaggeration to state that tens of thousands of papers have been written that either deal with the analysis of RWs or their applications. Embarking on a comprehensive survey would thus be meaningless. In all brevity, we mention that RWs have been utilized in a myriad of applications stemming from areas as diverse as biology, computer science, economics and physics. For instance, concrete examples of these applications in biology are the epidemic models described in [3], the Wright-Fisher model, and the Moran Model in [12] etc... RWs arise in the modeling and analysis of queuing systems [9], ruin problems [18], risk theory [15], and sequential analysis and learning theory as demonstrated in [5]. In addition to the above-mentioned *classical* application of RWs, recent applications include mobility models in mobile networks collaborative recommendation systems [7], web search algorithms [1], and reliability theory for both software/hardware components [4].

RWs are traditionally considered to be uni-dimensional (i.e., on the line). However, multi-dimensional RWs, that operate in a higher dimensional space, have been studied by generalizing their uni-dimensional counterparts.

**Analysis of Ergodic RWs on Trees:** Although RWs with with transitions on a line, such as the gambler's ruin problem, have been extensively studied for almost a century, as one can observe from [6], that problems involving the analysis of RWs on a tree are intrinsically hard and have received little research attention. This is because they involves the hardest concepts of two arenas: Firstly, they involve specific Random Walks with Jumps (RWJs), where the transitions are to non-neighbor states. Secondly, the non-neighbor states have an additional constraint in that they are associated with an underlying tree structure, as opposed to a line. In this paper, we consider the analysis of one such fascinating RWJ. Although the analysis is seemingly impossible, we shall show that because of the nature of the tree-based transitions, the phenomenon of time reversibility can be invoked, and thus the analysis can be achieved.

**Application of our tree-based RW:** Although the analysis of our chain is pioneering and is a contribution in its own right, it turns out that our RW is a generalization of a dichotomous search when the feedback about the direction of the search is faulty or erroneous. In fact, typical classically-known dichotomous search schemes are deterministic in the sense that they work with non-faulty feedback. Thus, it is possible to eliminate a whole subtree at each iteration, effectively shrinking the search space by half, until the target node is discovered. However, under faulty feedback possessing a stochastic nature, the most intuitive way to solve the problem is to perform a large enough number of tests at each level of the tree before deciding whether to eliminate either the left or right subtree, and to then proceed iteratively until the deepest level is reached [2]. Unfortunately, however, if by mistake a subtree that contains the target node is eliminated the whole search process is "misled". More precisely, schemes such as those reported in [2] are prone to error[1].

How then do we circumvent the problem alluded to above? In our solution, we propose that the RW does "backward" transitions in the tree so that to avoid the problem of sampling with a large number of tests at each level. In addition, the RW is made "non-absorbing" so that to avoid getting trapped into a subtree that does not include the target node. All of these concepts are explained below.

We believe that the current RW analyzed in this paper is important in its own right since it is known that a dichotomous search of a sorted list is an order of magnitude faster than a sequential search. One should further note that any line structure can be easily mapped into a tree structure if the nodes on the tree have some inherent ordering between themselves.

**History of tree-based RWs:** To present the results of this paper in the right perspective, we mention that the problem of considering RWs on a tree for parameter optimization [13] was first introduced by the authors of the current paper in [20] and extended by Zhang *et al.* [22] to the case of symmetrical environments.

[1] These algorithms can be designed to yield a high probabilistic guarantee of reaching the target. To do this, they formulate a limiting assumption: In order to choose a number of tests at each level that is large enough, one needs to know the exact probability by which the environment suggests the right direction of the search. In this paper, we assume that the latter probability is unknown and that the only known clue is that this quantity has a lower bound of $\frac{1}{2}$. The case when the probability of correct transition is less than $\frac{1}{2}$ is referred to as the symmetric environment [22].

The current work is a generalization of our previous work [20] as we derive a general result which does not involve the golden ratio condition as a lower bound on the transition probabilities along the optimal path in the tree, but rather the more intuitively appealing ratio, namely $\frac{1}{2}$. It can thus be applied in many random search problems. Furthermore, the transitions of the RW analyzed here are different from the one proposed in [20] as the only allowed self-transitions take place at the leaves while no self transition takes place at the root node.

The design of RW schemes can be rendered more useful in designing engineering applications involving some dichotomous faulty feedback responses. Finally, and most importantly, the current RW on a tree has a very genuine interpretation as a form for dichotomous search with faulty feedback.

The results of the paper are both deeply theoretical and quite applied. Clearly, all of these aspects cannot be included in a single conference publication. The intricate mathematical details of the analysis and additional experimental results are included in [21].

### 1.1 Organization of This Paper

After having introduced the problem in Sect. 1, we proceed in Sect. 2 to introduce the powerful concept of Time Reversible MCs which makes our analysis possible. In Sect. 3, we describe the RW and also present some analytical results. Section 4 contains experimental results that support our theoretical results which include steady-state results and simulation results for the chain's transient behavior.

## 2 Time Reversible Markov Chains

A fundamental contribution of this paper is the analysis of a RW that is superimposed on a tree structure using the concepts of time reversibility. This model can be see as a generalization of a deterministic dichotomous search. Since this is crucial to this paper, this phenomenon is briefly surveyed here. Some Markov chains have the property that the process behaves in just the same way regardless of whether time is measured forwards or backwards. Kelly [11] made an analogy saying that "if we take a film of such a process and then run the film backwards, the resulting process will be statistically indistinguishable from the original process." This property is described formally in the following definition.

**Definition.** A stochastic process $X(t)$ is time reversible if a sequence of states $(X(t_1), X(t_2), \ldots, X(t_n))$ has the same distribution as the reversed sequence $(X(t_n), X(t_{n-1}), \ldots, X(t_1))$ for all $t_1, t_2, \ldots, t_n$. □

Consider a stationary Ergodic Markov chain (that is, a Markov chain that has been in operation for a long time) having transition probabilities $M_{st}$ and stationary probabilities $P\{\pi_s\}$. Suppose that starting at some time we trace the sequence of states going backwards in time. That is, starting at time $t$, consider the sequence of states $X_t$, $X_{t-1}$, $X_{t-2}$, $\ldots$ $X_0$, It turns out that this sequence of states is itself a Markov chain with transition probabilities $Q_{st} =$

$(P\{\pi_t\}/P\{\pi_s\}) * M_{ts}$. If $Q_{st} = M_{st}$ for all $s, t$, then the Markov chain is said to be *time reversible.* Note that the condition for time reversibility, namely $Q_{st} = M_{st}$, can also be expressed as

$$P\{\pi_s\} M_{st} = P\{\pi_t\} M_{ts} \qquad \text{for all} \quad s \neq t. \tag{1}$$

The condition in the above equation can be stated as, for all states $s$ and $t$, the rate at which the process goes from $s$ to $t$ (namely $P\{\pi_s\} M_{st}$) is equal to the rate at which the process goes from $t$ to $s$ (namely $P\{\pi_t\} M_{ts}$). It is worth noting that this is an obvious necessary condition for time reversibility since a transition from $s$ to $t$ going backward in time is equivalent to a transition from $t$ to $s$ going forward in time. Thus, if $\pi_m = s$ and $\pi_{m-1} = t$, then a transition from $s$ to $t$ is observed if we are looking backward, and one from $t$ to $s$ if we are looking forward in time. The following theorem adapted from Ross and used universally [10,11,17] gives the necessary and sufficient condition for a finite ergodic Markov chain to be time reversible. The proof of the theorem can be found in [17, p. 143].

**Theorem 1.** *A finite ergodic Markov chain for which $M_{st} = 0$ whenever $M_{ts} = 0$ is time reversible if and only if starting in state s, any path back to s has the same probability as the reversed path. That is, if*

$$M_{s,s_1} M_{s_1,s_2} \ldots M_{s_k,s} = M_{s,s_k} M_{s_k,s_{k-1}} \ldots M_{s_1,s}$$

*for all states $s, s_1, \ldots, s_k$.* □

Using the above theorem, we state the result that any tree structure associated with a finite stationary Markov process is time reversible. This follows from the avenue that a Markov chain resulting from the “transition” operations on any tree structure is time reversible. In fact, this result is not totally new. Kelly (see [11, p. 9]) proved the following lemma.

**Lemma** (Adapted from Kelly [11]). *If the graph G associated with a stationary Markov process is a tree, then the process is time reversible.* □

Although Kelly reported this result, he did not demonstrate how to associate a tree with a stationary Markov chain. In this paper, we shall give a formal definition for one such tree structure by organizing the points on the line along a tree and prove the corresponding theorem regarding its time reversibility. The application of time reversibility in the domain of self-organizing lists has been reported elsewhere [14].

## 3 The Random Walk on Tree

The space of the search is arranged in the form of a binary tree with depth $D = log_2(N)$, where $N$ is an integer. The Random Walker (RW) searches for the target leaf node by orchestrating a controlled random walk on a tree. We assume the existence of an "Oracle", also referred to as the Environment, $\Xi$, which informs the RW, possibly erroneously (i.e., w. p. $p$), which way it should move to reach the target node. We emphasize, however, that we could have defined the RW without even invoking the concept of the "Oracle" or Environment, $\Xi$, since it is merely a fictitious concept introduced to ease the understanding and readability of the paper. In addition, it emphasizes the analogy with dichotomous search as we can see the Environment as a type of faulty Oracle as opposite to a "perfect" Oracle, as in a classical dichotomous search.

### 3.1 Definitions

**Construction of hierarchy.** The search space is constructed as follows: First of all, the hierarchy is organized as a balanced binary tree with maximal depth $D$. For convenience, we will use the same notation adopted in [8,20] and index the nodes using both their depth in the tree and their relative order with respect to the nodes located at the same the depth.

**Root node.** The hierarchy root (at depth 0), which we call $S_{\{0,1\}}$.

**Nodes at depth d.** Node $j \in \{1,...,2^d\}$ at depth $d$, called $S_{\{d,j\}}$, where $0 < d < D$ which has two children nodes $S_{\{d+1,2j-1\}}$ and $S_{\{d+1,2j\}}$. We say that $S_{\{d+1,2j-1\}}$ is the *Left Child* of $S_{\{d,j\}}$ and $S_{\{d+1,2j\}}$ is its *Right Child.*

**Nodes at depth D.** At depth $D$, which represents the maximal depth of the tree, the nodes do not have children.

**Convention regarding the Leaves' notation.** In a same vein, since level "$D+1$" is nonexistent, we use the convention that *Right Child* of a leaf node is the same as the leaf node in question itself. Similarly, the *Left Child* of a leaf node is the leaf node itself. Formally, we say that:

$$\textit{Left Child}(S_{\{D,j\}}) = \textit{Right Child}(S_{\{D,j\}}) = S_{\{D,j\}} \text{ for } j \in \{1,...,2^D\}.$$

**Target Node.** The **target node** is a unique leaf node. We will later show that by imposing a simple condition on the "effectiveness" of the Environment (a concept defined presently), that we can concentrate the RW to be arbitrarily close to the target node.

**Non-Target Node.** They are leaf nodes different from the target node.

**Effectiveness of the Walker.** The probability to make a transition along the shortest path to the target node is denoted $p$ and is a constant. Effectively, this probability can be seen as the probability of the RW making a "correct" movement towards the target node. However, whenever the random walker is situated at the target node itself, the effectiveness translates into the probability of it staying in the target node, i.e., this is represented by the probability of self-transition.

### 3.2 Structure of the Search Space

We intend to organize the search space in the form of a balanced binary tree. The random walker searches for the target node by operating a random walk on the tree, moving from one tree node to another. At any given time instance, the RW finds itself at a node $S_{\{d,j\}}$ in the tree, where $j \in \{1, \ldots, 2^d\}$ and $0 \leq d \leq D$. The Environment $\Xi$ essentially instructs the RW, possibly erroneously (i.e., with probability $p$), which way it should move to reach the target node.

- *Reverse Transitions*: Transitions of this type correspond to a movement to a lower level in the hierarchy. This happens when the RW moves to the immediate *Parent* which allows the RW to escape from getting trapped in a wrong subtree, i.e. one that does not contain target node.
- *Top-down Transitions*: Transitions of this type correspond to a movement to a deeper level in the hierarchy. The RW performs a transition to a deeper level in the hierarchy by choosing a *Child* node which is, hopefully, contained in a subtree that contains the target node.
- The only reflexive transitions (self-transitions) are allowed at the leaf node. We emphasize that we do not allow a reflexive transition in the root node as done in our previous work [20].

This concludes the description of our RW. We shall now investigate its convergence properties.

### 3.3 Analysis of the Solution

In this section, we shall prove that the RW is asymptotically optimal. We shall show that, eventually, based on an informed series of guesses, the RW will be able to concentrate its moves within nodes in the tree that are associated with the target node, and this will be true if $p$ is larger than 0.5.

**Theorem 1.** *We suppose that the effectiveness $p$ of the environment is strictly larger than* 0.5. *Let $S_{\{D,j^*_D\}}$ be the target node. Formally, $Lim_{D \to \infty} \pi_{\{D,j^*_D\}} \to 1$.*

Due to space limitations, the proof of Theorem 1 is omitted. It is found in [21].

## 4 Simulation Results

In this section, we report some representative experimental results submitted to confirm the validity of the theoretical results we have obtained. Since the convergence is asymptotic as the number of levels $D$ tends to infinity, and due to the consequential explosion of the memory requirements, we were not able to perform simulations for large values of the tree depth $D$, i.e., those that can be considered to be large enough to be seen to tend to "infinity". In order to counter this limitation, we defined a neighborhood $N^*(i)$ as the $i^{th}$ neighborhood of the target node which simply means all nodes that are located at the subtree rooted at node $\pi_{\{D-i,j^*_{D-i}\}}$. Observe that the target node belongs to subtree rooted at node $S_{\{D-i, j^*_{D-i}\}}$, and that the latter subtree is composed of $i$ levels. Thus, we have, instead, reported the steady state probability of the neighborhood $N^*(i)$ which, informally, is the sum of the steady state probabilities of the nodes in $N^*(i)$, i.e., belonging to the subtree rooted at node $S_{\{D-i, j^*_{D-i}\}}$.

### 4.1 Empirical Verification of the Optimality

In this section, we report the steady probability in $N^*(3)$ using simulation, by running the random walk for $10^7$ iterations. We varied the number of nodes in the tree $D$ from 8 to 12, and varied the effectiveness of the environment $p$ using the following three values: $p = 0.7$, $p = 0.85$ and $p = 0.95$. In intermediate nodes, the RW is always faced with three transition alternatives (parent, right child or left child). The Oracle/Environment suggests the correct transition with the probability $p$. To further enrich the experiments, we performed experiments with different ways to "unevenly" distribute the remaining probability $1 - p$ among the two incorrect transitions at the intermediate nodes. However, we discovered that distributing this probability mass "evenly" or "unevenly" (for example, with probabilities $\frac{1}{3}$ and $\frac{2}{3}$) had no effect on the convergence results[2]. We thus chose to "evenly" divide the remaining probability $1 - p$ in between the two incorrect transitions, i.e., each incorrect transition occurs with the probability $\frac{1-p}{2}$ in the intermediate nodes. More specifically, if we suppose, for example, that at an intermediate node, the correct search direction is a transition to the parent, i.e., to do a backtracking operation, then the transition to the left child and right child will both takes place with probability $\frac{1-p}{2}$, while the transition to the parent will occur with probability $p$. In all the experiments that we report the results for, we chose the target node to be the rightmost node in the hierarchy, i.e., node $S_{\{D, 2^D\}}$. As expected, from Table 1, we observe that the values of the steady state probability in $N^*(3)$ approached 1. The latter value increased with $p$ (which represented the quality of the Oracle/Environment, and it attained its maximum values for $p = 0.95$. In fact, for $p = 0.95$, the steady probability was

**Table 1.** Convergence to the neighborhood $N^*(3)$ for various values of $D$ and $p$.

| $D$ | $p = 0.7$ | $p = 0.85$ | $p = 0.95$ |
|---|---|---|---|
| 8 | 0.9530518 | 0.9922004 | 0.9993106 |
| 9 | 0.9527884 | 0.9922102 | 0.9993274 |
| 10 | 0.9532058 | 0.9920532 | 0.9993156 |
| 11 | 0.9526474 | 0.9922458 | 0.9992978 |
| 12 | 0.9529796 | 0.9921286 | 0.9999534 |

[2] This is also in line with the theoretical results where we did not impose any condition on distributing $1 - p$. Thus, for the sake of brevity, we will not report those experiments with "uneven" distributions of $1 - p$ over the incorrect alternatives, and limit the reported results to the case of "even" distributions, i.e., $\frac{1-p}{2}$.

greater than 0.999 for all the different values of $D$. Similarly, for $p = 0.85$, the steady probability exceeded 0.99 for all the different values of $D$.

## 5 Conclusions

This paper introduces a novel mathematical tool that can be used for the analysis of AI algorithms. It concerns the field of Random Walks (RWs), typically used to model birth and death processes, renewal processes and the gambler's ruin problem. RWs have been studied for more than a century, and they have been primarily analyzed for walks on a *line*. RWs for two and three dimensions, have been modeled by enforcing the random steps to be made to the corresponding neighboring positions in one or multiple dimensions. RWs typically operate "on a discretized line" by forcing the walker to perform small steps to the current-state's neighbor states, rendering the analysis tractable. When any of the transitions are to non-neighbour states (referred to as "jumps" as opposed to steps), a formal analysis of the RW is, typically, untractable, except in a very few cases such as the one examined in [19]. This paper concerns the case when the jumps are performed as though a tree is superimposed on the line, and the jumps are to the children or the parent of the node where the walker is. RWs on a tree entail moving to *non-neighbor* states in the space, which makes the analysis involved, and in many cases, impossible. However, in this case, we demonstrate that an analysis of the chain is feasible because we can invoke the phenomenon of "time reversibility". We, however, permit the operation of "backtracking", and interestingly, it is precisely these "backtracking" transitions that naturally render the chain to be "time reversible". Our analysis is possible because we have used the latter property of the chain. Further, this RW turns out to be a type of generalization of dichotomous search with faulty feedback, which has numerous real-life applications.

The paper has also listed the chain's fascinating limiting theoretical properties. Besides, it contains simulation results that justify the chain's analytic steady-state and transient characteristics.

## References

1. Altman, A., Tennenholtz, M.: Ranking systems: the pagerank axioms. In: EC 2005: Proceedings of the 6th ACM Conference on Electronic Commerce, pp. 1–8. ACM, New York (2005)
2. Ben Or, M., Hassidim, A.: The bayesian learner is optimal for noisy binary search (and pretty good for quantum as well). In: IEEE 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, pp. 221–230. IEEE (2008)
3. Berg, H.C.: Random Walks in Biology, Revised edn. Princeton University Press, Princeton (1993)
4. Bishop, P.G., Pullen, F.D.: A random walk through software reliability theory. Math. Struct. Softw. Eng., 83–111 (1991)
5. Bower, G.H.: A turning point in mathematical learning theory. Psychol. Rev. **101**(2), 290–300 (1994)

6. Feller, W.: An Introduction to Probability Theory and Its Applications, vol. 1, 3rd edn. Wiley, New York (1968)
7. Fouss, F., Pirotte, A., Renders, J.M., Saerens, M.: Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Trans. Knowl. Data Eng. **19**(3), 355–369 (2007)
8. Granmo, O.C., Oommen, B.J.: Solving stochastic nonlinear resource allocation problems using a hierarchy of twofold resource allocation automata. IEEE Trans. Comput. **59**, 545–560 (2009)
9. Gross, D., Harris, C.M.: Fundamentals of Queueing Theory (Wiley Series in Probability and Statistics). Wiley-Interscience, New York (1998)
10. Karlin, S., Taylot, H.: A First Course in Stochastic Processes. Academic Press, New York (1975)
11. Kelly, F.: Reversibility and Stochastic Networks. Wiley Series in Probability and Mathematical Statistics. Tracts on Probability and Statistics, Wiley, Chichester (1987)
12. Nowak, M.A.: Evolutionary Dynamics: Exploring the Equations of Life. Belknap Press of Harvard University Press, Cambridge (2006)
13. Oommen, B.J.: Stochastic searching on the line and its applications to parameter learning in nonlinear optimization. IEEE Trans. Syst. Man Cybern. Part B: Cybern. **27**(4), 733–739 (1997)
14. Oommen, J., Dong, J.: Generalized swap-with-parent schemes for self-organizing sequential linear lists. In: Leong, H.W., Imai, H., Jain, S. (eds.) ISAAC 1997. LNCS, vol. 1350, pp. 414–423. Springer, Heidelberg (1997). doi:10.1007/3-540-63890-3_44
15. Paulsen, J.: Ruin theory with compounding assets - a survey. Insur. Math. Econ. **22**(1), 3–16 (1998). Special issue on the interplay between insurance, finance and control
16. Pearson, K.: The problem of the random walk. Nature **72**(1867), 342 (1905). http://dx.doi.org/10.1038/072342a0
17. Ross, S.: Introduction to Probability Models. Academic Press, New York (1980)
18. Takacs, L.: On the classical ruin problems. J. Am. Stat. Assoc. **64**(327), 889–906 (1969)
19. Yazidi, A., Granmo, O.C., Oommen, B.J.: On the analysis of a random interleaving walk-jump process with applications to testing. Sequential Anal. **30**(4), 457–478 (2011)
20. Yazidi, A., Granmo, O.C., Oommen, B.J., Goodwin, M.: A novel strategy for solving the stochastic point location problem using a hierarchical searching scheme. IEEE Trans. Cybern. **44**(11), 2202–2220 (2014)
21. Yazidi, A., Oommen, B.J.: On the analysis of a random walk-jump chain with tree-based transitions, and its applications to faulty dichotomous search. Unabridged version of this paper (2016). To be submitted for publication
22. Zhang, J., Wang, Y., Wang, C., Zhou, M.: Symmetrical hierarchical stochastic searching on the line in informative and deceptive environments. To appear in IEEE Trans. Cybern. (2016)

# Travel Time Prediction for Trams in Warsaw

Adam Zychowski[(✉)], Konstanty Junosza-Szaniawski, and Aleksander Kosicki

Faculty of Mathematics and Information Science, Warsaw University of Technology,
Koszykowa 75, 00-662 Warsaw, Poland
a.zychowski@mini.pw.edu.pl

**Abstract.** The paper presents a comparison between different prediction methods for trams time travels in Warsaw. Predictions are constructed based on historical trams GPS positions. Three different prediction approaches were implemented and compared with the official timetables and real time travels. Obtained results show that the official timetables provides only approximated time travel especially in rush hours. Proposed prediction methods outperform the official schedule in the term of time travel precision and may be used as a more accurate source of travel time for passengers.

**Keywords:** Transportation · Travel time prediction · Neural network · Decision support system

## 1 Introduction

Time travel prediction is an interesting problem due to its complexity, many real life applications and everyday usability. Planned schedules are no longer enough. Passengers expect more precise information about vehicles arrival time and trip time length. On the other hand, real travel time depends on many factors such as weather conditions, traffic lights, road accidents, number of passengers, traffic flow, driver emotional state or roadworks. All of these parameters are difficult to predict or even measure. However, past researches show that using some additional information, for instance vehicles' GPS locations over a day or data collected by automatic passenger counters, some reasonable predictions could be obtained and their accuracy outperforms planned schedule. In recent years a growing interest in applying smart algorithms to travel time prediction has been observed, mainly due to its practical applicability in real life domains and at the same time an intrinsic complexity which makes the problem challenging. In effect, numerous new approaches to time travel prediction relying mainly on various statistical models or machine learning techniques were proposed. All methods could be divided into four groups: simple models based on the historical data,

The research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688380.

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_6

statistical models, machine learning approaches and hybrid models. First group contains methods based on observation that usually time travels are repeatable between days. Road conditions from previous days can be a reasonable forecast of future condition at the same time of the day and the same day of the week [14]. Some of proposed methods in this group rely on average travel time in previous days [11] and others use average speed as a prediction factor [13]. Methods in statistical model group used several identified factors (i.e. traffic lights, weather) as independent variables and make prediction based on their statistic distributions and correlations. Time series models [5], regression models [9] or Kalman Filters [3,16] are the most common examples in this group. Third category of travel time prediction is machine learning models. They perform some learning process on existing data to find an answer for unknown input data. They have good results with huge volume of information, non-linear relationships or noisy information. The most popular techniques are Artificial Neural Networks [4,6] and Support Vector Machines [15]. Also some hybrid models [8,12] were used to solve time travel prediction problem. In this group some combination from previously presented methods are combined. Increasing trend to utilize hybrid algorithms to improve the prediction accuracy could be observed in recent years. We refer to [1] for a more in-depth overview and detailed description of the respective methods. This paper presents comparison between travel time prediction methods for trams in Warsaw. Predictions are computed based on vehicles' GPS locations and the official timetables. Performed experiments show usability level of used methods for real life trip planning including vehicles changes. The remainder of this paper is arranged as follows. Section 2 presents overview of tram's infrastructure in Warsaw and shows used data sources, their quality problems and proposed solutions. In Sect. 3 detailed descriptions of implemented methods are provided. Experiment result are presented in Sect. 4. Last section is devoted to conclusions and directions for future research.

## 2 Warsaw Trams Infrastructure and Data Sources

Warsaw is the capital city of Poland with population estimated at 1.744 million residents [2]. Nearly 82% people travel every day and 47% of them use public transport as the main travel method [10]. The available means of transport in Warsaw are buses, trams, trains and subway. This paper considers only trams because at the moment of writing this paper travel agency provides complete information (GPS locations for all vehicles) only for trams. Warsaw trams infrastructure contains 26 trams lines with about 240 stops. Tram drivers are obligated to stop on all stops on planned route (no request stops). There are no night or special lines. Services for all lines run every day. The earliest tram departures at 3:30 AM and the latest arrives to the depot at about 1:30 AM. Three schedules versions are used during a week: weekday schedule (from Monday to Friday), Saturday schedule, and Sunday and holiday schedule. Two main data sources in this research are used. The first one is information about the official schedules planned by Warsaw Public Transport Authority. This source contains stops locations, routes and lines on stop planned times. For each stop time also brigade number is provided. This number, combined with line number, is unique identificator for vehicle. Data from this source are reliable, because it is used every day by travel agency workers to assign vehicles to given routes. Second source used in this paper is real trams' positions form GPS transmitters placed in vehicles. Every tram sends its current position with about 15 s frequency. They are available in files with information about line, brigade, log time and coordinates - one file for each day. Table 1 presents some computed properties for this data from 22nd September 2016. There are some quality issues such as trams positions outside Warsaw territory, vehicles disappears in the middle of the route, logs duplicates or abnormal tram's routes. The reason for most of them is probably wrong position sent by GPS transmitter. To make reliable predictions (and also get real tram on stop time) for these data some quality improvement actions were performed. All logs with coordinates outside Warsaw territory and duplicates were removed. Also average speed between all pairs of consecutive logs was computed and if this speed is greater than 100 km/h one of these logs are deleted. Also if distance between two consecutive logs was less than 2 m, it was recognized as GPS error and second log's coordinates were changed to the previous log's coordinates. Above mentioned modifications affected about 15% of original data. One of the first challenge was precisely estimating real time of arriving tram on a stop from GPS data. It is essential component for all methods to determine tram delays and compare predicted time with real tram on stop arrival time. Available data are stop's coordinates and tram coordinates with time from GPS transmitter. The most natural way to get tram to stop arrival time is to define a radius $r$ and when distance between the stop and the tram is less than $r$ it means that the tram is on the stop at that moment. However, in Sect. 2 it was showed that average time between subsequent GPS logs is 18.2 s and average distance that tram covers in this time is about 71 m. It means that $r$ should be at least 36 m, because otherwise tram may log its position before and after $r$ meters from the stop and algorithm will miss the stop. Moreover, this approach is too imprecise, because it does not provide exact tram's arrival time, but time that the tram is at most $r$ meters from the stop. For instance, traffic lights in near distance before stop may cause few minutes difference between real and computed arrival time. Thus, it was necessary to find another way. Figure 1 presents proposed method for computing tram on stop time based only on distances between GPS logs coordinates and stop position. At the beginning first ($A$) and last ($B$) tram's logs coordinates in the stop's ($S$) neighborhood (defined as a circle with given radius and center in considered stop's position) are obtained. This allows determining from what direction tram arrived to stop and what direction it moved after that. Based on this directions bisector of angle $ASB$ is created. Let call it *arrival line*. The moment of crossing that line is equivalent to the moment of visiting the stop by the tram. Time of first log after arrival line crossing may be quite imprecise because tram could move quite far from the stop at that moment. Better option is to assume that trams speed is constant and get time difference between last log before and first log after crossing the line and compute the moment of tram stop arrival proportionally to these logs distances from the stop. In experiments neighborhood radius was set

**Table 1.** Basic parameters of received real trams positions.

| Parameter | Value |
|---|---|
| All logs | 993263 |
| Unique logs (unique triples [time, line, brigade]) | 861054 (86.7%) |
| Logs from Warsaw territory | 991639 (99.84%) |
| Average time between logs from the same tram | 18.2 s |
| Average distance traveled between logs | 71.12 m |
| Unique pairs: [line, brigade] | 305 |
| Unique pairs: [line, brigade] in schedule | 375 |

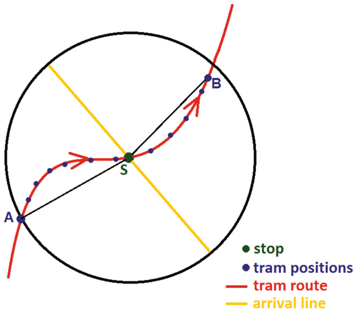


**Fig. 1.** Graphical illustration of method for determining tram to stop arrival time.

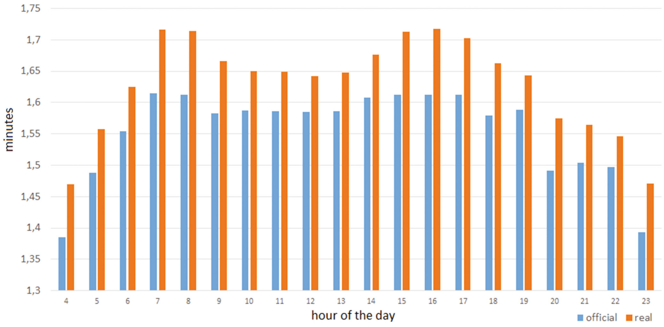


**Fig. 2.** Comparison between average time between all stops pairs by hour of the days for the official schedule and real travels.

to 300 m. Figure 2 provides average time between all stops pairs during the day obtained from the official timetable and computed form real trams positions. All values are greater in real data which means that the official schedule is imprecise and may be improved. Furthermore, morning (from 7 to 9) and afternoon (from 15 to 18) rush hours could be easily noticed. The official timetable provides longer travel times for those hours, however in reality differences between those hours and the others are greater.

## 3 Prediction Methods

This section provides detailed description of implemented trams travel time prediction methods. Three approaches are chosen for comparison: *current delay propagation*, *historical average time travel* and *artificial neural network model.* To simulate real time prediction arbitrary day $d$ and time $t$ are selected and available data in prediction process are: the official timetables for day $d$ and all days before that day, trams' GPS positions before day $d$ and for day $d$ until time $t$. Based on this data prediction for next 2 h for all trams is made.

### 3.1 Current Delay Propagation

One of the most natural way to make real time prediction for tram on stop arrival time may be finding current vehicle delay and propagate it to next stops on planned route. This simple idea is base for first proposed method - *current delay propagation.* In this approach for all trams independently last visited stop $s$ before time $t$ is determined based on vehicles' GPS positions. Current delay $d$ is difference between computed arrival time for the stop $s$ and time from the official timetable (i.e. time that this vehicle was planned to arrive to the stop $s$). Then, for all non-visited stops on considered vehicle's route, predicted arrival time is planned time from the official schedule summed with computed in previous step delay $d$. This method is very simple and does not use any historical data. It may give good result if the delay reason for the vehicle is a single accident (for example too late depot leave or congestion on traffic lights) and do not repeat until the end of route. However, this method is not able to predict future problems or delay changes, which is important disadvantage. It is expected to obtain better results for vehicles near the end of its route than for vehicles which recently leave the depot and probably their current delay is low. Furthermore, for vehicles that do not start service before prediction time $t$, this approach is not able to make any prediction and it just returns time from the official timetable.

### 3.2 Historical Average Time Travel

The second approach is modification of previous method. Basically, it is intended to reduce first method disadvantages, i.e. take into consideration historical time travels or usual road fraction traffic conditions. The main idea is to use average time travel from previous couple of days as an estimator for current day times. One of the main factors which has big impact on time travels is time of the day. Thus, to make predictions more accurate computed average times should depend on start travel time. It also may be better to consider the same kinds of days (week days, weekends) together, because road traffic differs from each other. *Historical average time travel* method same as previous method searches for last visited stop $s$ based on GPS positions. Then, iteratively for all stops from $s$ to the end of route, historical average time between consecutive stop pairs is added, i.e. in first step historical average time between the stop $s$ and the next stop is added to the arrival time to the stop $s$ computed based on GPS logs, in next step historical average time is added to arrival time predicted in previous step etc. Historical average time mentioned above is computed based on real times for the same days of week in last 30 days from all travels between considered stops pair at given hour. Only historical travel times for the same time of the day (the same hour) as prediction time of the day are included to computation.

### 3.3 Neural Network Model

Third approach for trams time travel prediction is machine learning technique and it uses artificial neural networks approach. In recent years neural networks gains popularity very fast in lots of domains, due to its ability to solve complex non-linear problems. Many different kinds of neural network were described in literature. In this paper one of the first and the most simple neural network architecture is used - multilayer perceptron (MLP) with backpropagation learning method. The main idea behind MLP is to learn some relationships between given examples in form of input and expected output data. Learning process is realized by changing weights between neurons according to some principle. Then MLP is able to correctly give information (make prediction) about unseen example. Neural networks can generalize information, ignore noises and identify underlying relationships even if they are hard to explain. For more details about neural networks, MLP architecture or backpropagation learning process we refer to [7]. In this paper separate neural network for each route is created. The output value is predicted tram's time travel from the stop $s_i$ to the stop $s_j$. As input four values were chosen:

- order number of stop $s_i$ in route,
- order number of stop $s_j$ in route,
- travel time from route's start (first stop in route) to stop $s_i$,
- prediction time of the day.

Third information (travel time from route's start to stop $s_i$) is computed based on GPS trams' positions. It provides information how trip is realized so far. It depends on actual traffic conditions, so it contains important information for time travel prediction. Time of the day (the last input information) also has influence on time travel, because traffic changes during the day. Based on performed experiments a one-hidden layer perceptron with 40 neurons in hidden layer was chosen. The learning rate was set to 0.03, the momentum to 0.9 and the number of training epochs was equal to 250. Figure 3 shows used neural network architecture. Learning data set contains information for the same days of week in last 30 days from all travels on given route. To make prediction for particular tram at given time of the day $t$ last visited stop $s$ before $t$ is determined based on vehicles' GPS position and travel time $x$ from route beginning to stop $s$ is computed. To predict arrival time for all non-visited stops in route learnt neural network is used, i.e. for stop $s_j$ arrival time is computed by adding to $t$ output value of neural network with input values: stop's $s$ order number in route, stop's $s_j$ order number in route, $x$ and $t$.

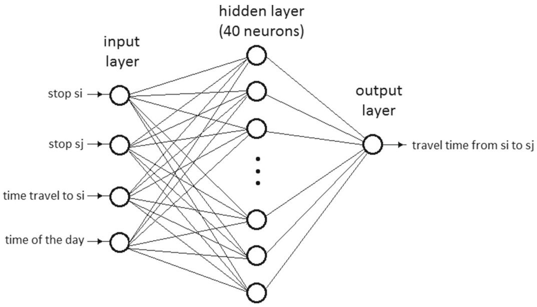


**Fig. 3.** Proposed neural network architecture.

## 4 Experimental Results

All methods presented in previous section were previously used in other researches to predict bus time travels. However, to the best of our knowledge, this paper presents the first experimental results for trams travels prediction. Also, unlike most of papers in this domain, performed experiments do not focus on single trips and verify prediction independently for different vehicles. Presented results take into consideration trams changes. It simulates passenger view, for whom the most important parameter is to know accurately how long trip form particular stop to another chosen stop takes (including all necessary changes). It is assumed that total trip time includes also time spend waiting on the start stop for the first tram. For the purposes of tests a sample of 1000 random tram stop pairs has been selected. Loop stops and pairs of stops closer to each other than 5000 m (in the straight line) has been excluded from the drawing. The pairs has been then used to compute actual trip itineraries on 22 September 2016 at four different hours (8:00, 12:00, 16:00 and 20:00) with five different timetables. Three of the timetables were generated based on three different prediction models, one was the official Warsaw Public Transport Authority timetable and one was computed based on the historical GPS records (real schedule). This setup yielded total of 20,000 tram trips itineraries that were later used for the analysis. The actual timetables were created in the General Transit Feed Specification (GTFS) format. That included both conversion of the official timetables to the GTFS format as well as creating the GTFS files from scratch for the prediction models. Trips itineraries were computed with the Open Trip Planner server instance. Default graph search timeouts were turned off and so all obtained results were fully deterministic. The most important measure to compare proposed prediction methods is difference between the real travel time and predicted travel time which can be called prediction error. This measure shows method's accuracy. Table 2 shows comparison between prediction methods and the official timetables by this measure for tested times of the day. In all cases *historical average time travel* approach yielded the best results. Average absolute prediction error is about 2 min. Average planned time travel was 45 min, so prediction error is

**Table 2.** Comparison between prediction methods in terms of average absolute differences (in seconds) with real travel times (Diff) and percentage of number of times given method obtained the best result (Best). Best results for each time of the day are bolded.

| | 08:00 | | 12:00 | | 16:00 | | 20:00 | | All | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Diff | Best | Diff | Best | Diff | Best | Diff | Best | Diff | Best |
| Official timetables | 180.53 | 36% | 172.30 | 29% | 203.61 | 25% | 166.75 | 29% | 180.79 | 30% |
| Current delay prop | 178.41 | 30% | 118.94 | 34% | 172.99 | 27% | 98.33 | 33% | 142.17 | 31% |
| Historical avg travel | **159.70** | 37% | **104.80** | **41%** | **137.83** | 40% | **89.25** | **49%** | **122.89** | **42%** |
| Neural network | 161.70 | **42%** | 138.40 | 38% | 149.92 | **44%** | 143.01 | 40% | 148.25 | 41% |

equal 4%. Worst results were obtained by the official timetables - more than 3 min. Notice also that in rush hours (8:00 and 16:00) all prediction errors gave worse results than for the others tested hours (12:00 and 20:00). Thus, travel time prediction in rush hours seems to be more difficult problem. *Neural network model* obtained nearly as good results as the best method for rush hours. *Neural network model* also is the best method for rush hours in the terms of number of travels for which method obtained the lowest absolute prediction error (42% and 44% travels).

For all methods predicted time travel is more often shorter than real time travel. This difference was the greatest for *current delay propagation* method (nearly 2 times often predicted time was shorter) and the lowest for *neural network model* (only 2% less longer predicted time travels). In performed experiments only time travel from one stop to another was considered (no matter what lines and how many of them), so trams lines in predicted travels may be different than those in real travels. Table 3 provides details about that differences in predicted and real travels. Nearly 75% overall predicted travels were realized by the same trams lines as in reality and 70% also by the same brigades. The better prediction method is (based on results from Table 2) the more travels are realized by the

**Table 3.** Percentage of travels realized with the same lines (Lines) and the same lines and brigades (Brig) as in predicted travel. Best results for each time of the day are bolded.

| | 08:00 | | 12:00 | | 16:00 | | 20:00 | | All | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Lines | Brig | Lines | Brig | Lines | Brig | Lines | Brig | Lines | Brig |
| Official timetables | 64% | 49% | 70% | 61% | 61% | 43% | 69% | 63% | 66% | 54% |
| Current delay prop | 68% | 60% | **81%** | **76%** | 68% | 57% | 82% | 79% | 75% | 68% |
| Historical avg travel | 71% | 62% | **81%** | **76%** | 72% | 62% | **84%** | **81%** | **77%** | **70%** |
| Neural network | **74%** | **66%** | 77% | 73% | **74%** | **66%** | 77% | 74% | 76% | **70%** |

same vehicles as predicted. There is also less trams lines matching in rush hours which confirms greater difficulty for making prediction for these cases.

## 5 Conclusions and Future Work

Three methods for travel times prediction for trams in Warsaw were proposed and tested: *current delay propagation*, *historical average time travel* and *neural network model*. They basic idea is to predict travel time based on current vehicles position (current delay) and historical travels. This data are obtained from trams GPS positions in every 15 s. All proposed methods outperform prediction based on the official timetables. The best method - *historical average time travel* reduced prediction error from 3 min (for the official schedule) to 2 min. Furthermore, results showed clear difference between rush hours (8:00 and 16:00) and the others tested hours (12:00 and 20:00). Time travels are longer in rush hours and more difficult to predict. In this case *neural network model* yielded the best results. This paper presented that even simple methods may improve travel time prediction and may be used to provide more accurate information for passenger. In further steps making comparison between delays and prediction methods during holiday week and normal week is planned. Traffic is observable smaller during school holidays in Warsaw so it would be interesting to check if this difference also could be noticed in time travel predictions. Also making similar researches for buses in Warsaw and comparison them with trams is planned.

## References

1. Altinkaya, M., Zontul, M.: Urban bus arrival time prediction: a review of computational models. Int. J. Recent Technol. Eng. (IJRTE) **2**(4), 164–169 (2013)
2. Central statistical office of Poland: area and population in the territorial profile in 2016. Statistical Information and Elaborations, Warsaw (2016). http://stat.gov.pl/en/topics/population/population/area-and-population-in-the-territorial-profile-in-2016,4,10.html
3. Chen, M., Chien, S.: Dynamic freeway travel time prediction using probe vehicle data: link-based vs. path-based. In: Transportation Research Board 80th Annual Meeting (2001)
4. Chien, S.I.J., Ding, Y., Wei, C.: Dynamic bus arrival time prediction with artificial neural networks. J. Transp. Eng. **128**(5), 429–438 (2002)
5. D'Angelo, M., Al-Deek, H., Wang, M.: Travel-time prediction for freeway corridors. Transp. Res. Rec. J. Transp. Res. Board **1676**, 184–191 (1999)
6. Gurmu, Z.K., Fan, W.D.: Artificial neural network travel time prediction model for buses using only gps data. J. Public Transp. **17**(2), 45–65 (2014)
7. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, Upper Saddle River (1998)
8. Liu, H., van Zuylen, H., van Lint, H., Salomons, M.: Predicting urban arterial travel time with state-space neural networks and kalman filters. Transp. Res. Rec. J. Transp. Res. Board **1968**, 99–108 (2006)
9. Patnaik, J., Chien, S., Bladikas, A.: Estimation of bus arrival times using APC data. J. Public Transp. **7**(1), 1–20 (2004)

10. PBS Sp. z o.o.: Warszawskie badanie ruchu, Cracow University of Technology, Warsaw University of Technology (2015). http://transport.um.warszawa.pl/wbr2015. Accessed 10 Dec 2016
11. Ramakrishna, Y., Ramakrishna, P., Lakshmanan, V., Sivanandan, R.: Bus travel time prediction using gps data. Proceedings Map India (2006)
12. Van Lint, J., Hoogendoorn, S., van Zuylen, H.J.: Accurate freeway travel time prediction with state-space neural networks under missing data. Transp. Res. Part C: Emerg. Technol. **13**(5), 347–369 (2005)
13. Weigang, L., Koendjbiharie, W., de M Juca, R., Yamashita, Y., MacIver, A.: Algorithms for estimating bus arrival times using gps data. In: Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems, pp. 868–873. IEEE (2002)
14. Williams, B.M., Hoel, L.A.: Modeling and forecasting vehicular traffic flow as a seasonal arima process: theoretical basis and empirical results. J. Transp. Eng. **129**(6), 664–672 (2003)
15. Wu, C.H., Ho, J.M., Lee, D.T.: Travel-time prediction with support vector regression. IEEE Trans. Intell. Transp. Syst. **5**(4), 276–281 (2004)
16. Yang, J.S.: Travel time prediction using the gps test vehicle and kalman filtering techniques. In: Proceedings of the 2005, American Control Conference, pp. 2128–2133. IEEE (2005)

# Diagnostic Rule Extraction Using the Dempster-Shafer Theory Extended for Fuzzy Focal Elements

Sebastian Porebski(✉) and Ewa Straszecka

Division of Biomedical Electronics, Faculty of Automatic Control,
Electronics, and Computer Science, Institute of Electronics,
Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
{sebastian.porebski,ewa.straszecka}@polsl.pl

**Abstract.** The Dempster-Shafer theory along with the fuzzy set theory are suitable tools for the medical diagnosis support. They can deal with medical knowledge uncertainty and data imprecision. This paper presents a study of medical knowledge representation by means of the Dempster-Shafer theory extended with the fuzzy set theory and introduces the new rule selection algorithm. The presented method gives an opportunity of interpretable and reliable rule extraction. The method is elaborated and its performance is tested on a popular medical data set. Results show that the presented method can be useful for the knowledge engineer and diagnostician cooperation due to the simple rule base and clear inference method.

**Keywords:** Rule extraction · Dempster-Shafer theory · Fuzzy sets · Medical diagnosis support · Thyroid disease

## 1 Introduction

Medical knowledge extraction is the second most popular area in data mining applications [2]. A diagnosis support tool should deal with the diagnosis uncertainty and medical data imprecision. Simultaneously, the support tool should offer not only diagnosis efficiency but also an inference that is easy to understand for physicians and a reasonable size of extracted knowledge [3]. These are inevitable requirements of a knowledge engineer and a diagnostician cooperation. In medicine even reliable tools can not substitute a diagnostician, they can only support him. The most popular medical knowledge extraction approaches are decision trees [5], support vector machines [4] and neural networks [1]. Their inference parameters (weights or structure) can be translated to the readable IF-THEN rules with additional methods. In this paper the connection of the Dempster-Shafer theory (DST) and the fuzzy set theory (FST) is used to perform diagnostic inference. With the basic probability assignment (critical component of DST) it is possible to represent the knowledge uncertainty. On the other

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_7

hand, FST is used as well-known method to represent data imprecision in many fields. As a result of a study on these methods, new rule extraction algorithm is described. The main requirement in the proposed algorithm is to provide a consistent and easy to interpret diagnostic rule base, also paying attention to its reliability. Experimental results evaluate the approach from the point of view of a diagnostician and knowledge engineer cooperation.

## 2 Medical Diagnosis Support Based on the DST and FST

Medical diagnosis support is usually based on with the IF-THEN rules [2], which shape the inference from symptoms (conditions in a premise) and diagnosis (conclusion). Let us denote $\boldsymbol{x}$ as the medical data case represented with $p$ symptoms:

$$\boldsymbol{x} = [x_1, x_2, ..., x_j, ..., x_p]^\top. \tag{1}$$

A simultaneous use of the DST and FST [8] makes it possible to define the fuzzy focal element, which is a simple or complex logical condition related to the specific medical data symptoms, which can be test results or natural language statements like "*high temperature*" [8]. The simplest focal element considers only one symptom and the most complex focal element considers $p$ symptoms. It is clear that the possible number of fuzzy focal elements for one diagnosis equals $2^p-1$. General form of the fuzzy focal element can be represented in the following way:

$$s^{(r_l)} : X_j \text{ is } A_j^{(r_l)} \wedge \cdots \wedge X_k \text{ is } A_k^{(r_l)}, \tag{2}$$

where $A_j^{(r_l)}$ is the fuzzy set described by the fuzzy membership function $\mu_j^{(r_l)}$, ($j=1$, ..., $p$, $r_l=1$, ..., $n^{(l)}$, $l=1$, ..., $C$). This membership function describes $j$-th symptom values characteristic for the $l$-th diagnosis. The $m_l(s^{(r_l)})$ value is assigned to this focal element. The $m_l(s^{r_l})$, $r_l=1$, ..., $n^{(l)}$, make the basic probability assignment. In the decision support area the latter can be treated as the uncertainty of diagnostic rules.

### 2.1 Basic Probability Assignment

The BPA can be given by a human expert, i.e. diagnostician, but it can be also calculated with the training data. It must fulfil the following conditions:

$$m_l(f) = 0, \quad \sum_{s^{(r_l)} \in \boldsymbol{S}_l} m_l(s^{(r_l)}) = 1, \tag{3}$$

where $f$ is a false premise and $\boldsymbol{S}_l$ is the fuzzy focal element set for the $l$-th diagnosis. In the [6] the basic probability assignment is calculated as a measure of correspondence between $s^{(r_l)}$ focal element and suitable $l$-th diagnosis cases. Its calculation can be briefly described by the following formula:

$$m_l(s^{(r_l)}) = \frac{\sum\limits_{i=1}^{n_l} \gamma_i^{(r_l)}}{\sum\limits_{r_l=1}^{n^{(l)}} \left( \sum\limits_{i=1}^{n_l} \gamma_i^{(r_l)} \right)}, \tag{4}$$

where $n_l$ is $l$-th diagnosis data set size, $n^{(l)}$ is the $\boldsymbol{S}_l$ size and

$$\gamma_i^{(r_l)} = \begin{cases} 1 & \text{for } \eta^{(r_l)} > \eta_{BPA}, \\ 0 & \text{for } \eta^{(r_l)} \leq \eta_{BPA}, \end{cases} \tag{5}$$

where $\eta^{(r_l)}$ is the matching level of the $s^{(r_l)}$ with the data case. According to (4), the basic probability value is the number of cases conformed with the $l$-th diagnosis cases. This number is normalized with the sum of all $s^{(r_l)}$ conformation numbers. This operation provides truth of (3). The conformation condition is based on the matching level, which can be calculated in the following way:

$$\eta^{(r_l)} = \min_{j \in \mathbb{J}^{(r_l)}} \mu_j^{(r_l)}(x_j), \tag{6}$$

where $\mathbb{J}^{(r_l)}$ is the set of considered symptoms in the $s^{(r_l)}$ and $x_j$ is the $j$-th symptom value of data case $\boldsymbol{x}$. According to (5), the $\eta_{BPA}$ influences the basic probability assignment. Lower values of the threshold allow more uncertainty of the focal elements, which are treated as conformed. On the other hand, higher threshold values cause that only the best adjusted focal elements can obtain the highest BPA value.

### 2.2 Diagnosis Belief Measure

The DST-based diagnosis support is related to the diagnosis belief calculation, which adds the basic probability values of the $s^{(r_l)}$ elements, which are sufficiently conformed with the medical data case. The belief measure is calculated as:

$$Bel_l(\boldsymbol{x}) = \sum_{r_l=1}^{n^{(l)}} \gamma_{Bel}^{(r_l)} m_l(s^{(r_l)}), \tag{7}$$

where

$$\gamma_{Bel}^{(r_l)} = \begin{cases} 1 & \text{for } \eta^{(r_l)} > \eta_T, \\ 0 & \text{for } \eta^{(r_l)} \leq \eta_T. \end{cases} \tag{8}$$

According to (7) and (8), $m_l(s^{(r_l)})$ is added to $Bel_l$, if the $s^{(r_l)}$ matching level with data case is higher than determined threshold $\eta_T$. Belief measure should be calculated for all considered diagnoses ($D_l$). Final diagnosis (denoted as $D$) is chosen according to the following formula:

$$D(\boldsymbol{x}) = D_l \text{ for } l = \underset{1 \leq l \leq C}{\operatorname{argmax}} \, (Bel_l(\boldsymbol{x})). \tag{9}$$

It is possible that the highest $Bel$ value can be obtained for more than one diagnosis. In this situation medical diagnosis should not be stated arbitrarily. Hence the data case is treated as undiagnosed. Similarly to the $\eta_{BPA}$, the $\eta_T$ threshold is the parameter, which tunes the inference and allow to use in the belief calculations more or less suited focal elements.

## 3 The Design of Membership Functions

Membership functions for fuzzy focal elements can be designed through statistical analysis of training data. In [6] different membership function shapes are considered and discussed. In this paper, trapezoidal membership functions are used. Their shape can be described as:

$$\mu(x,[a,b,c,d]) = \begin{cases} 0, & x \le a, \\ \frac{x-a}{b-a}, & a < x \le b, \\ 1, & b < x \le c, \\ \frac{d-x}{d-c}, & c < x \le d, \\ 0, & x > d. \end{cases} \tag{10}$$

Let us denote $\boldsymbol{X}_l$ as the subset of the medical data set $\boldsymbol{X}$, for which every data case is assigned to the $l$-th diagnosis. Hence, $\boldsymbol{X}_l$ can be denoted as:

$$\boldsymbol{X}_l = \{\boldsymbol{x}_{l1}, \cdots, \boldsymbol{x}_{li}, \cdots, \boldsymbol{x}_{ln_l}\}, \tag{11}$$

where $n_l$ is the $\boldsymbol{X}_l$ set size. Moreover, let us denote $\boldsymbol{x}_{jl}$ as $j$-th symptom values from $\boldsymbol{X}_l$ as:

$$\boldsymbol{x}_{jl} = \{x_{j1} \cdots, x_{ji} \cdots, x_{jn_l}\}. \tag{12}$$

In the research one membership function is designed for each $j$-th symptom and $l$-th diagnosis data subset. Hence $C{\cdot}p$ membership functions are designed. To this end, statistical measures of the $\boldsymbol{x}_{jl}$ should be found: $\bar{x}_{jl}$ and $\sigma_{jl}$ as the mean and standard deviation values. This values can be used to treat the $\boldsymbol{x}_{jl}$ as normally distributed data. Remaining values, which are used for the trapezoidal function design are: lower and upper quartiles ($Q_{1_{jl}}$, $Q_{3_{jl}}$) of normal distribution and the intersection point $x_c^{(l)}$ between two neighbouring ($l$-th and $(l+1)$-th) distributions. It is clear that the trapezoidal function shapes should be designed individually for data subsets with the lowest, intermediate and highest mean values. The equations used to trapezoidal membership function parameter calculation are presented in the Table 1. The $z$ and $Z$ in Table 1 are sufficiently small and big numbers according to minimal and maximal values of $\boldsymbol{x}_{jl}$, respectively. This operation provides, that trapezoidal function are left- and right-opened for the lowest and the highest value of training data, respectively. However, for few data couples it is possible that lower or upper quartile outreach suitable $x_c^{(l)}$ value. Hence the design must be protected from incorrectly parameter calculation. It can be done with two conditions:

- if $b \le x_c^{(l-1)}$:
  $a = 0.99 \cdot x_c^{(l-1)}$,
  $b = 1.01 \cdot x_c^{(l-1)}$,
- if $c \ge x_c^{(l)}$:
  $c = 0.99 \cdot x_c^{(l)}$,
  $d = 1.01 \cdot x_c^{(l)}$,

which recalculate function parameters to obtain sufficiently steep slopes [7].

**Table 1.** Trapezoidal membership function parameters calculated according to the data statistical analysis

| $\boldsymbol{x}_{jl}$ with the lowest $\bar{x}_{jl}$ | $\boldsymbol{x}_{jl}$ with intermediate $\bar{x}_{jl}$ | $\boldsymbol{x}_{jl}$ with the highest $\bar{x}_{jl}$ |
|---|---|---|
| $a = z \cdot \min\limits_{1 \leq i \leq n_l} (x_{ji})$ | $a = b - 2(b - x_c^{(l-1)})$ | $a = b - 2(b - x_c^{(l-1)})$ |
| $b = \min\limits_{1 \leq i \leq n_l} (x_{ji})$ | $b = Q_{1_{jl}}$ | $b = Q_{1_{jl}}$ |
| $c = Q_{3_{jl}}$ | $c = Q_{3_{jl}}$ | $c = \max\limits_{1 \leq i \leq n_l} (x_{ji})$ |
| $d = c + 2(x_c^{(l)} - c)$ | $d = c + 2(x_c^{(l)} - c)$ | $d = Z \cdot \max\limits_{1 \leq i \leq n_l} (x_{ji})$ |

## 4 Rule Selection Algorithm

The main aim of the rule extraction is to find the best subset of $2^p - 1$ possible focal elements of the $l$-th diagnosis. The solution should provide satisfactory results in the view of the diagnosis efficiency and rule number. The rule elimination algorithm was used to obtain simple rules in the [6,8]. According to these results a different way to the rule extraction can be proposed when an acceptable effectiveness must go along with the simplest rules. Here, the rule selection algorithm is described. Its performance is based on iterative focal element adding to the created fuzzy focal element set to obtain the best diagnosis efficiency. The rule selection algorithm can be described in the following steps:

1. Choose training data $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_i, ..., \boldsymbol{x}_N\}$ as well as the $\eta_{BPA}$ and $\eta_T$ thresholds.
2. Design fuzzy membership functions for each symptom and each diagnosis $\mu_j^{(l)}(x_j)$, $l = 1, \cdots, C$, $j = 1, \cdots, p$ through statistical analysis of $\boldsymbol{X}$.
3. Create initial fuzzy focal element sets $\hat{\boldsymbol{S}}_l$ and calculate the basic probability assignment for each set of diagnostic rules $s^{(r_l)}$, $r_l = 1, \cdots, n^{(l)}$, $l = 1, \cdots, C$.
4. For each diagnosis, find the first focal element with the maximal $m_l(s^{(r_l)})$ value in $\hat{\boldsymbol{S}}_l$ and move it to the new created focal element set $\boldsymbol{S}_l$.
5. Recalculate the basic probability assignment for every fuzzy focal element $s^{(r_l)}$ in $\boldsymbol{S}_l$, $l = 1, \cdots, C$.
6. Calculate the belief measure $Bel_l(\boldsymbol{x}_i)$ for every data case, $i = 1, ..., N$. Select the final diagnosis, $l = 1, \cdots, C$.
7. Calculate the diagnosis error as the number of cases misdiagnosed and cases, for which the diagnosis can not be selected.
8. If it is the first iteration, go to step **11**. Otherwise go to step **9**.
9. If the diagnosis error increased, go to the step **13**. Otherwise go to step **10**.
10. If yet achieved error is the lowest, go to step **11**. Otherwise go to step **12**.
11. Save actual focal element sets $\boldsymbol{S}_l$ as the best ($\boldsymbol{S}'_l$), $l = 1, \cdots, C$.
12. For each diagnosis where error occurs, find the focal element with the maximal $m_l(s^{(r_l)})$ value in $\hat{\boldsymbol{S}}_l$ and move them to the created focal element set $\boldsymbol{S}_l$ and go to the step **5**.
13. Terminate the algorithm. Final focal element sets are $\boldsymbol{S}'_l$.

In this algorithm the "most valuable" focal elements are gradually moved into the final rule set. This allows for prevalence of focal elements of the best differentiation force among diagnoses. The initial fuzzy focal element set $\hat{\boldsymbol{S}}_{\boldsymbol{l}}$ ($l = 1, ..., C$) can contain all possible combinations of $p$ symptoms, hence its size grows exponentially with the symptom number. On the other hand, the objective of the algorithm is to extract simple rules. Hence, the $\hat{\boldsymbol{S}}_{\boldsymbol{l}}$ size can be limited. It is possible to specify the maximum symptom number in the focal element. Thus, focal element with too many symptom conditions (which have bad influence on their readability) are not created. This detail also affects the reduction of calculations when the high-dimensional data are proceed.

## 5 Experiments

*Thyroid Disease* [9] is commonly known medical data set and often used in the diagnostic rule extraction methods. This source provides 10 different data sets related to the thyroid gland functionality. In this paper the "*new-thyroid*" set is considered. The set contains 215 medical data cases divided into three thyroid diagnoses: normal activity (150 cases), hypothyroidism (30 cases) and hyperthyroidism (35 cases). Each data case is also the set of 5 medical test results. All of them are numerical values related to following hormones:

1. triiodothyronine-resin uptake test (percentage value),
2. total serum thyroxin (T4),
3. total serum thiiodothyronine (T3),
4. total serum thyrotropine (TSH),
5. maximal TSH serum difference after 200 mg of TSH-releasing hormone injection.

Trapezoidal membership functions designed for the *Thyroid Disease* data are presented in Fig. 1. The $\eta_{BPA}$ and $\eta_T$ thresholds influence the rule extraction performance. The couple that provides the highest efficiency is: $\eta_{BPA} = 0.4$, $\eta_T = 0.25$. The performance of the algorithm of rule selection is presented in Fig. 2. Diagnosis error and rule number in every iteration are depicted with points and bars, respectively. The former are connected by line for readability. Since data symptom number is low, size of the initial fuzzy focal element sets does not have be limited. Initial focal element sets ($\hat{\boldsymbol{S}}_l$, $l = 1, ..., C$) include 31 focal elements for each diagnosis. According to the Fig. 2, fuzzy focal element set obtained in the eighth iteration is chosen as the best set due to only two misdiagnosed cases. There were no further iteration, for which the diagnosis error has decreased. The efficiency of the fuzzy focal element set can be calculated in the following way:

$$E = \frac{N - \epsilon}{N} \cdot 100\% \tag{13}$$

where $\epsilon$ is the diagnosis error and $N$ is the data set size. According to (13), the set of 23 fuzzy focal elements created in the eighth iteration provides 99.07% efficiency, which is better results than published in [6]. The rule set for the

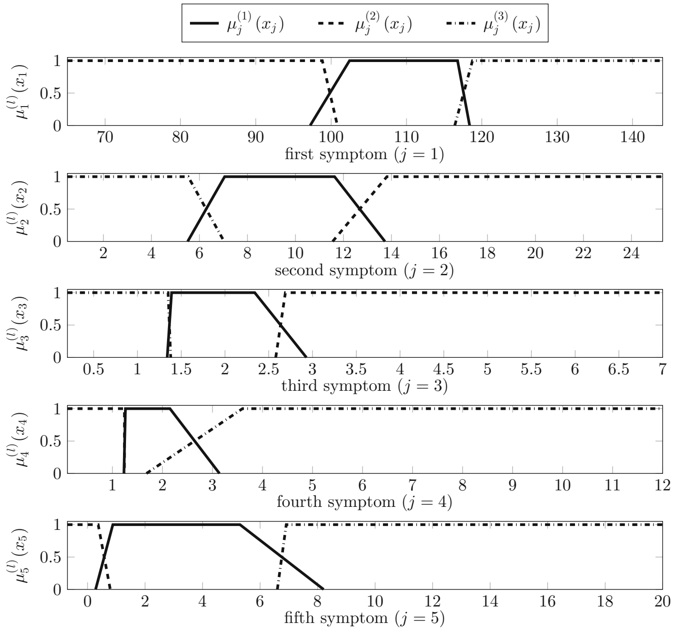


**Fig. 1.** Trapezoidal membership functions designed for the *Thyroid Disease* data set. Leftmost, middle and rightmost functions describe “low”, “normal” and “high” symptom value, respectively.

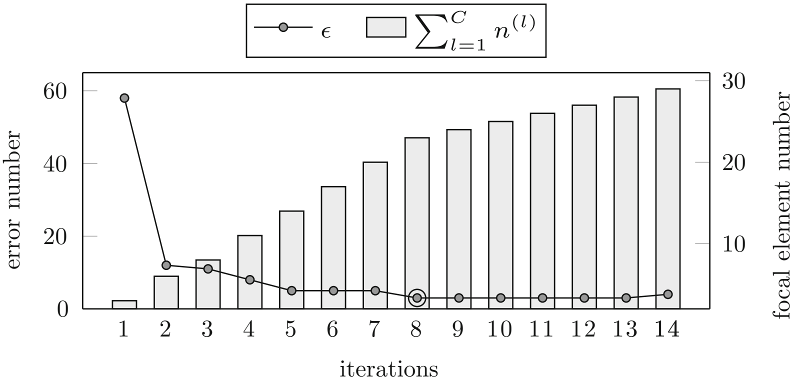


**Fig. 2.** Rule selection algorithm performance with the *Thyroid Disease* data set

thyroid disease diagnosis extracted from the training data are presented in the Table 2. It can be noticed that this efficiency comes at a price of numerous focal elements. This issue is related to the well-known trade-off between high reliability and good interpretability of the extracted knowledge [3]. With the $\eta_{BPA}$ and $\eta_T$ threshold there is possibility to tune the extracted knowledge. Diagnosis

**Table 2.** Fuzzy focal element sets obtained as a result of the rule selection algorithm. The $\eta_{BPA}$ and $\eta_T$ thresholds are set to 0.5 and 0.25, respectively.

| $s^{(r_l)}$ | $m_1(s^{(r_l)})$ | $m_2(s^{(r_l)})$ | $m_3(s^{(r_l)})$ |
|---|---|---|---|
| $X_1$ is $A_1^{(1)}$ | 0.13 | 0.11 | 0.14 |
| $X_2$ is $A_2^{(2)}$ | 0.15 | 0.14 | 0.17 |
| $X_3$ is $A_3^{(3)}$ | 0.13 | 0.12 | 0.14 |
| $X_4$ is $A_4^{(4)}$ | – | 0.11 | 0.15 |
| $X_5$ is $A_5^{(5)}$ | 0.14 | 0.15 | 0.13 |
| $X_1$ is $A_1^{(6)}$ and $X_2$ is $A_2^{(6)}$ | 0.11 | – | 0.13 |
| $X_1$ is $A_1^{(9)}$ and $X_5$ is $A_5^{(9)}$ | 0.10 | – | – |
| $X_2$ is $A_2^{(10)}$ and $X_3$ is $A_3^{(10)}$ | 0.11 | – | – |
| $X_2$ is $A_2^{(11)}$ and $X_4$ is $A_4^{(11)}$ | – | – | 0.14 |
| $X_2$ is $A_2^{(12)}$ and $X_5$ is $A_5^{(12)}$ | 0.13 | 0.14 | – |
| $X_3$ is $A_3^{(14)}$ and $X_5$ is $A_5^{(14)}$ | – | 0.11 | – |
| $X_4$ is $A_4^{(15)}$ and $X_5$ is $A_5^{(15)}$ | – | 0.11 | – |

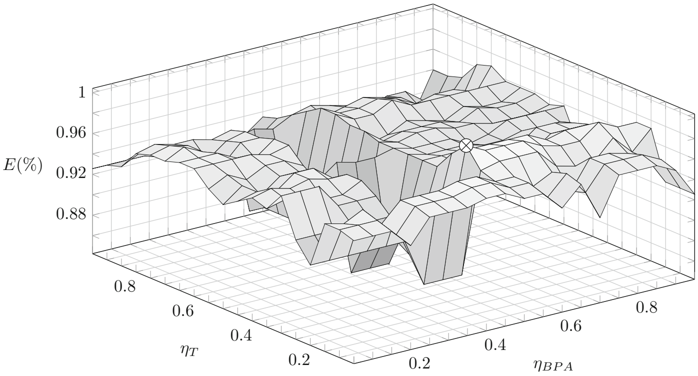


**Fig. 3.** Diagnosis efficiencies obtained for all combination of $\eta_{BPA}$ and $\eta_T$ thresholds. The highest efficiency is indicated by ⊗.

efficiency dependence on the thresholds determination in the range 0.05 to 0.95 with 0.05 step is presented in Fig. 3. The highest obtained efficiency is indicated by mark (⊗). With the $\eta_{BPA}$ and $\eta_T$ thresholds set to 0.05 and 0.1 it is possible to obtain the lowest possible number of focal elements equal six providing slightly worse efficiency equal 95.35%. A lower $\eta_{BPA}$ threshold allows the simplest rules to obtain higher BPA values and the final efficiency is acquired for very few rules. This rule set is presented in Table 3. It is possible to translate the fuzzy focal elements to the heuristic rules, which represent extracted knowledge from the medical data. According to the Fig. 1 fuzzy membership functions $\mu_j^{(l)}(x_j)$,

**Table 3.** Fuzzy focal element sets obtained as a result of the rule selection algorithm. The $\eta_{BPA}$ and $\eta_T$ thresholds are set to 0.2 and 0.3, respectively.

| $s^{(r_l)}$ | $m_1(s^{(r_l)})$ | $m_2(s^{(r_l)})$ | $m_3(s^{(r_l)})$ |
|---|---|---|---|
| $s^{(2)}$: $X_2$ is $A_2^{(2)}$ | 0.5 | 0.5 | 0.5 |
| $s^{(3)}$: $X_3$ is $A_3^{(3)}$ | – | 0.5 | – |
| $s^{(4)}$: $X_4$ is $A_4^{(4)}$ | – | – | 0.5 |
| $s^{(5)}$: $X_5$ is $A_5^{(5)}$ | 0.5 | – | – |

**Table 4.** Heuristic rules for the thyroid gland functionality diagnosis according to the knowledge extracted from the *Thyroid Disease* data (*new-thyroid* set)

| 1. *euthyroid* diagnosis ($n^{(1)} = 2$): | |
|---|---|
| $s^{(2)}$: "T4 level is normal" | $m_1(s^{(2)}) = 0.50$ |
| $s^{(5)}$: "TSH change is normal" | $m_1(s^{(5)}) = 0.50$ |
| 2. *hyperthyroidism* diagnosis ($n^{(2)} = 2$): | |
| $s^{(2)}$: "T4 level is high" | $m_2(s^{(2)}) = 0.50$ |
| $s^{(3)}$: "T3 level is high" | $m_2(s^{(3)}) = 0.50$ |
| 3. *hypothyroidism* diagnosis ($n^{(3)} = 2$): | |
| $s^{(2)}$: "T4 level is low" | $m_3(s^{(2)}) = 0.50$ |
| $s^{(4)}$: "TSH level is low" | $m_3(s^{(4)}) = 0.50$ |

which are used in fuzzy focal elements sets are also describing fuzzy sets related to the natural language statements like "*low*", "*normal*" and "*high test result*". Natural language statements occurring in Table 4 are presented by suitable fuzzy sets in Fig. 1.

## 6 Conclusions

Experimental results show that the Dempster-Shafer theory along with the fuzzy set theory can be used to create a diagnostic rule base and to interpret heuristically data-driven medical knowledge. This framework successfully deals with medical data issues like imprecision and uncertainty. As a result of the rule selection algorithm the fuzzy focal element set can be chosen to provide sufficiently effective diagnosis. Extracted knowledge is comprehensive for a human expert thanks to the intuitive understanding of trapezoidal membership functions of fuzzy focal elements. Performance of the rule extraction algorithm can be tuned with thresholds, which determine the BPA calculation and final diagnosis determination. The search of the best thresholds may involve computational complexity but the rule extraction algorithm is always fast. Generally, the solution with best diagnosis efficiency may not be related to the lowest number of focal elements. Hence, the threshold determination is also vague. The aim is to obtain a good compromise between the rule base reliability and its readability. This paper presents results of the knowledge extraction with the training data. Further research will deal with the mentioned compromise as well as with the rule base generalization quality.

**Acknowledgements.** This research is financed from the statutory activities of the Institute of Electronics of the Silesian University of Technology.

## References

1. Amato, F., López, A., Pena-Meñdez, E.M., Vaňhara, P., Hampl, A., Havel, J.: Artificial neural networks in medical diagnosis. J. Appl. Biomed. **11**(2), 47–58 (2013)
2. Esfandiari, N., Babavalian, M.R., Moghadam, A.-M.E., Tabar, V.K.: Knowledge discovery in medicine: current issue and future trend. Expert Syst. Appl. **41**(9), 4434–4463 (2014)
3. Gacto, M.J., Alcalá, R., Herrera, F.: Interpretability of linguistic fuzzy rule-based systems: an overview of interpretability measures. Inf. Sci. **181**, 4340–4360 (2011)
4. Han, L., Luo, S., Yu, J., Pan, J., Chen, S.: Rule extraction from support vector machines using ensemble learning approach: an application for diagnosis of diabetes. IEEE J. Biomed. Health Inform. **19**(2), 728–734 (2015)
5. Liu, X.D., Feng, X., Pedrycz, W.: Extraction of fuzzy rules from fuzzy decision trees: an axiomatic fuzzy sets (afs) approach. Data Knowl. Eng. **84**, 1–25 (2013)
6. Porebski, S., Straszecka, E.: Membership functions for fuzzy focal elements. Arch. Control Sci. **26**(3), 281–313 (2016)
7. Straszecka, E., Straszecka, J.: Interpretation of Medical Symptoms Using Fuzzy Focal Elements. In: Kurzyński M., Puchała E., WoŹniak M., żołierek A. (eds) Computer Recognition Systems: Proceedings of the 4th International Conference on Computer Recognition Systems CORES 2005. Advances in Soft Computing, vol 30, pp. 287–293. Springer, Heidelberg (2005)
8. Straszecka, E.: Combining uncertainty and imprecision in models of medical diagnosis. Inf. Sci. **176**, 3026–3059 (2006)
9. UCI Machine Learning Repository: Thyroid Disease Data Set. https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease. Accessed 22 Dec 2016

# Gait Recognition Using Motion Trajectory Analysis

Muhammad Hassan Khan[1](✉), Frederic Li[1], Muhammad Shahid Farid[2], and Marcin Grzegorzek[3]

[1] Research Group of Pattern Recognition, University of Siegen, Siegen, Germany
hassan.khan@uni-siegen.de
[2] College of Information Technology, University of the Punjab, Lahore, Pakistan
[3] Faculty of Informatics and Communication, University of Economics in Katowice, Katowice, Poland

**Abstract.** Gait recognition has received significant attention in the recent years due to its applications in numerous fields of computer vision, particularly in automated person identification in visual surveillance and monitoring systems. In this paper, we propose a novel algorithm for gait recognition using spatio-temporal motion characteristics of a person. The proposed algorithm consists of four steps. First, motion features are extracted from video sequence which are used to generate a codebook in the second step. In a third step, the local descriptors are encoded using Fisher vector encoding. Finally, the encoded features are classified using linear Support Vector Machine (SVM). The performance of the proposed algorithm is evaluated and compared with state-of-the-art on two widely used gait databases TUM GAID and CASIA-A. The recognition results demonstrate the effectiveness of the proposed algorithm.

**Keywords:** Gait recognition · Spatiotemporal model · Fisher vector encoding · Visual surveillance

## 1 Introduction

Gait is a walking style of a person and can be used to recognize the individuals. Unlike other identification modalities such as fingerprints, faces or iris biometric which require a cooperation and physical contact of human, gait collection do not require any interaction of human with the system and can be performed at distance or at low resolution in a noninvasive and hidden manners, which is unobtrusive. They are extremely useful in many applications such as surveillance system and service robots interacting with human in daily life. Gait recognition is a challenging task, considering it is affected by various factors, such as the type of clothing or shoes, the walking pace, the nature of the floor, injuries or other similar reasons. Although, gait may not be as powerful as fingerprints or other modalities to identify the person but the characteristic to recognize a person from distance makes it irreplaceable in many cases such as visual

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_8

surveillance [20]. In this paper, we introduce a novel spatio-temporal gait representation to characterize the distinctive motion information of an individual's gait using dense trajectories. It is important to mention that the proposed algorithm does not require silhouette extraction or information related to it like contour and skeleton of a human-body. A sample of dense points is selected in each frame and being tracked in successive frames based on displacement information from a dense optical flow field. Five different local descriptors are computed along dense trajectories. In particular, various combinations of local descriptors are evaluated for gait recognition and the best combination is found. We used Principal Component Analysis (PCA) to project our local descriptors into a low-dimensional subspace to alleviate the curse of dimensionality problem. A codebook approach based on Gaussian Mixture Model (GMM) is exploited to encode the features. We encode our local descriptors using Fisher vectors (FV) and fuse them using representation level fusion [23]. Finally, linear Support Vector Machine (SVM) is employed to classify the encoded features.

## 2 Related Work

Numerous gait recognition methods have been proposed in literature and they can be categorized into two classes: (1) model-based approaches and (2) model-free approaches. In model-based approaches, the structure of human body and its motion are used as a model by tracking the different body parts and joint position over time [22]. Such models which may include stick figure, interlinked pendulum and ellipse are generally constructed based on the prior knowledge of human body shape. Bouchrika et al. [4] proposed a motion-based model using the elliptic Fourier descriptors to extract features from human joints and incorporated them to establish a model for person identification. In [5], the author split the human's body region into three parts and the variance of these parts are combined and used as gait features. Wang et al. [32] constructed structure-based and motion-based models using a condensation framework to refine the feature extraction for gait recognition. Recent studies [4,22,36] have shown that such models can deal, to some extent, with the occlusion and rotation problems, however, the performance of the model-based approaches is highly dependent on the localization of torso, which is not easy to extract. Model-based techniques are computationally expensive and also sensitive to the quality of the video, therefore they are not considered suitable for real-world applications [36]. The model-free approaches operate directly on the sequence of extracted human silhouettes. These techniques either exploit the temporal information of human motion in the recognition process [1,6,20,30] or convert the images of a complete gait sequence into a single template [2,18,28], and used them to recognize the individual's gait. In [20], the human's body silhouettes are extracted using background modeling and averaged over the time in a gait-cycle. This new representation is known as gait energy image (GEI) and then classified using Bayesian classifier. Several improvement in GEI, such as frame difference energy image (FDEI) [6], Gait Entropy Image (GEnI) [1], Chrono-Gait Image (CGI) [30] and Gait Energy Volume (GEV) [27], Depth Gradient Histogram Energy Image (DGHEI) [11] have also been proposed. In [37], various parameters of a person's silhouette and contour (e.g., height and width ratio, silhouette area, width and centroid of the contour) are extracted and gait is approximated by radial basis function (RBF). Microsoft Kinect is also used for gait recognition due to its built-in features of depth data and skeleton information [10,11,27] for human-body segmentation. However, its biggest restriction is the field-of-view, which is very limited (1–4 m) [14]. In comparison with model-based approaches, the model free techniques have shown more convincing recognition results on various gait datasets and they are computationally efficient as well. Several algorithms have used the motion information for gait recognition. Little et al. [18] developed the shape of motion using optical flow field and used it to generate a feature vector for gait recognition. The method in [2] captured the motion intensity and its direction information using optical flow field, and formulated in a histogram based gait representation. In [16], authors calculated the optical flow fields on extracted silhouettes and formulated a new representation for gait recognition, known as Gait Flow Image (GFI). Sivapalan et al. [28] proposed the fusion of HOG and Local Directional Patterns (LDP) to identify the gait and claims significant improvement in recognition. The authors in [36] compute the HOF using the silhouette images of a gait sequence and average them on a full gait cycle. This new gait representation is known as Flow Histogram Energy Image (FHEI). The major disadvantage with silhouette based gait recognition techniques is that they require the precise segmentation of silhouette from the background images which could not always be possible and still a challenging problem in the literature [37]. The rest of the paper is organized as follows: Sect. 3 describes the proposed gait recognition algorithm. Experiments and results are reported in Sect. 4 and the conclusions are drawn in Sect. 5.

## 3 Proposed Method

The proposed gait recognition algorithm works in four steps. In the first step, dense trajectories are generated based on optical flow field and their motion information is encoded using local descriptors. In second step, a codebook is built based on GMM from one million randomly selected motion descriptors. In third step, we encode the local descriptors using Fisher vector encoding and fuse them in a representation level fusion. In the final step, the computed features are classified using linear Support Vector Machine (SVM). Each step of the algorithm is explained in the following sections.

### 3.1 Motion Descriptor

Numerous features have been proposed in recent years and have been successfully exploited in various computer vision problems; SIFT (Scale Invariant Feature Transform) [19], SURF (Speeded-Up Robust Feature) [3], HOG (Histogram of Oriented Gradient) [7], HOF (Histogram of Optical Flow) [17], MBH (Motion

Boundary Histogram) [8], and trajectory [31] are a few to mention. Recently, dense trajectories have shown excellent results in action recognition [23,31]. Our motivation to use dense trajectories for gait recognition is because they encode the local motion patterns and they can be easily extracted from video sequences. A sample of dense points is selected from each frame and being tracked in successive frames based on displacement information obtained from a dense optical flow field. Specifically, each point $P_t = (x_t, y_t)$ at frame $t$ is tracked in frame $t+1$ by median filtering in a dense optical flow field. Given a trajectory of length $L$, a sequence of displacement vector $S$ is formed as follow:

$$S = (\varDelta P_t, ....., \varDelta P_{t+L-1}), \tag{1}$$

where $\varDelta P_t = (P_{t+1} - P_t) = (x_{t+1} - x_t, y_{t+1} - y_t)$. The resulting vector $S$ is then normalized by the sum of the magnitudes of the displacement vector. That is,

$$S' = \frac{(\varDelta P_t, ....., \varDelta P_{t+L-1})}{\sum_{j=t}^{t+L-1} \|\varDelta P_j\|} \tag{2}$$

The descriptor $S'$ encodes the shape of trajectory. Wang et al. [31] proposed the HOG and HOF features along the dense trajectories. Moreover, two derivatives along the horizontal and the vertical components of the optical flow are computed, known as MBH$_x$ and MBH$_y$ respectively, to encode the relative motion information between pixels along the respective axis. The orientation information of each local descriptors are quantized into 8-bin histograms, with an additional zero bin for HOF (i.e., in total 9 bins) and normalized with $L_2$-norm separately. We tested various combinations of all local descriptors on TUM GAID dataset [11] to evaluate their performance for gait recognition. Figure 1 shows the accuracy achieved by various combinations of these features. From the experimental results, we concluded that HOG in combination with MBH outperforms the others. HOG contains the information of static appearance whereas the MBH

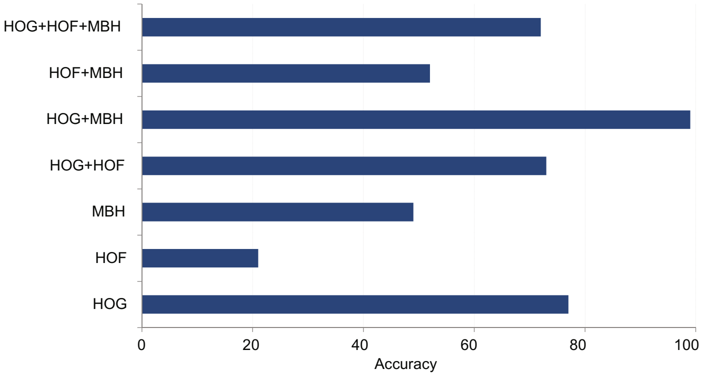


**Fig. 1.** Performance of various motion descriptors for gait recognition on TUM GAID gait database.

highlights the information about the changes in optical flow field (i.e., motion boundaries). Combining information about person appearance and local motion characteristics therefore greatly improves the result of gait recognition.

### 3.2 Codebook Generation

To build a codebook, we randomly select one million features from each descriptors. Gaussian mixture model (GMM) is a generative model to describe the distribution over feature space [23]:

$$p(X \mid \theta) = \textstyle\sum_{i=1}^{K} w_i \mathcal{N}(x \mid \mu_i, \sum_i) \tag{3}$$

where $i$ is the mixture number (i.e., cluster number), $K$ is total number of clusters, $w_i$ is the weight of $i$th cluster and $\mu_i$, $\sum_i$ are the mean and covariance matrix of the $i$th cluster, respectively. Moreover, $\theta = \{w_i, \mu_i, \sum_i, i = 1, 2, ...., K\}$ is the set of model parameters. $\mathcal{N}(x \mid \mu_i, \sum_i)$ is the $D$-dimensional Gaussian distribution and can be expressed as,

$$\mathcal{N}(x \mid \mu_i, \textstyle\sum_i) = \frac{1}{\sqrt{(2\pi)^D \mid \sum_i \mid}} e^{\frac{-1}{2}(x-\mu_i)^\top} \textstyle\sum_i^{-1}(x - \mu_i) \tag{4}$$

In a given feature set $X = \{x_1, ..., x_t\}$, the optimal parameters of GMM are learned through maximum likelihood estimation [21]. Then the soft assignment of data $x_t$ to cluster $i$ (also known as posterior probability) is defined as,

$$q_t(i) = \frac{w_i \mathcal{N}(x_t \mid \mu_i, \sum_i)}{\sum_{j=1}^{K} w_j \mathcal{N}(x_t \mid \mu_j, \sum_j)} \tag{5}$$

We consider that each model represents a specific motion pattern shared by the descriptors in the codebook. Unlike the k-means clustering, which performs hard assignment of feature descriptor to codeword, Expectation maximization (EM) algorithm of GMM performs soft assignments of feature descriptor to each mixture component. In this way, the local descriptors will be assigned to multiple clusters in a weighted manner using the posterior component probability given by the descriptor.

### 3.3 Feature Encoding

Once the local descriptors are extracted, they are used to construct a signature to describe the video. Feature encoding converts the local descriptors into a fixed length vector and this task is usually accomplished by vector quantization of feature vectors and building a histogram of visual words [29]. Inspired by the recent success of Fisher vector (FV) encoding in image classification, object detection and action recognition [24,26], we choose it to encode our descriptors under GMM. FV is an extension of bag-of-visual words (BoV) and derived from Fisher kernel [26] which combines the characteristics of both discriminative and generic approaches. A given feature set $X = \{x_t, t = 1, ...., T\}$ of local descriptors can be modeled into a vector by using the probability density function $p(X|\theta)$ (See Eq. 3). The $X$ can be mapped into a vector by computing the gradient vector of its log-likelihood function at the current model parameters $\theta$ [29],

$$F_X = \frac{1}{T}\nabla_\theta logp(X|\theta), \tag{6}$$

where $F_X$ representing the FV and its dimensionality depends on the number of parameters in $\theta$. The gradient of the log-likelihood function $\nabla_\theta$ describes the contribution of parameters in the generation process. Assuming that $x_t$ is $D$-dimensional local descriptor, its gradient vector with respect to mean $\mu_i$ and covariance $\sum_i$ is defined as [25]:

$$u_i = \frac{1}{T\sqrt{w_i}} \sum_{t=1}^{T} q_t(i) \frac{x_t - \mu_i}{\sum_i}, \tag{7}$$

$$v_i = \frac{1}{T\sqrt{2w_i}} \sum_{t=1}^{T} q_t(i) \left[ \frac{(x_t - \mu_i)^2}{\sum_i^2} - 1 \right], \tag{8}$$

where $q_t(i)$ is the soft assignments of $t$ descriptor to $i$th Gaussian component (5) and $u$, $v$ are $D$-dimensional vectors. Equations 7 and 8 are known as the first and second order differences of descriptor points to cluster centers, respectively. The Fisher encoding for the set of local descriptors $X$ is computed by concatenating the all $u$ and $v$ for all $K$ components. That is,

$$f = [u_1^\top, v_1^\top, u_2^\top, v_2^\top, ....., u_K^\top, v_K^\top]^\top \tag{9}$$

The total size of encoded vector is $2KD$ where $K$ is total number of clusters and $D$ is the dimension of descriptor. We encode the HOG, MBH$_x$ and MBH$_y$ descriptors using the above description and fuse them using representation level fusion [23].

### 3.4 Classification

The encoded vectors are classified using Linear Support Vector Machine (SVM). SVM is considered a powerful tool for solving classification problems in many applications. Due to the high dimensionality of our features, we decided to use SVM as a classifier. In the comparison of SVM, the other similarity based classifiers like K-Nearest Neighbor and probability based classifiers such as Naive Bayes do not perform well on high dimensional features [12,13]. SVM first maps the training samples in high dimensional space and then extracts a hyper-plane between the different classes of objects using the principle of maximizing the margin. Because of this principle, the generalization error of the SVM is theoretically independent from the number of feature dimensions. In this paper, we used LIBLINEAR SVM library [9] for classification.

## 4 Experiments and Results

The performance of the proposed method is evaluated on two widely used gait recognition databases: TUM GAID database [11] and CASIA dataset A [34]. In all experiments, we compute the local descriptors on each video sequence using dense trajectories as described in the previous section. Each of our local descriptor is 96 dimensional long. We used PCA to project them into lower dimension space to alleviate the curse of dimensionality problem, prior to compute codebook and FV. We observed from experiments that the reduced dimension of 48 produces the best results. The number of clusters in GMM modeling is fixed to 256 in all experiments.

### 4.1 Performance Evaluation on TUM GAID Database

The TUM GAID [11] is one of the biggest gait databases. It consists of 3370 video sequences of 305 subjects, captured in outdoor environment. The database was recorded at 30 frames/second at Technical University of Munich, Germany in two different sessions using Microsoft Kinect camera. The first session was recorded in January 2012, which is the winter season in the region and temperature is around −15 °C. Thus, the subjects were wearing heavy jackets and mostly winter boots. A total of 176 subjects were recorded in this session. The second session was recorded in April 2012 when temperature is around +15 °C, thus, subjects were wearing significantly different clothes. Therefore, there was a substantial variation in the clothing of participants. A total of 161 subjects were participated in the second session. There was a subset of 32 subjects who participated in both sessions. Ten walk sequences were captured for each subject namely: normal walk ($N$), walk with bag-pack ($B$) and walk with coating shoes ($S$). Each subject has six sequences of $N$, two sequences of $B$ and two sequences of $S$, recorded while the subject is moving from left-to-right and right-to-left in the lateral view. Each subject in the subset of 32 peoples (who participated in both sessions) has 10 more walk sequences which are represented as: normal walk after time ($TN$), walk with bag-pack after time ($TB$), and walk with coating shoes after time ($TS$). This subset allows the research community to analyze the effectiveness of their methods under time and clothing variations. The author in [11] has divided the dataset of 305 subjects in two sets: (1) development set consist of 150 subjects; (2) test set contains the recording of 155 subjects. We used the same division to evaluate the performance of our algorithm. We compare the performance of the proposed gait recognition algorithm with several competing approaches, including DGHEI [11], SVIM [35], GEI [11], and GEV [11]. The results are reported in Table 1, the best results in each experiments are indicated in bold font. The results show that in most experiments the proposed algorithm outperforms the other approaches by significant margins. In experiment C, SVIM [35] shows better performance and in experiment A, DGHEI [11] performs marginally better then ours.

### 4.2 Performance Evaluation on CASIA-A Database

The CASIA-A dataset contains walking sequences of 20 subjects. Each subject in the dataset has four sequences in three different directions; parallel to the image plane (i.e., lateral view), 45° (i.e., oblique view) and 90° (i.e., frontal view). Each subject walks twice, from left to right and right to left. The proposed method is evaluated on the sequences recorded in lateral view. We used the first 3 out of 4 sequences in gallery set and the fourth one is used in probe set. Further details of the experiment are described in Table 2. We also compared the performance of the proposed algorithm with the state-of-the-art methods: Wang [34], Ning [33], Bashir [2], and Kusakunniran [15]. The results are outlined in Table 3, which shows that both proposed algorithm and Kusakunniran [15] approach achieve 100% recognition. The results of the experiments obtained on these two large gait databases show the effectiveness of the proposed algorithm. On TUM GAID database, the difference of recognition rate of the proposed algorithm and the state-of-the-art is up to 35%. On CASIA-A database, the proposed algorithm achieves 100% recognition.

**Table 1.** Comparison of recognition performance on TUM GAID gait database. GS is Gallery Set and PS is Probe Set. The best results in each experiment are indicated in bold font.

| Experiment | GS | PS | DGHEI [11] | SVIM [35] | GEI [11] | GEV [11] | Proposed |
|---|---|---|---|---|---|---|---|
| A | N | N | **99.4** | 99.0 | 96.8 | 94.2 | 97.74 |
| B | N | B | 27.1 | 18.4 | 3.9 | 13.9 | **39.35** |
| C | N | S | 56.2 | **96.1** | 88.7 | 87.7 | 85.16 |
| D | N | TN | 44 | 15.6 | 28 | 28 | **46.88** |
| E | N | TB | 6 | 3.1 | 0 | 0 | **18.75** |
| F | N | TS | 9 | 28.1 | 22 | 22 | **34.38** |

**Table 2.** Details of experiment on CASIA-A gait database.

| Experiment | A |
|---|---|
| Gallery set | First 3 sequences of each subject |
| Probe set | Forth sequence of each subject |
| Gallery size | 60 |
| Probe size | 20 |

**Table 3.** Comparison of recognition performance on CASIA-A gait database. The best results in each experiment are indicated in bold font.

| Experiment | Wang [34] | Ning [33] | Bashir [2] | Kusakunniran [15] | Proposed |
|---|---|---|---|---|---|
| A | 82.5 | 88.75 | 97.5 | **100** | **100** |

## 5 Conclusion

A new gait recognition algorithm based on spatio-temporal characteristics of a human motion is presented in this paper. The proposed method constructs a set of dense trajectories by tracking a set of points in successive frames. Five different local descriptors are computed along the dense trajectories and various combinations of all local descriptors are evaluated to test their performance for gait recognition. The descriptors are encoded using Fisher vector encoding and classified using Linear SVM. The experimental results on two large public datasets reveal that proposed algorithm is highly accurate.

## References

1. Bashir, K., Xiang, T., Gong, S.: Gait recognition without subject cooperation. Pattern Recognit. Lett. **31**(13), 2052–2060 (2010)
2. Bashir, K., Xiang, T., Gong, S., Mary, Q.: Gait representation using flow fields. In: BMVC, pp. 1–11 (2009)
3. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: speeded up robust features. In: ECCV 2006, pp. 404–417 (2006)
4. Bouchrika, I., Nixon, M.S.: Model-based feature extraction for gait analysis and recognition. In: Gagalowicz, A., Philips, W. (eds.) MIRAGE 2007. LNCS, vol. 4418, pp. 150–160. Springer, Heidelberg (2007). doi:10.1007/978-3-540-71457-6_14
5. Chai, Y., Wang, Q., Jia, J., Zhao, R.: A novel human gait recognition method by segmenting and extracting the region variance feature. IEEE ICPR **4**, 425–428 (2006)
6. Chen, C., Liang, J., Zhao, H., Hu, H., Tian, J.: Frame difference energy image for gait recognition with incomplete silhouettes. Pattern Recognit. Lett. **30**(11), 977–984 (2009)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE CVPR, vol. 1, pp. 886–893, June 2005
8. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: ECCV, pp. 428–441 (2006)
9. Fan, R.E., et al.: LIBLINEAR: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)
10. Gianaria, E., Balossino, N., Grangetto, M., Lucenteforte, M.: Gait characterization using dynamic skeleton acquisition. In: Proceedings of International Workshop Multimedia Signal Process, (MMSP), pp. 440–445, September 2013
11. Hofmann, M., Bachmann, S., Rigoll, G.: 2.5D gait biometrics using the depth gradient histogram energy image. In: IEEE BTAS, pp. 399–403 (2012)
12. Khan, M.H., Helsper, J., Yang, C., Grzegorzek, M.: An automatic vision-based monitoring system for accurate Vojta-therapy. In: IEEE/ACIS ICIS, pp. 1–6 (2016)
13. Khan, M.H., Helsper, J., Boukhers, Z., Grzegorzek, M.: Automatic recognition of movement patterns in the Vojta-therapy using RGB-D data. In: IEEE ICIP, pp. 1235–1239 (2016)
14. Khan, M.H., Shirahama, K., Farid, M.S., Grzegorzek, M.: Multiple human detection in depth images. In: IEEE International Workshop on MMSP, pp. 1–6 (2016)
15. Kusakunniran, W.: Attribute-based learning for gait recognition using spatio-temporal interest points. Image Vis. Comput. **32**(12), 1117–1126 (2014)

16. Lam, T.H., Cheung, K.H., Liu, J.N.: Gait flow image: a silhouette-based gait representation for human identification. Pattern Recognit. **44**(4), 973–987 (2011)
17. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: IEEE CVPR, pp. 1–8, June 2008
18. Little, J., Boyd, J.: Recognizing people by their gait: the shape of motion. Videre J. Comput. Vis. Res. **1**(2), 1–32 (1998)
19. Lowe, D.G.: Object recognition from local scale-invariant features. In: EEE ICCV, vol. 2, pp. 1150–1157 (1999)
20. Man, J., Bhanu, B.: Individual recognition using gait energy image. IEEE Trans. Pattern Anal. Mach. Intell. **28**(2), 316–322 (2006)
21. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley, New York (2004)
22. Nixon, M.S., Tan, T., Chellappa, R.: Human Identification Based on Gait. Springer, Heidelberg (2010)
23. Peng, X., Wang, L., Wang, X., Qiao, Y.: Bag of visual words and fusion methods for action recognition: comprehensive study and good practice. Comput. Vis. Image Underst. **150**, 109–125 (2016). http://www.sciencedirect.com/science/article/pii/S1077314216300091
24. Peng, X., Zou, C., Qiao, Y., Peng, Q.: Action recognition with stacked fisher vectors. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, pp. 581–595. Springer, Heidelberg (2014)
25. Perronnin, F., Sánchez, J., Mensink, T.: Improving the Fisher Kernel for large-scale image classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15561-1_11
26. Sanchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: theory and practice. Int. J. Comput. Vis. **105**(3), 222–245 (2013)
27. Sivapalan, S., Chen, D., Denman, S., Sridharan, S., Fookes, C.: Gait energy volumes and frontal gait recognition using depth images. In: IEEE IJCB, pp. 1–6 (2011)
28. Sivapalan, S., Chen, D., Denman, S., Sridharan, S., Fookes, C.: Histogram of weighted local directions for gait recognition. In: IEEE CVPR Workshop, pp. 125–130 (2013)
29. Sun, C., Nevatia, R.: Large-scale web video event classification by use of fisher vectors. In: IEEE WACV, pp. 15–22 (2013)
30. Wang, C., Zhang, J., Wang, L., Pu, J., Yuan, X.: Human identification using temporal information preserving gait template. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2164–2176 (2012)
31. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: IEEE ICCV, pp. 3551–3558 (2013)
32. Wang, L., Ning, H., Tan, T., Hu, W.: Fusion of static and dynamic body biometrics for gait recognition. IEEE Trans. Circuits Syst. Video Technol. **14**(2), 149–158 (2004)
33. Wang, L., Tan, T., Hu, W., Ning, H.: Automatic gait recognition based on statistical shape analysis. IEEE Trans. Image Process. **12**(9), 1120–1131 (2003)
34. Wang, L., Tan, T., Ning, H., Hu, W.: Silhouette analysis-based gait recognition for human identification. IEEE Trans. Pattern Anal. Mach. Intell. **25**(12), 1505–1518 (2003)
35. Whytock, T., Belyaev, A., Robertson, N.M.: Dynamic distance-based shape features for gait recognition. J. Math. Imaging Vis. **50**(3), 314–326 (2014)
36. Yang, Y., Tu, D., Li, G.: Gait recognition using flow histogram energy image. In: IEEE ICPR, pp. 444–449. IEEE (2014)
37. Zeng, W., Wang, C., Yang, F.: Silhouette-based gait recognition via deterministic learning. Pattern Recognit. **47**(11), 3568–3584 (2014)

# Methodology of the Construction of a GDPLL($k$) Grammar-Based Syntactic Pattern Recognition System

Mariusz Flasiński(✉) and Janusz Jurek

Information Technology Systems Department, Jagiellonian University,
ul. prof. St. Łojasiewicza 4, 30-348 Cracow, Poland
mariusz.flasinski@uj.edu.pl

**Abstract.** GDPLL($k$) grammars have been introduced as a tool for the construction of syntactic pattern recognition-based systems. The grammars have been successfully used in several different applications. The practical experience with the implementation of a syntactic pattern recognition system based on GDPLL($k$) grammars has served to define methodological guidelines for constructing such systems. In the paper key methodological issues are presented.

## 1 Introduction

GDPLL($k$) grammars have been defined [3,13] as a tool for constructing syntactic pattern recognition systems. Syntactic pattern recognition [1,2,5,6,10,11,16,18] is one of the main approaches in the field of computer recognition, based on the theory of formal languages, grammars and automata. According to the approach the recognition is made in two phases. In the first phase a structural representation of a pattern is generated. The structure could be of the form of string, tree or graph, consisting of some elementary components, called *primitives*. A set of such structures is treated as a formal language. In the second phase, the structural representation is analyzed by a formal automaton, which is constructed on the basis of a formal grammar generating the corresponding formal language. In result a derivation of the analyzed structure is obtained. The derivation is used by a syntactic pattern recognition system for describing structural features of the pattern and for recognizing (classifying) it. The GDPLL($k$) grammars are string grammars which have several advantages in comparison to other types of string grammars that can be used in syntactic pattern recognition systems [5,6]. They are characterized by very good discriminative properties (they are able to generate a considerable subclass of context-sensitive languages). On the other hand efficient parsing algorithm for GDPLL($k$) grammars has been constructed [13] as well as a grammatical inference algorithm [12,15]. Finally, let us point out that a GDPLL($k$) grammar could be designed as stochastic one [4] (a feature also very important in some applications). Although the grammars have been proven to be a suitable tool for the use in a syntactic pattern recognition system,

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_9

there is a lot of practical issues concerning the construction of such a system for a real-world application. During several years we have conducted research into practical usage of GDPLL($k$) grammars, e.g. we have built a diagnostic system for evaluating of organ of hearing in neonates in electric response audiometry [9], and we have applied GDPLL($k$) grammars in a system for electrical load forecasting [7]. The practical experience with the implementation of a syntactic pattern recognition system based on GDPLL($k$) grammars have allowed us to define methodological guidelines for constructing such systems. In the paper we present and discuss key methodological issues. Section 2 contains basic definitions concerning GDPLL($k$) grammars and a syntactic pattern recognition system based on such grammars. In Sect. 3 the structural representation generator, being the first element of the system, is described. Section 4 concerns the definition of a proper GDPLL($k$) grammar for a given application. In Sect. 5 we discuss the problem of whole system construction and tuning, whereas conclusions are contained in the final section.

## 2 Syntactic Pattern Recognition System Based on GDPLL($k$) Grammars

Let us introduce two basic definitions corresponding to GDPLL($k$) grammars [13].

**Definition 1.** A *generalized dynamically programmed context-free grammar* is a six-tuple $G = (V, \Sigma, O, P, S, M)$, where: $V$ is a finite, nonempty alphabet; $\Sigma \subset V$ is a finite, nonempty set of terminal symbols (let $N = V \setminus \Sigma$); $O$ is a set of basic operations on the values stored in the memory; $S \in N$ is the starting symbol; $M$ is a memory; $P$ is a finite set of productions of the form: $p_i = (\mu_i, L_i, R_i, A_i)$ in which $\mu_i : M \longrightarrow \{TRUE, FALSE\}$ is the predicate of applicability of the production $p_i$ defined with the use of operations ($\in O$) performed over $M$; $L_i \in N$ and $R_i \in V^*$ are left- and right-hand sides of $p_i$ respectively; $A_i$ is the sequence of operations ($\in O$) over $M$, which should be performed if the production is to be applied. □

**Definition 2.** Let $G = (V, \Sigma, O, P, S, M)$ be a generalized dynamically programmed context-free grammar. The grammar $G$ is called a *GDPLL*($k$) *grammar*, if the following two conditions are fulfilled.

1. Stearns's condition of LL($k$) grammars. (The top-down left-hand side derivation is deterministic if it is allowed to look at $k$ input symbols to the right of the current position of the input head in the string).
2. There exists a certain number $\xi$ such that after the application of $\xi$ productions in a left-hand side derivation we get at the "left-hand side" of a sentence at least one new terminal symbol. □

A derivation in GDPLL($k$) grammars is defined in the following way. Before application of a production $p_i$ we test whether $L_i$ occurs in a sentential

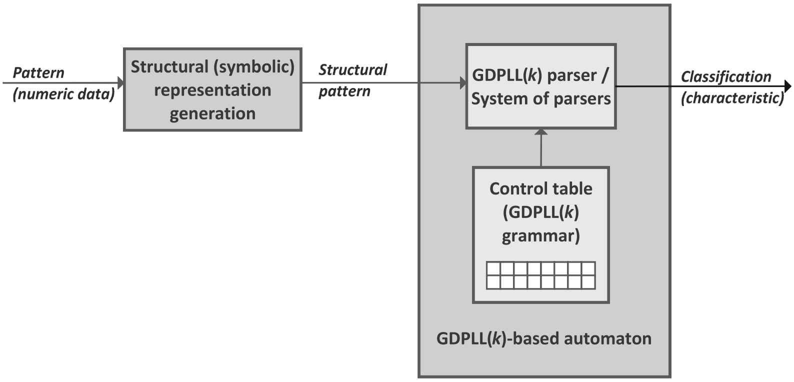

Pattern
(numeric data)
Structural (symbolic) representation generation
Structural pattern
GDPLL(k) parser / System of parsers
Classification (characteristic)
Control table (GDPLL(k) grammar)
GDPLL(k)-based automaton

**Fig. 1.** Syntactic pattern recognition system based on GDPLL($k$) grammars

form derived. Then we check the predicate of applicability of the production. The predicate is defined as an expression based on variables stored in the memory. If the predicate is true, we replace $L_i$ with $R_i$ and then we perform the sequence of operations over the memory. The execution of the operations changes the contents of the memory. It is done with the help of arithmetical and assignment instructions. The automaton for GDPLL($k$) grammars (GDPLL($k$) parser) has been described in [13]. We do not present it in the paper. We just notice here that the algorithm reflects the way how the derivation in the grammars is performed (it proceeds in a top-down manner). The scheme of a syntactic pattern recognition system based on GDPLL($k$) grammars is presented in Fig. 1. The first element of the system is a module responsible for generation of a structural representation of a pattern. It converts numeric data describing a pattern into symbolic information (string of primitives of a string of *fuzzy* primitives) that can be analyzed by a GDPLL($k$)-based automaton. The second element is the automaton. Its construction depends on the application. It may be implemented in the form of a basic GDPLL($k$) parser [13] or it can be a system of parsers dealing with *fuzzy* symbolic information, so called a FGDPLL($k$) transducer [8] The architecture of these two elements of a syntactic pattern recognition system based on GDPLL($k$) grammars is discussed in succeeding sections.

## 3 Construction of Structural Representation Generator

One of the main practical problems with the applications of the syntactic pattern recognition methods concerns elements of noise/distortions or uncertainty in the analyzed patterns [18]. If a pattern is distorted or fuzzy, then the structural representation generator could output wrong or inaccurate primitives. In such case the GDPLL($k$)-based automaton usually does not bring proper results of recognition either. Let us stress, that even one misrecognized primitive could lead to the failure of the analysis of the whole pattern. For example, this problem had to be solved in case of a system for electrical load forecasting [7].

The input data for the system describing a pattern to be recognized contained some forecasted values (air temperature and insolation forecasts). Such values are of course uncertain so they had to be treated as fuzzy. In the system we used probabilistic neural networks (PNN) introduced by Specht [17] to generate *fuzzy primitives*, i.e. a string of *vectors* (instead of a string of alphabet symbols, as usual in case of syntactic pattern recognition applications). The vectors are of the form: $((a_1, p_1), \ldots, (a_n, p_n))$, where $a_i$ is the possible symbolic value of the primitive, and $p_i$ is the probability that the primitive is equal $a_i$. PNN were proved to be a suitable tool, because apart from classification, the networks could have delivered additional information on *how far* the presented pattern was related to all classes (not only to the winner one). It is worth to point out that PNN can be trained on some samples. It makes the construction of the structural representation generator easier. Summing up our considerations, let us formulate the first methodological recommendation concerning the construction of the structural representation generator.

**I.** *If a syntactic pattern recognition system is to be used for distorted or fuzzy patterns, the structural representation generator should be able to deliver additional (fuzzy) information about recognized primitives. A suitable tool that may be used for this purpose is a probabilistic neural network.*

If a GDPLL($k$)-based automaton is to receive a string of fuzzy primitives as an input data, its construction has to be more complex than a basic parser [8].

## 4 Definition of a GDPLL($k$) Grammar

According to the syntactic pattern recognition approach, a set of strings of primitives (or fuzzy primitives) is treated as a formal language. In most practical applications, a syntax of the language is not known in an explicit way, and only a sample of patterns is given. Since usually a number of sample patterns is big, defining a grammar "by hand" is impossible. Fortunately, in case of GDPLL($k$) grammars, a grammatical inference algorithm (an algorithm that generates a grammar automatically on the basis of the sample) has been successfully developed [12,15]. The algorithm can be used to define a proper GDPLL($k$) grammar for a particular application. Let us present main definitions needed to discuss the grammatical inference algorithm [15].

**Definition 3.** Let $A$ be a set of all (terminal) symbols which appear in the sample, $Z$ set of integer variables. *Polynomial specification of a language* is of the form: $L_p(A, Z) = S_i^{p_j(n_k)}$ where: $p_j$ is a polynomial of a variable $n_k \in Z$; variable $n_k$ can be assigned only values greater or equal 1; $S_i$, called *polynomial structure*, is defined in a recursive way:

(a) $S_i = (a_{i_1} ... a_{i_r})$, where $a_{i_j} \in A$. ($S_i$ is a *basic* polynomial structure), or:

(b) $S_i = (S_{i_1}^{p_{i_1}(n_{i_1})} ... S_{i_r}^{p_{i_r}(n_{i_r})})$, where $S_{i_k}$ is defined as in (a) or (b). ($S_i$ is a *complex* polynomial structure). □

**Definition 4.** *Extended* polynomial specification of a language $L_{ep}(A, Z)$ is defined in the following way:

(a) $L_{ep}(A, Z) = L_p(A, Z)$ or:
(b) $L_{ep}(A, Z) = L^1_{ep}(A_1, Z_1)\ L^2_{ep}(A_2, Z_2)\ \ldots\ L^z_{ep}(A_z, Z_z)$, or:
(c) $L_{ep}(A, Z) = L^1_{ep}(A_1, Z_1)\ +\ L^2_{ep}(A_2, Z_2)\ +\ \ldots\ +\ L^z_{ep}(A_z, Z_z)$

(Intuitively, an extended polynomial specification of a language is a polynomial specification with the additional possibility of the application of *alternative* polynomial structures). □

The grammatical inference algorithm for GDPLL($k$) grammars is divided into two phases:

- The first phase is responsible for extraction of the structural features of the sample and generalization of the sample. An extended polynomial specification of the language is obtained as the result of the phase.
- In the second phase, a GDPLL($k$) grammar is generated on the basis of extended polynomial specification of the language.

Both phases are independent of each other. The second phase is well-defined, i.e. the result of this phase is strictly determined by the input data (an extended polynomial specification of language). On the contrary, the first phase is not well-defined since there may be many approaches to the generalization of the sample and the "quality" of an approach depends on a particular application and a particular sample of the language. For example, the generation of a proper grammar was a crucial problem when we built a syntactic pattern recognition system for evaluating the organ of hearing in neonates in Electric Response Audiometry (ERA) [9]. A multitude of sample patterns had been automatically analyzed by the algorithm, which resulted in a definition of an extended polynomial specification of the language. However, the specification had to be adjusted manually. There were some sophisticated medical symptoms in ERA (known to experts) that were not identified automatically as features of the language. Therefore, on the basis of our experience we could recommend a human supervision over the process of extraction of the features of the sample and generalization of the sample. At the end of this section let us we formulate the second methodological remark concerning construction of a syntactic pattern recognition system based on GDPLL($k$) grammars.

**II.** *A definition of a proper GDPLL($k$) grammar could be obtained by the application of the grammatical inference algorithm. The specification of the language being the result of the first phase of the algorithm should be verified (and if its necessary adjusted) by the human designer of the system in order to include all accessible expert knowledge in it.*

Having a GDPLL($k$) grammar defined, we can construct the second module of the syntactic pattern recognition system: a GDPLL($k$)-based automaton.

## 5 System Construction and Tuning

The first element of a GDPLL($k$)-based automaton that should be built is a parser's control table. The control table is a constant structure (for a given grammar) determining an action which the parser should perform at its state characterized by a lookahead string and the top element of the stack. More formally, we may say that the control table contains a specification of *transition functions* of the parser. The control table can be generated automatically by one of the algorithms defined in our previous works [14]. As it has been mentioned in Sect. 3, the construction of a GDPLL($k$)-based automaton could be more complex if the automaton is to receive a string of *fuzzy* primitives as an input data. In order to analyze such input, it is necessary to trace many different paths of derivation (led to alternative symbols in a fuzzy primitive specification). What is more, the number of paths to be considered could increase exponentially every input primitive read. It is also worth pointing out that if the input string is fuzzy the result of the analysis (a classification) should also be fuzzy. It means that we could obtain alternative classifications (having different probabilities). Let us formulate the third methodological recommendation.

**III.** *The syntactic analysis of a pattern described as a string of fuzzy primitives should be performed with the use of an automaton that is able to efficiently analyze all possible paths of derivation and generate a classification of the pattern in the form of several most probable recognitions. FGDPLL(k) transducer can be used for this purpose.*

An FGDPLL($k$) transducer [8] includes a pool of GDPLL($k$) parsers which are needed to simultaneously perform several derivation processes for each possible value of an input primitive. The control of the automaton is responsible for verifying, which of the derivation processes should be continued. For this purpose it uses an auxiliary memory which contains computed probabilities for each derivation process. The output of the automaton is of the fuzzy form and it is called a variant classification of the pattern. Once the GDPLL($k$)-based automaton is constructed, we can start testing and tuning up the whole pattern recognition system. A general scheme of system tuning is shown in Fig. 2 below. Tasks that could be done automatically (with the use of defined algorithms) are marked with dark-grey color. Tasks that need human expertise to be completed are marked with light-grey. Let us consider nine steps presented in Fig. 2.

1. *Primitive definitions' adjustment.* Proper definition of primitives is crucial for the success of the recognition. The definition is sometimes quite straightforward, but more often it needs thorough experimental verification. In order to make the process of primitive specifications adjustment easier we may use *parameters* for primitives. For example in case of recognizing time-series patterns we can parametrize the interval being the base for primitive definition, or in case of recognizing shape structures we can parametrize size of a primitive.

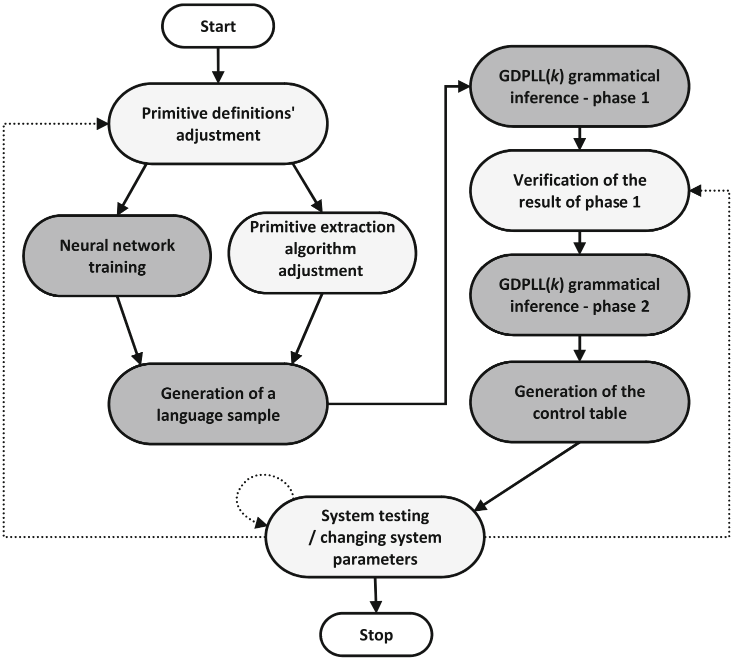

Start
Primitive definitions' adjustment
Neural network training
Primitive extraction algorithm adjustment
Generation of a language sample
GDPLL(k) grammatical inference - phase 1
Verification of the result of phase 1
GDPLL(k) grammatical inference - phase 2
Generation of the control table
System testing / changing system parameters
Stop

**Fig. 2.** Scheme of the system tuning

2. *Primitive extraction algorithm adjustment.* If we change the definition of primitives, the algorithm of their recognition usually should be changed as well. It may require redesign of the algorithm, unless it is parametrized (correspondingly to primitives).
3. *Neural network training.* In case fuzzy primitives are to be used in a given application, a recommended tool for their identification is a probabilistic neural network, PNN (see: Sect. 3). The advantage of using a PNN is the possibility of its *automatical training.*
4. *Generation of a language sample.* A language sample is necessary for grammatical inference algorithm. It can be obtained automatically by the use of primitive extraction algorithm (or a trained PNN) on the samples of input patterns.
5. *GDPLL($k$) grammatical inference - phase 1.* Phase 1 of GDPLL($k$) grammatical inference is responsible for recognition of structural features of the sample and generalization of the sample. An extended polynomial specification of the language is obtained as the result of the phase (see: Sect. 4).
6. *Verification of the result of phase 1.* As we have mentioned in methodological recommendation no. IV, an extended polynomial specification should be reviewed and modified by the human designer of the system in order to include all accessible expert knowledge in it.

7. *GDPLL($k$) grammatical inference - phase 2.* In the second phase of grammatical inference, a GDPLL($k$) grammar is automatically generated on the basis of extended polynomial specification of the language.
8. *Generation of the control table.* Generation of a new control table for GDPLL($k$) parser is the last step of the system tuning/rebuilding (accordingly to the changes in primitives' definitions and in grammar definition).
9. *System testing/changing system parameters.* When the system is built (or rebuilt), the tests can be started. There are some parameters of system functioning that can be adjusted like the maximum number of simultaneous derivation processes in case of FGDPLL($k$) transducer. If the test results are not satisfactory (even after experiments with different parameters), we should return to step 1 (primitive definitions' adjustment) or step 6 (verification of the result of phase 1 of grammatical inferencing).

To sum up the considerations above, let us formulate the last methodological remark.

**IV.** *Tuning a GDPLL($k$)-based syntactic pattern recognition system could be a difficult and complex process. Although there are some supporting algorithms already developed (grammatical inference, neural network training), the process still needs human supervision. The main tasks the designer of the system should take care of are: primitive definitions' adjustment and verification of the result of the first phase of grammatical inference. The whole process should proceed according to the scheme included in Fig.* 2.

## 6 Concluding Remarks

In the paper we have presented some key methodological issues concerning construction of syntactic pattern recognition systems based on GDPLL($k$) grammars. We have successfully implemented such systems in different application areas and developed some useful tools supporting the implementation process. However, there is still need of inventing new methods that can make the generation and tuning of the systems easier. Our research is now aimed at the following goals.

- For now the grammatical inference algorithm for GDPLL($k$) grammars is based on *positive* sample only (i.e. only the words belonging to the language are taken into account). In some applications it could be useful to consider a *negative* sample as well (exemplary words that don't belong to the language). Therefore, the grammatical inference algorithm should be enhanced to take advantage of such information.
- The form of the primitives depends on a particular application. Nevertheless, we are going to develop an universal model of parametrizing of the primitives, that can be used to automate the tuning of the whole pattern recognition system.

The results of the research into problems mentioned above will be a subject of further publications.

## References

1. Bunke, H.O., Sanfeliu, A. (eds.): Syntactic and Structural Pattern Recognition – Theory and Applications. World Scientific, Singapore (1990)
2. Flasiński, M.: Chapter 1: Syntactic pattern recognition: paradigm issues and open problems. In: Chen, C.H. (ed.) Handbook of Pattern Recognition and Computer Vision, 5th edn., pp. 3–25. World Scientific, London (2016)
3. Flasiński, M., Jurek, J.: Dynamically programmed automata for quasi context sensitive languages as a tool for inference support in pattern recognition-based real-time control expert systems. Pattern Recogn. **32**, 671–690 (1999)
4. Flasiński, M., Jurek, J.: On the analysis of fuzzy string patterns with the help of extended and stochastic GDPLL(k) grammars. Fundamenta Informaticae **71**, 1–14 (2006). IOS Press, Amsterdam, Berlin, Oxford
5. Flasiński, M., Jurek, J.: Syntactic Pattern Recognition: Survey of Frontiers and Crucial Methodological Issues, Advances in Intelligent and Soft Computing 95, 187–196. Springer, Heidelberg (2011)
6. Flasiński, M., Jurek, J.: Fundamental methodological issues of syntactic pattern recognition. Pattern Anal. Appl. **17**, 465–480 (2014). Springer, Heidelberg
7. Flasiński, M., Jurek, J., Peszek, T.: Application of syntactic pattern recognition methods for electrical load forecasting. Adv. Intell. Syst. Comput. **403**, 599–608 (2016). Springer, Switzerland
8. Flasiński, M., Jurek, J., Peszek, T.: Analysis of fuzzy string patterns with the help of syntactic pattern recognition. Adv. Intell. Syst. Comput. **400**, 115–126 (2016). Springer, Switzerland
9. Flasiński, M., Reroń, E., Jurek, J., Wójtowicz, P., Atłasiewicz, K.: Mathematical linguistics model for medical diagnostics of organ of hearing in neonates. In: Wyrzykowski, R., Dongarra, J., Paprzycki, M., Waśniewski, J. (eds.) PPAM 2003. LNCS, vol. 3019, pp. 746–753. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24669-5_98
10. Fu, K.S.: Syntactic Pattern Recognition and Applications. Prentice Hall, Englewood Cliffs (1982)
11. Gonzales, R.C., Thomason, M.G.: Syntactic Pattern Recognition: An Introduction. Addison-Wesley, Reading (1978)
12. Jurek, J.: Towards grammatical inferencing of GDPLL($k$) grammars for applications in syntactic pattern recognition-based expert systems. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 604–609. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24844-6_91
13. Jurek, J.: Recent developments of the syntactic pattern recognition model based on quasi-context sensitive languages. Pattern Recogn. Lett. **26**, 1011–1018 (2005). Elsevier, Amsterdam
14. Jurek, J.: Syntactic Pattern Recognition with the Help of GDPLL(k) Grammars (in Polish). Jagiellonian University Publishers, Cracow (2005)
15. Jurek, J.: Grammatical inference as a tool for constructing self-learning syntactic pattern recognition-based agents. LNCS, vol. 5103, pp. 712–721. Springer, Heidelberg (2008). doi:10.1007/978-3-540-69389-5
16. Pavlidis, T.: Structural Pattern Recognition. Springer, New York (1977)
17. Specht, D.F.: Probabilistic neural networks. Neural Netw. **3**, 109–118 (1990)
18. Tanaka, E.: Theoretical aspects of syntactic pattern recognition. Pattern Recogn. **28**, 1053–1061 (1995)

# Determining of an Estimate of the Equivalence Relation on the Basis of Pairwise Comparisons

Leszek Klukowski(✉)

Systems Research Institute Polish Academy of Sciences,
Newelska 6, 01-447 Warsaw, Poland
Leszek.Klukowski@ibspan.waw.pl
http://www.ibspan.waw.pl

**Abstract.** The paper presents two approaches for solving of a discrete programming problem necessary for determining of estimates of the equivalence relation on the basis of pairwise comparisons with random errors. The problem minimizes sum of absolute differences between relation form and comparisons. The problem is NP hard and can be solved with the use of exact algorithms for moderate size of sets - about 50 elements. In the case of larger sets, at least 100 elements and/or multiple comparisons, it is necessary to apply heuristic algorithms. The paper presents original results: a statistical preprocessing, based on two tests proposed, which allows determining of the optimal or suboptimal solution for large sets with acceptable computational cost.

**Keywords:** Estimation of the equivalence relation · Pairwise comparisons · Nearest adjoining order idea

## 1 Introduction

The estimators of equivalence relation based on multiple pairwise comparisons with random errors, proposed in Klukowski [8,9], require optimal solutions of a discrete programming problem. The problem minimizes differences between relation form, determined in appropriate way, and comparisons. The estimates are consistent, under non-restricted assumptions about comparisons errors; the speed of convergence is of exponential type [8] - for increasing number of comparisons of each pair. The optimization problems can be solved with the use of appropriate algorithms: the complete enumeration – for sets including not more than several elements, discrete mathematical programming – up to 50 elements (assuming single comparison of each pair), heuristic approach - for sets exceeding 50 elements. Heuristic algorithms reduce computational costs but can be not effective, especially in the case of probabilities of comparisons errors not close to zero. However, large number of elements is advantageous - it allows application of a statistical preprocessing (two statistical tests), which generate new comparisons with significantly reduced probabilities of errors. This features

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_10

make the approach, based on nearest adjoining order idea [2,11], highly efficient and applicable for any size of a set. The paper consists of five sections. The second section presents the estimation problem, assumptions about distributions of errors of pairwise comparisons and the form of the estimator. In the third section are described concisely well-known exact optimization problems for equivalence relation, suitable for the sets with moderate number of elements. Next section presents new topics: statistical tests generating pairwise comparisons with reduced probabilities of errors and the idea of algorithm proposed. Last section summarizes the results.

## 2 Estimation Problem, Assumptions About Comparisons, Form of Estimator

### 2.1 Estimation Problem

We are given a finite set of elements $\mathrm{X} = \{x_1, \ \ldots, \ x_m\}$ $(3 \leqslant m < \infty)$. It is assumed that there exists in the set $\mathbf{X}$ the equivalence relation, i.e.: reflexive, transitive, symmetric. The relation generates some family of subsets $\chi_1^*, \ \ldots, \ \chi_n^* \quad (n \geqslant 2)$; each subset includes equivalent elements only. The family $\chi_1^*, \ \ldots, \ \chi_n^*$ have the following properties:

$$\bigcup_{q=1}^{n} \chi_q^* = \mathrm{X}, \quad \chi_r^* \cap \chi_s^* = \{0\}, \tag{1}$$

where: $\{0\}$ - the empty set,

$$x_i, x_j \in \chi_r^* \equiv x_i, x_j \text{ - equivalent elements,} \tag{2}$$

$$(x_i \in \chi_r^*) \wedge (x_j \in \chi_s^*) \equiv x_i, x_j \ \ (i \neq j, r \neq s) \text{ - non-equivalent element.} \tag{3}$$

The relation defined by (1)–(3) can be expressed, alternatively, by the values $T(x_i, x_j) \quad ((x_i, x_j) \in \mathrm{X} \times \mathrm{X})$:

$$T(x_i, x_j) = \begin{cases} 0 \; if \; exists \;\; r \; such \; that \;\; (x_i, x_j) \in \chi_r^*, \\ 1 \; otherwise. \end{cases} \tag{4}$$

### 2.2 Assumptions About Distributions of Errors of Pairwise Comparisons

The relation $\chi_1^*, \ \ldots, \ \chi_n^*$ is to be determined (estimated) on the basis of $N$ $(N \geqslant 1)$ comparisons of each pair $(x_i, x_j) \in \mathrm{X} \times \mathrm{X}$; any comparison $g_k(x_i, x_j)$ $(k = 1, \ldots, N)$ evaluates the value of $T(x_i, x_j)$ and can be disturbed by a random error. The following assumptions are made: A1. The number of subsets $n$ is unknown. A2. The probabilities of errors $g_k(x_i, x_j) - T(x_i, x_j) \quad (k = 1, \ \ldots, \ N)$ have to satisfy the following assumptions:

$$P(g_k(x_i, x_j) = T(x_i, x_j)) \geq 1 - \delta \ \ (\delta \in (0, 1)), \tag{5}$$

$$P(g_k(x_i,x_j)=T(x_i,x_j))+P(g_k(x_i,x_j)\neq T(x_i,x_j))=1. \tag{6}$$

A3. The comparisons $g_k(x_i,x_j)$ $((x_i,x_j)\in \mathrm{X}\times \mathrm{X};\ k=1,\ldots,\ N)$ are independent random variables. The assumptions A2–A3 reflect the following properties of distributions of comparisons errors: • each probability of a correct comparison is greater than of incorrect one (inequalities (5), (6)), • zero is the median (in "sharp" form) and the mode of each distribution of comparison error, • the comparisons are realizations of independent random variables, • the expected value of any error can differ from zero. They are weaker than in the literature - see David 1988. Such the properties have comparisons resulting from a statistical test.

### 2.3 The Form of Estimator

The estimator presented in Klukowski ([8], Chap. 3, [9]), is based on the total sum of absolute differences between relation form (values $T(x_i,x_j)$) and comparisons $g_k(x_i,x_j)$ $((x_i,x_j)\in\ \mathrm{X}\ \times\ \mathrm{X})$. The estimates will be denoted $\hat{\chi}_1,\ \ldots,\ \hat{\chi}_{\hat{n}}$ or $\hat{T}(x_i,x_j)$. They are obtained on the basis of the discrete minimization problem:

$$\min_{\chi_r,\ \ldots,\ \chi_r\in F_{\mathrm{X}}}\{\sum_{<i,j>\in R_m}\sum_{k=1}^{N}|g_k(x_i,x_j)-t(x_i,x_j\ \ )|\}, \tag{7}$$

where: $F_{\mathrm{X}}$ - the feasible set: the family of all relations $\chi_1,\ \ldots,\ \chi_r$ in the set $\mathbf{X}$, $t(x_i,x_j)$ - the values describing any relation $\{\chi_1,\ \ldots,\ \chi_r\}$ from $F_{\mathrm{X}}$, $R_m$ - the set of the form $R_m=\{<i,j>\mid 1\leqslant i,j\leqslant m;\ j>i\}$.

### 2.4 Properties of Estimators

The analytical properties of the estimates, resulting from (7) are based on the random variables: $\sum_{R_m}\sum_k|g_k(x_i,x_j)\ -T(x_i,x_j\ \ )|$. The following results have been obtained by the author [8]:

(i) the expected values: $E(\sum_{R_m}\sum_k|g_{\ k}(x_i,x_j)-T(x_i,x_j\ \ )|)$ and $E(\sum_{R_m}\sum_k |g_k(x_i,x_j)-\tilde{T}(x_i,x_j)|)$, i.e. corresponding – respectively - to actual and any other relation $\tilde{T}(x_i,x_j)$, satisfy the inequality:

$$E(\sum_{<i,j>\in R_m}\sum_{k=1}^{N}|g_k(x_i,x_j)-T(x_i,x_j)|)$$
$$<E(\sum_{<i,j>\in R_m}\sum_{k=1}^{N}|g_k(x_i,x_j)-\tilde{T}(x_i,x_j)|); \tag{8}$$

(ii) the variances of the above random variables divided by the number of comparisons $N$ converge to zero, as $N \to \infty$, i.e.:

$$\lim_{N\to\infty} Var(\tfrac{1}{N} \sum_{<i,j>\in R_m} \sum_{k=1}^{N} |g_k(x_i,x_j) - T(x_i,x_j)|) = 0,$$

$$\lim_{N\to\infty} Var(\tfrac{1}{N} \sum_{<i,j>\in R_m} \sum_{k=1}^{N} |g_k(x_i,x_j) - \tilde{T}(x_i,x_j)|) = 0; \qquad (9)$$

(iii) the probability of the inequality $\sum_{R_m} \sum_k |g_{\ k}(x_i,x_j) - T(x_i,x_j\ )| < \sum_{R_m} \sum_k |$
$g_k(x_i,x_j) - \tilde{T}(x_i,x_j\ )|$ converges to one, as $N \to \infty$, i.e.:

$$\lim_{N\to\infty} P( \sum_{<i,j>\in R_m} \sum_{k=1}^{N} |g_k(x_i,x_j) - T(x_i,x_j)| < \sum_{<i,j>\in R_m} \sum_{k=1}^{N} |g_k(x_i,x_j) - \tilde{T}(x_i,x_j)|) = 1, \qquad (10)$$

moreover:

$$P( \sum_{<i,j>\in R_m} \sum_{k=1}^{N} |g_k(x_i,x_j) - T(x_i,x_j)| < \sum_{<i,j>\in R_m} \sum_{k=1}^{N} |g_k(x_i,x_j) - \tilde{T}(x_i,x_j)| \geqslant 1 - \exp\{-2N(\tfrac{1}{2} - \delta)^2\} \qquad (11)$$

(inequality (12) is based on the Hoeffding [6] inequality).

The relationships (i)–(iii) guarantee consistency and fast convergence to actual relation.

## 3 Optimization Problems for the Equivalence Relation

The optimal solutions of the problem (3) can be obtained with the use of the discrete optimization algorithms, applied also in cluster analysis. The discrete algorithms are presented: in Hansen et al. [5], Hansen, P., Jaumard, B. [4], Chopra, R. Rao, M.R. [1], Gordon [3]. An initial approach [10] has a form:

$$\min\{\sum_{j=1}^{n} \sum_{k=1}^{m} \sum_{l=1}^{m} d_{kl}\ z_{kj}\ z_{lj}\} \qquad (12)$$

$$\sum_{j=1}^{n} z_{kj}\ = 1\ (k = 1, \ldots, m), \qquad (13)$$

$$z_{kj} \in \{0,\ 1\} \quad (j = 1,\ \ldots,\ n;\ \ k = 1,\ \ldots,\ m), \tag{14}$$

where: $d_{kl}$ - distance (dissimilarity) between elements $x_k, x_l$, $z_{kj}$ - decision variable equal 1 if an element $x_k$ is assigned to $j$-th cluster, zero otherwise. The problem (12)–(14) has quadratic criterion function, linear constraints and zero-one variables. It can be applied for single comparison of each pair - distances $d_{kl}$ ought to be replaced by comparisons $g_k(x_i, x_j)$; the optimal solution $z^*_{kj}$ determines the form of $n$ subsets. The problem can be applied also for the case $N > 1$ using median from comparisons $g_1(x_i, x_j),\ \ \ldots,\ \ g_N(x_i, x_j)$. The problem (12)–(14) is hard to solve in original form and other approaches have been proposed, e.g. the problem of the minimum weight equivalence relation [4,5]:

$$\min\{\sum_{k=1}^{m-1} \sum_{l=k+1}^{m} d_{kl}\ z_{kl}\} \tag{15}$$

$$z_{kj} + z_{lq} - z_{kq} \leqslant 1 \quad (k = 1,\ \ldots,\ m), \tag{16}$$

$$- z_{kj} + z_{lq} + z_{kq} \leqslant 1 \quad (l = k+1,\ \ldots,\ m-1), \tag{17}$$

$$z_{kl} + z_{lq} + z_{kq} \leqslant 1 \quad (q = l+1,\ \ldots,\ m-1), \tag{18}$$

$$z_{kj} \in \{0,\ 1\} \quad (k = 1,\ \ldots,\ n-1;\ \ l = k+1,\ \ldots,\ m). \tag{19}$$

The problem (15)–(19) can be solved with the use of the dual linear relaxation and revised simplex algorithm. However, the approach need not always provide optimal solution and other approaches have been developed too (see Hansen et al. [5]; Hansen, P., Jaumard, B. [4]); in general, they can be used for the number of elements not (significantly) greater than 50.

## 4 The Algorithm Based on Test Reducing Probabilities of Errors

The problem (7) can be effectively solved with the use of heuristic algorithms in the case of probabilities of errors close to zero. Such the probabilities indicate low fraction of incorrect comparisons - their expected value is equal $(m(m-1)/2)\delta N$. Large number of elements, i.e. $m \geqslant 100$, together with multiple comparisons of each pair ($N > 1$) or $m \geqslant 200$, allows obtaining of "new" comparisons with significantly lower probabilities of errors than $\delta$. The base for such comparisons are statistical tests which verify identity of distributions of "parallel" comparisons, i.e.: $g_k(x_i, x_1)$ and $g_k(x_r, x_1), \ldots, g_k(x_i, x_m)$ and $g_k(x_r, x_m)$ $(k = 1,\ \ldots,\ N; r \neq i)$. The null hypothesis has a form $H_0$: all comparisons $g_k(x_i, x_j)$ and $g_k(x_r, x_j)$ $(k = 1,\ \ldots,\ N;\ \ r \neq i, j;\ \ i \neq j)$ have the same distributions, under alternative $H_1$: some of these comparisons have different distributions. The hypotheses can be replaced by: $H_0$: $x_i$,$x_r$ are equivalent and $H_1$: $x_i$,$x_r$ are not equivalent. The test statistic, proposed by the author, is based on values of the comparisons $g_k(x_i, x_j)$ and $g_k(x_r, x_j)$ $(k = 1,\ \ldots,\ N;\ \ r \neq i, j)$; it has, for $(m-1)N \geqslant 200$, Gaussian limiting distribution. The test allows determining of both errors; it is proper to fix them on similar level. It is clear that such the test reduces significantly the probability of error $\delta$.

### 4.1 The Test for Equivalency of Elements

The test proposed is based on random variables:

$$\eta_{irjk} = \begin{cases} 1 \; if \; g_k(x_i, x_j) = g_k(x_r, x_j), \\ 0 \; if \; g_k(x_i, x_j) \neq g_k(x_r, x_j). \end{cases} \quad (k = 1, \ldots, N; \;\; r \neq i, j) \tag{20}$$

The parameters of these (zero-one) variables are as follows: the expected value assumes, under $H_0$, the form:

$$E(\eta_{irjk} \mid H_0) = (1-\delta)^2 + \delta^2 \quad (r \neq i, j; \;\; j \neq i), \tag{21}$$

the variance – the form:

$$Var(\eta_{irjk} \mid H_0) = 2\delta(1 - 3\delta + 4\delta^2 - 2\delta^3). \tag{22}$$

If $H_1$ is true the parameters of the variable $\eta_{irjk}$ assume the form:

$$E(\eta_{irjk} \mid H_1) = 2\delta(1-\delta) \text{ and } Var(\eta_{irjk} \mid H_1) = 2\delta(1 - 3\delta + 4\delta^2 - 2\delta^3) \tag{23}$$

It is obvious that:

$$E(\eta_{irjk} \mid H_0) = (1-\delta)^2 + \delta^2 > E(\eta_{irjk} \mid H_1) = 2\delta(1-\delta) \tag{24}$$

and that the difference of both expressions is equal: $1 - 4\delta(1-\delta)$. The same parameters can be determined for the variables $\eta_{irik}$ $(k = 1, \ldots, N)$, i.e. for $j = i$. They assume the form:

$$E(\eta_{irik} \mid H_0) = 1 - \delta, \quad Var(\eta_{irik} \mid H_0) = \delta(1-\delta), \tag{25}$$

$$E(\eta_{irik} \mid H_1) = \delta, \quad Var(\eta_{irik} \mid H_1) = \delta(1-\delta). \tag{26}$$

The variables $\eta_{irik}$ have higher expected value and lower variance than the variables $\eta_{irjk}$ $(j \neq i)$. The above results show that the distributions of the variables:

$$\tfrac{1}{(m-1)N} \sum_{r \neq i,j} \sum_{k=1}^{N} E(\eta_{irjk} | \, H_0) \text{ and } \tfrac{1}{(m-1)N} \sum_{r \neq i,j} \sum_{k=1}^{N} E(\eta_{irjk} | \, H_1) \tag{27}$$

are not the same: the expected value of the variable corresponding to $H_1$ is lower, while the variances of both variables are the same. Thus, the null hypothesis can be formulated in the form:

$$H_0 : \sum_{r \neq i,j} \sum_{k=1}^{N} E(\eta_{irjk}) = N(m-1)((1-\delta)^2 + \delta^2) + N(1-\delta), \tag{28}$$

the alternative:

$$H_1 : \sum_{r \neq i,j} \sum_{k=1}^{N} E(\eta_{irjk}) < N(m-1)((1-\delta)^2 + \delta^2) + N(1-\delta); \tag{29}$$

The variance of both variables is equal:

$$Var(\sum_{r\neq i,j}\sum_{k=1}^{N}\eta_{irjk}|H_0) = Var(\sum_{r\neq r,j}\sum_{k=1}^{N}\eta_{irjk}|H_1)$$
$$= 2(m-1)N\delta(1-3\delta+4\delta^2-2\delta^3)+N\delta(1-\delta). \tag{30}$$

The test statistics for null hypothesis assumes the form:

$$N((1-\delta)^2+\delta^2+\tfrac{1-\delta}{m-1},\ \tfrac{1}{(m-1)N}2\delta(1-3\delta+4\delta^2-2\delta^3)+\tfrac{\delta(1-\delta)}{(m-1)^2N}). \tag{31}$$

The test has one sided rejection region, i.e. values lower than the value, corresponding to assumed significance level $\alpha$. The expected value of the statistics converges to $(1-\delta)^2+\delta^2$, as $m\to\infty$, while the variance to zero, as $m$, $N\to\infty$. It can be shown [7] that estimates based on the problem (7) are also consistent as $m\to\infty$, in the case of constant number of subsets $n$. The example. Let us examine an example: $\delta=0,1$; $m=100$; $N=3$. The difference of expected values $E(\eta_{irjk}|\ H_0)-E(\eta_{irjk}|\ H_1)$ equals 0,64, the variables $E(\eta_{irik}|\ H_0)-E(\eta_{irik}|\ H_1)$ equals 0,8, the variance of the distribution (31) is equal 0,0005 (standard deviation 0,02236). In the case of elements $x_i\in\chi_p^*$ and $x_r\in\chi_q^*$, $(i\neq r,\quad p\neq q)$ included in different subsets, each with 10 elements, the difference of statistics (28) and (29) is equal 0,1244, i.e. more than 5,56 of standard deviation. Therefore, the test based on Gaussian distribution guarantees both probabilities of errors lower than 0,003 and the expected value of incorrect comparisons lower than 15 (total number of comparisons equals 4550). It is clear that actual form of the relation is not known before estimation; therefore, the probability of the second type errors, cannot be determined precisely. Thus, it is rational to determine this probability for the subset $\chi_{\min}^*$ with assumed minimal number of elements. The minimal subset ought to include at least several percent of number $m$. The subsets lower than $\chi_{\min}^*$, ought to be detected and excluded; their elements can be associated with an estimate based on a reduced set, as a next step. The detection of elements from small subsets can be done also on the basis of a statistical test. The null hypothesis assumes the form: $\sum_{j\neq i}T(x_i,x_j)=m-\nu-1$ $(i=1,\ \ldots,m)$ under alternative: $\sum_{j\neq i}T(x_i,x_j)<m-v-1$, where: $\nu$ natural number satisfying $\nu\leqslant\zeta(m-1)$, where $\zeta\geqslant 0,05$. The test can be based on the properties of the statistics: $\frac{1}{N}\sum_{j\neq i}\sum_{k=1}^{N}g_k(x_i,x_j)$. Its expected value and variance can be determined under null hypothesis – they are equal, respectively, $(m-1-\nu)(1-\delta)$ and $(m-1-\nu)\delta(1-\delta)/N$; for the alternative hypothesis the expected value is lower than $(m-1-\nu)(1-\delta)$. In the case $mN\geqslant 200$ the Gaussian asymptotic distribution can be applied. Rejecting of the null hypothesis, for an element $x_i$, means that it does not belong to a small subset; rejecting it to whole set $\mathbf{X}$ indicates lack of small subsets in the estimated relation. The comparisons obtained after above preprocessing, i.e. with low probabilities of errors and without small subsets, are satisfactory for heuristic algorithms performing partitioning or agglomeration of elements. The algorithm proposed below belongs to the second group.

### 4.2 The Form of the Algorithm

The comparisons obtained on the basis of the hypotheses $H_0$ and $H_1$ are denoted $\Gamma = \gamma(x_i, x_j)$ $(< i, j > \in R_m)$. The result $\gamma(x_i, x_j) = 0$ corresponds to $H_0$, while $\gamma(x_i, x_j) = 1$ - to $H_1$. The comparisons $\gamma(x_i, x_j)$ allow determining, for each element $x_i \in$ X, two sets: the first one $\varPsi(x_i)$ comprises indexes of elements indicating acceptance of $H_0$, the second $\varOmega(x_i)$ - indexes indicating rejection of $H_0$. It is clear that equivalent elements $x_i$,$x_j$, have the same sets $\varOmega(x_i) = \varOmega(x_j)$, while the sets $\varPsi(x_i)$, $\varPsi(x_j)$ satisfy the relationship $\varPsi(x_i) - \{j\} = \varPsi(x_j) - \{i\}$. Thus, the algorithm minimizing the function (7) can be based on detection of subsets $\hat{\chi}_r$ $(r = 1, \ \ldots, \ \hat{n})$ with these features or close to them. It can be presented in the following simplified form.

START

**$1^0$.** To exclude small subsets from the estimated relation (using the test from the end of the point 4.1) and to determine the probabilities of errors in the test (28), (29). To verify the null hypothesis $H_0$ (28) for $(x_i, x_j) \in$ X × X under alternative $H_1$ (29), on the basis of comparisons $g_k(x_i, x_j)$, $g_k(x_r, x_j)$ $(k = 1, \ ..., \ N; \ r \neq i, j)$, for assumed probabilities of errors (the results $\varGamma = \gamma(x_i, x_j)$ $(i = 1, \ ..., \ m, \ j \neq i)$). To determine the sets $\varOmega(x_i)$, $\varPsi(x_i)$ for each element $x_i \in \mathbf{X}$ on the basis of results of the test. To determine the upper limit $m_d$ of a difference $\sum_{j \neq i} |\gamma(x_i, x_j) - \gamma(x_r, x_j|$: $m_d = int[2\alpha(1-\alpha)(m-1) + 3((2\alpha(1-\alpha)(1-2\alpha(1-\alpha))(m-1)))^{0,5} + 0,5]$ where: $\alpha$ - significance level in the test $H_0$, int$[z]$ - integer part of $z$.

**$2^0$.** To create subsets $\check{\chi}_s$ $(s = 1, \ ..., \ \check{n})$ from elements of the set $\mathbf{X}$ having the following property: the difference $\sum_{j \neq i} |\gamma(x_i, x_j) - \ \gamma(x_r, x_j)| \leqslant m_d$ for each $x_i, x_r \in \check{\chi}_s$. To determine the value of the criterion function (7) after this operation; the value is denoted $F_{cur}$.

**$3^0$.** To determine the set $\varDelta$, comprising elements of the set $\mathbf{X}$, having significant contribution to the value of criterion function $F_{cur}$, i.e. all elements $x_i$ satisfying the inequality:

$$\sum_{j \neq i} |\check{t}(x_i, x_j) - \gamma(x_i, x_j| > m_h,$$

where:

$$m_h = (m-1)\alpha + 3(\alpha(1-\alpha)(m-1))^{0,5}.$$

If the set $\varDelta$ is empty $(\#\varDelta = 0)$ go to $5^0$.

**$4^0$.** To determine the best relocation of each element of the set $\varDelta$, i.e. into a subset $\check{\chi}_q$ $(1 \leqslant q \leqslant \check{n})$ or new subset $\check{\chi}_{\check{n}+1}$ providing maximal decrease of the function (7). To perform the relocations starting from an element providing maximal decrease of the function (7). If the value $F_{cur}$ has been decreased in this point go to $3^0$.

**$5^0$.** Assume $\check{\chi}_q$ $(q = 1, \ ..., \ \check{n})$ as the estimate $\hat{\chi}_q$ $(q = 1, \ ..., \ \hat{n})$.

END

The above algorithm is composed of two phases. The first phase is agglomeration of elements with the same or similar sets $\varOmega(\cdot)$, $\varPsi(\cdot)$. The level of “similarity” is determined by the maximal value $m_d$ of the difference $(\sum_{j \neq i} |\gamma(x_r, x_j) - \gamma(x_i, x_j)|)$, $((x_i, x_r) \in \chi_q^*$ $(1 \leqslant q \leqslant n))$; it is determined on the basis of the sum of: the expected value $E(\sum_{j \neq i} |\gamma(x_r, x_j) - \gamma(x_i, x_j)|)$ and its three standard deviations (resulting from binomial distribution with probability $\alpha$, i.e. significance level in the test). The second phase is oriented at “improvement” of the estimate obtained. The elements $x_i$ of the estimate $\breve{\chi}_q$ $(q = 1,\ ...,\ \breve{n})$ having significant contribution to the function (7) are detected. The minimal value of the contribution $m_h$ is determined on the basis of expected value $E(\sum_{j \neq i} |\breve{t}(x_i, x_j) - \gamma(x_i, x_j)|)$ $((x_i, x_r) \in \chi_q^*(1 \leqslant q \leqslant n))$ and three standard deviations of the variable. The elements with significant contribution, i.e. $\sum_{j \neq i} |\breve{t}(x_i, x_j) - \gamma(x_i, x_j)| \geqslant m_h$, are relocated to subsets providing a decrease of the function. The phase is finished after exhaustion of such elements. The estimate with the criterion function equal to zero is equivalent to the optimal solution, while with a positive value – can be close or equal to it. The comparisons $\gamma(x_i, x_j)$ having very low probabilities of errors are also usable for discrete programming algorithms. It is intuitive that the algorithm performs nearly perfectly for $\alpha \leq 0,01$ and $\zeta \geq 0,1$ (see point 4.1). It results from the fact that the value $E \sum_{j \neq i} |\gamma(x_r, x_j) - (x_r, x_j)|$ is significantly different in the cases $(x_i, x_r) \in \chi_q^*$ and $(x_i, x_r) \notin \chi_q^*$. The estimate obtained in such a way can be verified with the use of tests stating existence of the relation against randomness of comparisons or equivalency of all elements – see Klukowski [9], Gordon ([3], Chap. 7); verification of individual subsets $\hat{\chi}_r$ $(1 \leqslant r \leqslant \hat{n})$ can be also done with the use of known tests, e.g. Cochran test.

## 5 Concluding Remarks

The paper presents the methods of solving of the optimization problem necessary for obtaining the estimates of the equivalence relation. They are applicable for moderate size $(m \leqslant 50)$ and large sets $((m-1)N \geqslant 200)$. The moderate case can be solved with the use of well-known exact algorithms. Large number of elements indicates another approach – the preprocessing, composed of two statistical tests, generating new comparisons with reduced probabilities of errors; they are the base for the heuristic algorithm proposed in the paper. The result (estimate) is final, if the value of the criterion function approaches zero; the positive value can be accepted as estimate or used as a starting point for exact algorithms. Thus, the approach based on minimization of differences between comparisons and relation form is useful computationally efficient and reliable for any size of the set.

## References

1. Chopra, R., Rao, M.R.: The partition problem. Math. Program. **59**, 87–115 (1993)
2. David, H.A.: The Method of Paired Comparisons, 2nd edn. Charles Griffin, London (1988)
3. Gordon, A.D.: Classification, 2nd edn. Chapman&Hall/CRC, Boca Raton (1999)
4. Hansen, P., Jaumard, B.: Cluster analysis and mathematical programming. Math. Program. **79**, 191–215 (1997)
5. Hansen, P., Jaumard, B., Sanlaville, E.: Partitioning problems in cluster analysis: a review of mathematical programming approaches. In: Diday, E., et al. (eds.) Studies in Classification, Data Analysis, and Knowledge Organization. New Approaches in Classification and Data Analysis, pp. 228–240. Springer, Heidelberg (1994)
6. Hoeffding, W.: Probability inequalities for sums of bounded random variables. JASA **58**, 13–30 (1963)
7. Klukowski, L.: Some probabilistic properties of the nearest adjoining order method and its extensions. Ann. Oper. Res. **51**, 241–261 (1994)
8. Klukowski, L.: Methods of estimation of relations of: equivalence, tolerance, and preference in a finite set. IBS PAN, Series: Systems Research, vol. 69, Warsaw (2011)
9. Klukowski, L.: Estimation of the relations of equivalence, tolerance and preference on the basis of pairwise comparisons. In: Burduk, R., et al. (eds.) CORES 2013. Springer, Cham (2013)
10. Rao, M.R.: Cluster analysis and mathematical programming. J. Am. Stat. Assoc. **66**, 622–626 (1971)
11. Slater, P.: Inconsistencies in a schedule of paired comparisons. Biometrika **48**, 303–312 (1961)

# Classification of Body Regions Based on MRI Log Files

Nadine Kuhnert[1,2](✉), Oliver Lindenmayr[1,2](✉), and Andreas Maier[1](✉)

[1] Pattern Recognition Lab, FAU Erlangen Nuremberg, Erlangen, Germany
Nadine.Kuhnert@siemens.com, Lindenmayr.Oliver@gmail.com,
Andreas.Maier@fau.de
[2] Siemens Healthcare GmbH, Erlangen, Germany

**Abstract.** Every Siemens Magnetic Resonance Imaging (MRI) system consistently writes events into log files while the system is running. The log files and their contents are constantly refined by software developers. This results in different information contents depending on the software version. One information that is missing in some log files is the examined body region. As the body region is crucial for usage analysis, we used pattern recognition methods to estimate the examined body region for software versions not logging it automatically. We learned the examined body region from a set of used MRI acquisition parameters such as grid and voxel size and could classify body region information with a classification rate up to 94.7%. We compared Bayesian Network augmented Naïve Bayes, Decision Trees, and Neural Networks, and found Neural Networks resulting in the best classification rate.

**Keywords:** Classification · Data mining · Pattern recognition · Log file analysis · System usage

## 1 Introduction

Medical imaging devices not only produce medical images but also generate log files. The log file content is developed with every released software version. Thus, the log files provide insight into the detailed, subsequent events of a running system. This information can constitute tremendous added value e.g. for predictive maintenance, trouble shooting, and usage analysis. With the customer's approval, the log files are systematically analysed also for Magnetic Resonance Imaging (MRI) systems. MRI systems produce images of the anatomy and physiological processes of the body. Mostly, only specific body parts are of interest and thus, only those body parts are examined. Especially for usage analysis and understanding requirements to the hardware and software of the system, knowing the examined body region is essential. However, this information is logged automatically only by systems with a software version released later than 2010. Thus, many currently used MRI scanners do not deliver the examined body part automatically. We want to solve this problem using Machine Learning (ML), learn from scans with given examined body parts, and classify those examinations

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_11

where the body region is unknown. Data mining is omnipresent in web usage and more and more data mining is also applied in medical applications. For instance, Asaro et al. [1] used medical record access logs for the identification of non-characteristic user behaviour in order to detect unauthorized access to medical records. Moreover, Gallagher et al. [2] use log file information of medical systems to address security issues as well as operational aspects. Thus, they implemented an audit system that tracks the access information per user. Data mining, as well as ML, describes generating additional knowledge of the given information [3]. Many ML algorithms are shared with concepts used in pattern recognition (PR) and are presented by Duda et al. [4]. Both, ML and PR, include the approach of Neural Networks which process, dispose, and generate new information from given data, similar to the human brain [5]. Furthermore, Decision Tree Learning is a predictive modelling approach by partitioning the example space. E.g. Patil et al. [6] use Decision Trees, Support Vector Machines, and k-Nearest Neighbour for the classification of examination types in interventional X-ray imaging. Also Chen et al. [7] use path analysis, sequential pattern discovery and other methods to clinical information system log files in order to discover patient-specific clinician information needs. With this gained knowledge, they displayed the most decisive patient information and thus, could improve the user interface as well as workflow efficiency. Due to its simplicity and good results Naive Bayes is used in many discussions, compared with other approaches, and modified in different ways [8,9].

## 2 Materials and Methods

In this section, we present the ML approaches Augmented Naïve Bayesian Network (BAN), Decision Tree Learning, and Neural Networks and their solutions to our problem implemented in SAS® Enterprise Miner™. Figure 1 illustrates the main steps. We used 70% of the data set with available body region for training and 30% for validation. Our data set contains 16147 scans recorded in log files from 105 MRI scanners belonging to six different system types. The scans have been executed during a time frame of three days on scanners all over the world.

### 2.1 Preprocessing

All three methods were applied to the same training and validation set after preprocessing the data. First of all, we neglected empty examples as well as

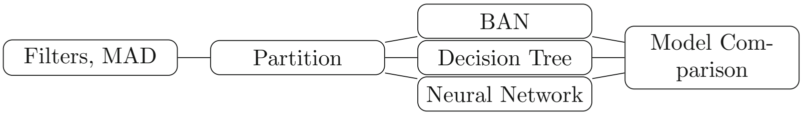


**Fig. 1.** Process chain illustrating the filters, transformation, and partition applied to the training data. The preprocessed data presents the input for the methods BAN, Decision Tree Learning, and Neural Networks. In the final step their results are tested on the validation set and compared.

outliers using the Median Average Deviation [10] and only selected scans with known body region. This yields 15036 scans. Subsequently, we transformed the data using quantile binning and applied stratified partitioning such that we use 70% (10527) for training and 30% (4509) for validation. Afterwards, three in the following described classification methods have been applied to the preprocessed data. We present in Table 1 the distribution of body regions in percent in our training set which constitute the classes. Furthermore, Table 2 shows the parameters given in the log file data which we used for the classification.

**Table 1.** Class-wise distribution of sample data.

| Variable | Distribution (%) |
|---|---|
| Head | 39.53 |
| Spine | 15.85 |
| Heart | 12.80 |
| Abdomen | 8.96 |
| Knee | 8.42 |
| Brain | 4.96 |
| Shoulder | 3.97 |
| Extremity | 2.78 |
| Pelvis | 2.72 |

**Table 2.** MRI parameters used as features for classification of body regions.

| Interval features | Min | Max | Mean | Skewness |
|---|---|---|---|---|
| Echo time | 0 | 107.0 | 24.967 | 1.429 |
| Inversion time | 80 | 1918.0 | 439.779 | 1.359 |
| Repetition time | 2.5 | 5960.0 | 1196.475 | 1.316 |
| Flip angle | 0 | 180.0 | 88.379 | 0.187 |
| Number of images | 1 | 30.0 | 6.853 | 1.641 |
| Slice measurement duration | 15 | 512047.0 | 87005.770 | 0.916 |
| Total scan time | 1 | 714.0 | 121.133 | 1.015 |
| Slice thickness | 0.5 | 13.6 | 5.198 | 0.599 |
| Nominal features | Levels | | | |
| Field of view | 33 | | | |
| Coil | 33 | | | |
| Contrast bolus | 2 | | | |
| Acquisition matrix | 33 | | | |
| Patient position | 4 | | | |

### 2.2 Augmented Naïve Bayesian Network

The first approach is based on the state-of-the-art classifier naïve Bayes and Bayesian networks. Introduced by Friedman et al. [11] as augmented naïve Bayesian network (BAN), the classifier combines the simplicity of naïve Bayes and the ability to cope with independence. A Bayesian network consists of random variables, represented by nodes, and edges connecting the nodes. The edges stand for conditional dependency between two nodes, whereas a naïve Bayesian network only connects every input variable to the attribute class due to the assumption of independence. On the contrary, BAN may have edges connecting the variables with non-class attributes in addition [12]. In this work the maximum number of edges one node is connected with others is restricted to five. This allows a fair amount of dependence between variables but also keeps the search space low. The network structure is built on independence tests (constraint-based) and such that the structure fits the training data best (score-based) [13]. According to the result shown in Table 3, the used classifier has a misclassification rate of 7.1%.

**Table 3.** Result of the augmented naïve Bayesian network in form of a class-wise confusion matrix.

| Actual class | Predicted class | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Spine | Shoulder | Pelvis | Knee | Heart | Head | Extremity | Brain | Abdomen |
| Spine | 645 | 7 | 5 | 0 | 0 | 4 | 7 | 16 | 6 |
| Shoulder | 10 | 120 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| Pelvis | 5 | 2 | 66 | 0 | 0 | 0 | 1 | 1 | 9 |
| Knee | 0 | 0 | 3 | 271 | 0 | 0 | 8 | 0 | 0 |
| Heart | 0 | 0 | 1 | 0 | 401 | 1 | 0 | 0 | 4 |
| Head | 22 | 32 | 7 | 10 | 6 | 1799 | 22 | 230 | 3 |
| Extremity | 1 | 0 | 1 | 0 | 0 | 0 | 56 | 0 | 3 |
| Brain | 14 | 7 | 1 | 0 | 0 | 13 | 0 | 369 | 4 |
| Abdomen | 11 | 5 | 8 | 6 | 14 | 7 | 4 | 1 | 258 |

### 2.3 Decision Tree Learning

Another method with which we used to learn the missing body part information from is called Decision Tree Learning. It learns and makes decisions based on a decision tree. A decision tree consists of one root node describing the input and several branches and leaves. Branches constitute decision rules and leaves represent classes containing a subset of the input. In our implementation, decisions for the division are based on variance reduction. Each node is divided into two branches successively. The maximum depth is ten and the minimal number of entries in one node is set to five [14]. The resulting confusion matrix is given in Table 4 and shows a misclassification rate of 7.0%.

**Table 4.** Result of the Decision Tree Learning in form of a class-wise confusion matrix.

| Actual class | Predicted class | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Spine | Shoulder | Pelvis | Knee | Heart | Head | Extremity | Brain | Abdomen |
| Spine | 657 | 1 | 3 | 0 | 0 | 2 | 1 | 10 | 16 |
| Shoulder | 9 | 116 | 0 | 1 | 1 | 1 | 0 | 1 | 3 |
| Pelvis | 7 | 1 | 63 | 0 | 2 | 0 | 0 | 0 | 11 |
| Knee | 0 | 0 | 0 | 280 | 0 | 2 | 0 | 0 | 0 |
| Heart | 4 | 0 | 1 | 0 | 400 | 0 | 0 | 0 | 2 |
| Head | 10 | 0 | 0 | 4 | 1 | 2074 | 0 | 36 | 6 |
| Extremity | 3 | 0 | 2 | 10 | 0 | 1 | 46 | 0 | 5 |
| Brain | 10 | 0 | 1 | 0 | 3 | 104 | 0 | 283 | 7 |
| Abdomen | 13 | 4 | 7 | 0 | 10 | 1 | 1 | 1 | 271 |

**Table 5.** Result of the Neural Network in form of a class-wise confusion matrix.

| Actual class | Predicted class | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Spine | Shoulder | Pelvis | Knee | Heart | Head | Extremity | Brain | Abdomen |
| Spine | 674 | 0 | 1 | 0 | 0 | 3 | 4 | 5 | 3 |
| Shoulder | 7 | 119 | 0 | 0 | 1 | 1 | 0 | 2 | 2 |
| Pelvis | 1 | 1 | 75 | 0 | 0 | 0 | 1 | 0 | 6 |
| Knee | 0 | 0 | 0 | 280 | 0 | 0 | 2 | 0 | 0 |
| Heart | 0 | 0 | 3 | 0 | 402 | 0 | 0 | 1 | 1 |
| Head | 13 | 0 | 0 | 3 | 0 | 2072 | 0 | 42 | 1 |
| Extremity | 5 | 0 | 2 | 5 | 0 | 0 | 52 | 0 | 3 |
| Brain | 8 | 1 | 0 | 0 | 0 | 96 | 0 | 302 | 1 |
| Abdomen | 7 | 0 | 1 | 0 | 5 | 0 | 0 | 2 | 293 |

### 2.4 Neural Networks

We classified the MRI scans to the nine classes of body regions also using a Neural Network. During the training phase the Neural Network learns and adapts the weights in between the units such that the correct class is assigned depending on the scan parameters [15]. We chose three layers, maximal iteration of 300, and used 14139 weights. We show the class-wise confusion matrix in Table 5 indicating a classification rate of 94.7%. Furthermore, it lists the number of scans assigned to the respective body region and the actual one. It illustrates that the differentiation of brain and head poses the hardest decision with 96 and 42 wrong assignments, and the predominant main diagonal presenting the correct classification. For training the Neural Network, we used the MRI parameters and settings for the individual scans. The chosen coil constitutes the main indicator with an importance of 66.6%. Table 6 lists all used features and their importance to the weights of the Neural Network. Moreover, we performed a ten-fold cross validation for training and validation, separately. We used a data set containing

754707 examples. Figure 2 illustrates the resulting misclassification rates using a bar plot. The horizontal axis carries the 10 folds named F1 to F10, whereas, the vertical axis depicts the misclassification rate in percent. The best rate was reached by F7 with 5.63%, the worst by F5 with 5.88% misclassified scans.

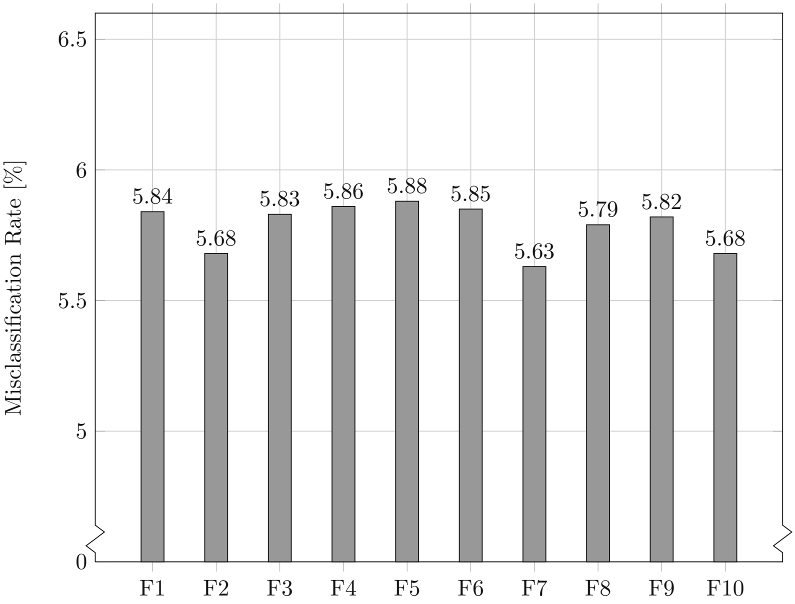


**Fig. 2.** Misclassification rates of a 10-fold cross-validation method.

**Table 6.** The feature importance in percent to the weights of the Neural Network.

| Variable | Importance (%) |
|---|---|
| Coil | 66.6 |
| Flip angle | 26.2 |
| Number of images | 25.9 |
| Patient position | 19.1 |
| Repetition time | 16.6 |
| Echo time | 14.4 |
| Slice thickness | 13.5 |
| Slice measurement duration | 12.9 |
| Field of view | 12.7 |
| Acquisition matrix | 11.9 |
| Total scan time | 8.9 |
| Inversion time | 4.6 |
| Contrast bolus | 4.0 |

**Table 7.** Data sets of 10-fold cross-validation. Number of used examples for training and validation are given next to the misclassification rate (MCR) and the root mean square error (RMSE) per run.

| Folds | Training | Validation | MCR | RMSE |
|---|---|---|---|---|
| Total | 528691 | 226016 | 5.89 | 0.086 |
| 1 | 476033 | 203328 | 5.84 | 0.085 |
| 2 | 475697 | 203491 | 5.68 | 0.084 |
| 3 | 475927 | 203387 | 5.83 | 0.085 |
| 4 | 475845 | 203369 | 5.86 | 0.085 |
| 5 | 475967 | 203330 | 5.88 | 0.085 |
| 6 | 476038 | 203408 | 5.85 | 0.085 |
| 7 | 475714 | 203409 | 5.63 | 0.083 |
| 8 | 475810 | 203345 | 5.79 | 0.085 |
| 9 | 475677 | 203620 | 5.82 | 0.085 |
| 10 | 475511 | 203457 | 5.68 | 0.084 |

The number of examples used for training and validation for every fold is given in Table 7. In addition to the misclassification rate (MCR), it shows the root mean square error (RMSE) with a maximal value of 0.085.

## 3 Discussion

We classified 16147 MRI scans by their examined body regions using BAN, Decision Tree Learning, and Neural Network. The Neural Network delivered the best classification rate of the validation set of 94.7%. BAN classified the scans resulting in a classification rate of 92.9%, we reached 93.0% with the approach of Decision Tree Learning. We showed in Table 6 that the chosen coil for the examination is a main contributor to the correct classification. However, the best classification requires the incorporation of all nine parameters. In future work, the Neural Network should be tested and developed further for other incomplete data sets and parameters in order to counteract inconsistent log file data sets due to different software versions of the systems.

## References

1. Asaro, P., Ries, J.: Data mining in medical record access logs. In: Proceedings of the AMIA Symposium, Washington, DC, p. 855 (2001)
2. Gallagher, R., Sengupta, S., Hripcsak, G., Barrows, R., Clayton, P.: An Audit Server for Monitoring Usage of Clinical Information Systems (1997)
3. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2005)
4. Hart, P., Stork, D., Duda, R.: Pattern Classification. Willey, New York (2001)

5. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall PTR, Upper Saddle River (1998)
6. Patil, M., Patil, R., Krishnamoorthy, P., John, J.: A machine learning framework for auto classification of imaging system exams in hospital setting for utilization optimization. In: 2016 IEEE 38th Annual International Conference of the Engineering in Medicine and Biology Society (EMBC), Orlando, pp. 2423–2426 (2016)
7. Chen, E., Cimino, J.: Automated discovery of patient-specific clinician information needs using clinical information system log files. In: AMIA Annual Symposium Proceedings, Washington, DC, pp. 145–149 (2003)
8. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Mach. Learn. **29**, 103–130 (1997)
9. McCallum, A., Nigam, K.: A comparison of event models for Naive Bayes text classification. In: AAAI-98 Workshop on Learning for Text Categorization, vol. 752, pp. 41–48 (1998)
10. Hampel, F.: The influence curve and its role in robust estimation. J. Am. Stat. Assoc. **69**, 383–393 (1974)
11. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Mach. Learn. **29**, 131–163 (1997)
12. Keogh, M., Pazzani, M.: Learning Augmented Bayesian Classifiers: A Comparison of Distribution-based and Classification-based Approaches (1999)
13. SAS Institute Inc: SAS®Enterprise Miner$^{TM}$14.1 Reference Help (2015)
14. Ville, B., Neville, P.: Decision Trees for Analytics Using SAS Enterprise Miner (2013)
15. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann Publishers, San Francisco (2006)

# Semi-automatic Segmentation of Scattered and Distributed Objects

Muhammad Shahid Farid[1,2](✉), Maurizio Lucenteforte[2],
Muhammad Hassan Khan[3], and Marco Grangetto[2]

[1] College of Information Technology, University of the Punjab, Lahore, Pakistan
shahid@pucit.edu.pk
[2] Dipartimento di Informatica, Università Degli Studi di Torino, Torino, Italy
[3] Pattern Recognition Group, University of Siegen, Siegen, Germany
hassan.khan@uni-siegen.de

**Abstract.** This paper presents a novel object segmentation technique to extract objects that are potentially scattered or distributed over the whole image. The goal of the proposed approach is to achieve accurate segmentation with minimum and easy user assistance. The user provides input in the form of few mouse clicks on the target object which are used to characterize its statistical properties using Gaussian mixture model. This model determines the primary segmentation of the object which is refined by performing morphological operations to reduce the false positives. We observe that the boundary pixels of the target object are potentially misclassified. To obtain an accurate segmentation, we recast our objective as a graph partitioning problem which is solved using the graph cut technique. The proposed technique is tested on several images to segment various types of distributed objects e.g. fences, railings, flowers. We also show some remote sensing application examples, i.e. segmentation of roads, rivers, etc. from aerial images. The obtained results show the effectiveness of the proposed technique.

**Keywords:** Object segmentation · Gaussian mixture model · Graph-cuts

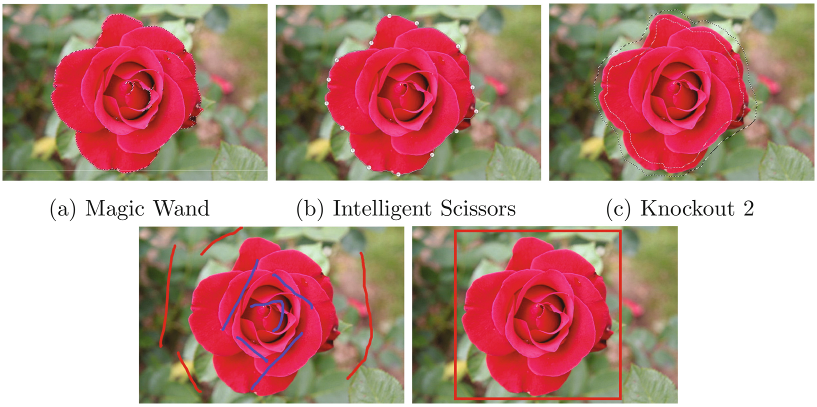
## 1 Introduction

Object segmentation refers to the extraction of a particular object from an image. It is a binary pixel labeling problem that partitions the image into two regions; foreground and background. Various interactive solutions to this problem have been proposed in literature, e.g. [3,7,9,14,23,24,29] and that usually require human user assistance to obtain satisfactory results. User assistance in object segmentation is used to guide the segmentation process and it is usually provided in the form of few scribbles on the target object and on the background. Each segmentation approach requires a different level of user assistance to obtain a neat segmentation. Magic Wand and Lasso tool [1] provided by many image edit-

M. Shahid—The major part of this research was done when the author was associated with institute[2].

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_12

ing tools are considered to be the oldest and the simplest object segmentation techniques. Their accuracy totally depends on the user expertise and usually it takes a huge amount of time to obtain an accurate segmentation. Intelligent scissors [21] is an interactive segmentation tool which exploits the object contours to allows quick and accurate object segmentation. The user provides few seeds on the object boundary and it uses 'live-wire' boundary detection [20] to find the optimal contour between the seed points. Its results are usually satisfactory with adequate user assistance. Many other objects contour based segmentation methods have also been proposed e.g., [11,12,17]. Corel Knockout 2 [2] allows the user to specify the object trimap with the pencil tool and the pixel membership of the unknown region to foreground or background class is decided by computing the ratio of the distance between of the pixel color to the object color and background color. Graph based segmentation [5] has gained widespread popularity due to its superior performance. In graph based segmentation approaches, the image is represented as weighted undirected graph and the segmentation is solved by finding global minimum of the energy function defined over the graph. According to [28], a considerable user effort is required to obtain satisfactory results. A review of well-known graph based image segmentation techniques can be found in [25]. Lazy Snapping [15] is another interactive segmentation which requires the user to mark few lines on the target object and the background. The segmentation is performed through graph cut technique [5]. It shows adequate results, however the user is fatigued in terms of post processing boundary editing which is quite similar to the seeding step in Intelligent scissors method. GrabCut [28] is an iterative solution to graph cut optimization based on initial hard segmentation. Border matting [22] is used to obtain fine segmentation of object boundaries. GrabCut requires the user to draw a rectangle around the target object, the region outside of this rectangle is treated as background. The foreground and the background are represented with Gaussian mixture models (GMM) and the segmentation is obtained through global optimization which can be improved interactively. GrabCut is easy to use, however to refine the segmentation the user may need to feedback the system. The segmentation quality of GrabCut is significantly better than its ancestors. Other well known graph based segmentation approaches are [10,25,26,30–32]. A comprehensive description of energy functions that can be minimized using graph cuts is given in [13] and an overview of graph cuts may be found in [6]. Figure 1 shows the user assistance required in the interactive segmentation techniques described above. Most existing segmentation techniques take trimap as input and are capable to accurately segment a single coherent object. However, these techniques are not effective in segmenting distributed objects which may cover the entire image, e.g. fences, roads, rivers, railings. Indeed, in such cases the definition of the trimap turns to be a very time consuming, tiresome and error prone task. In this paper a novel object segmentation technique is presented to extract objects that are spread on the whole image. The main contributions are:



(a) Magic Wand

(b) Intelligent Scissors

(c) Knockout 2

(d) Graph Cut, Lazy Snapping

(e) GrabCut

**Fig. 1.** User assistance required in existing well-known object segmentation techniques.

- Proposal of a novel segmentation technique to extract distributed or scattered objects;
- Minimal and easy user assistance in the form of few mouse clicks on the target object is required;
- The undesired background is automatically inferred without user intervention and Gaussian mixtures are exploited for primary segmentation;
- A trimap is created automatically and then graph based segmentation is used as the last refinement step;

The effectiveness of the proposed technique is tested on heterogeneous images to segment various types of objects. Moreover, its application in remote sensing to segment the roads, streams and rivers from aerial images is also presented.

## 2 Proposed Object Segmentation Technique

The proposed object segmentation technique is a multi-stage algorithm that takes the user input in the form of few mouse clicks on the target object. Based on this input the color characteristics the target object are estimated using mixture of Gaussians and a rough segmentation is obtained which is then refined through simple morphological operations. In the final stage the fine segmentation is achieved by recasting the problem as graph partition and solving it trough graph-cuts. Let $I$ be an input image and let $P$ be the $n$ points marked by the user on the target object. To achieve better modeling accuracy the sample data is increased by including the $\kappa$-neighboring points around each $p_i \in P$, defined as $\kappa \times \kappa$ square matrix centered at $p_i$, with $\kappa$ odd. It turns out that a total of $\kappa^2 n$ points representative of the target objects are collected. Each pixel is represented with its red, green and blue components in RGB color space.

### 2.1 Estimating Gaussian Mixture Models

Gaussian Mixture Model (GMM) is considered to be a quite general and accurate approach to represent the statistical properties of a data. A GMM is a parametric probabilistic model given by a weighted summation of Gaussian density functions. In particular, the likelihood of a pixel x, given the Gaussian mixture model $\mathcal{G}$ is defined as:

$$p(\mathrm{x}|\mathcal{G}) = \sum_{i=1}^{K} \pi_i g(\mathrm{x} \mid \mu_i, \Sigma_i) \tag{1}$$

where $K$ is the number of Gaussian components, $\pi_i$, $\mu_i$ and $\Sigma_i$ are the weight, mean and covariance of $i$-th Gaussian component with

$$g(\mathrm{x} \mid \mu_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^3 \mid \Sigma_i|}} \exp(-\frac{1}{2}(\mathrm{x} - \mu_i)^\top \Sigma_i^{-1}(\mathrm{x} - \mu_i)) \tag{2}$$

The model parameters $\{\pi_{1,\cdots,K}, \mu_{1,\cdots,K}, \Sigma_{1,\cdots,K}\}$ can be estimated through expectation maximization (EM) algorithm [8,16]. The only limitation of GMM and almost all other clustering approaches is to specify the number of clusters $K$ to use for modeling. To this end, we exploit the Minimum Description Length (MDL) method [27] to estimate the model order. The MDL principal selects the model that results in minimum code length of the data and the model parameters. Indeed, it attempts to minimize the following function $\mathcal{C}$:

$$\mathcal{C}(\mathcal{G}, K) = -\log p(\mathrm{x} \mid \mathcal{G}) + \frac{1}{2} L \log(M) \tag{3}$$

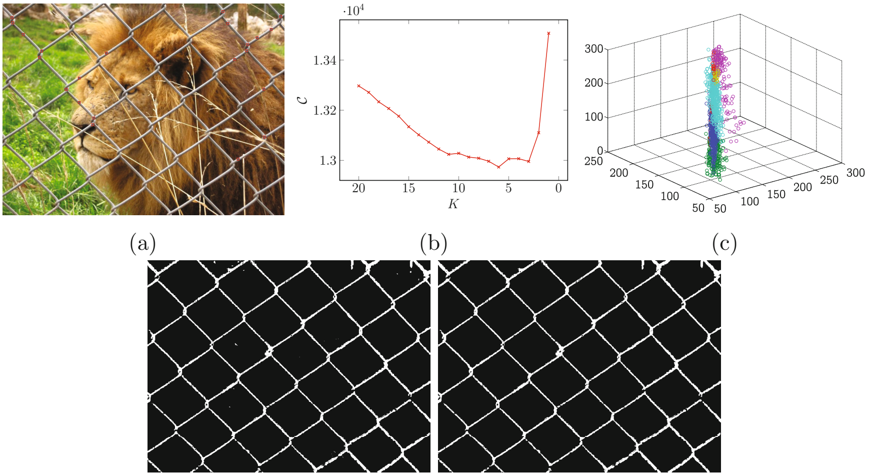
where $L$ is the code length of model parameters $\mathcal{G}$. $M$ is the size of the data i.e., $M = 3\kappa^2 n$ as there are $\kappa^2 n$ points and each is represented with 3 values. The minimum value of $\mathcal{C}$ in Eq. 3 is achieved by minimizing the log-likelihood term (increasing the maximum-likelihood (ML)) and the code length required to describe the model parameters and the data. Starting with $K_0$, a fixed value greater than the number of expected components, EM algorithm is used to estimate $\mathcal{G}$ and MDL is used to find the optimal order. After each iteration, the two most similar clusters are merged and the process is repeated until $K = 1$. Finally, $\mathcal{G}$ and $K$ corresponding to smallest MDL value $\mathcal{C}$ are selected as optimal model parameters.

### 2.2 Initial Segmentation and Rectification

The Gaussian Mixture Model $\mathcal{G}$ is used to obtain the raw segmentation of the target object using Mahalanobis distance [18], computed as:

$$d(\mathrm{x}, \mathcal{G}) = \sum_{i=1}^{K} \pi_i \sqrt{(\mathrm{x} - \mu_i)^\top \Sigma_i^{-1}(\mathrm{x} - \mu_i)} \tag{4}$$

Any given pixel triple x is classified as an object pixel if its distance from the model is lower than a threshold $\tau$, as a background pixel otherwise. Let $\Omega$ be the obtained object mask defined as:



(a) (b) (c)

(d) (e)

**Fig. 2.** (a) Image with user input (red squares), (b) Model order estimation through MDL method with EM algorithm, (c) Cluster wise data distribution in the estimated optimal order GMM, (d) The initial segmentation map (zoom in to see the isolated blobs appeared as a result of false positives), (e) Segmentation map after false positive elimination.

$$\mathit{\Omega}(x,y)=\begin{cases}1 & \text{if } d(I(x,y),\mathcal{G})\leq\tau\\ 0 & \text{otherwise}\end{cases}$$

Here $\mathit{\Omega}$ represents the initial segmentation result. We observed that the initial segmentation may not be perfect; in particular, some non-target pixels may be classified as object pixels (false positives). Furthermore, some object regions, especially in the proximity of object boundaries, may be classified as background. We use morphological operators to eliminate the false positives and graph-cuts to include the false negatives. The false positives in the initial segmentation usually appear as isolated blobs. We use *open* ($\circ$) morphological operator to remove all the blobs with size less than $\gamma$. Figure 2 shows an image with user selected points on the target object, i.e. the fence. The GMM parameters and model order are iteratively estimated starting with $K_0 = 20$ clusters using the approach described in the previous section. Figure 2b shows the value of $\mathcal{C}$ (Eq. 3) for different values of $K$: the minimum occurs at $K = 6$ which represents the optimal number of clusters for this example. Figure 2c shows the cluster wise distribution of the sample points and Fig. 2d shows the initial segmentation map $\mathit{\Omega}$ obtained with threshold $\tau = 4.7$. Small isolated islands of false positives can be seen in Fig. 2d whereas Fig. 2e shows the segmentation map after eliminating them using $\gamma = 30$.

### 2.3 Segmentation Refinement via Graph-Cuts

The false negatives usually appear in the proximity of the object boundaries and inside the segmentation map (see Fig. 2e). To improve the segmentation results we utilize the graph-cut techniques which have emerged as a powerful solution to many optimization problems in computer vision. A rough classification of the image pixels into at least two and ideally three classes is usually the input to a graph cut algorithm. Here we divide the image pixels into three classes namely, *definite foreground* $F$, *definite background* $B$ and *unknown* $U$. The $F$ and $B$ classes are assumed as fixed and known whereas the pixels belonging to $U$ class are to be decided upon. The image is then represented as an undirected weighted graph where the pixels form the vertices and edges link the vertices corresponding to neighbor pixels in the image. Two terminal nodes are added in the graph representing the $F$ and $B$ classes and they are linked to every other node in the graph. The edge weight represents the similarity between its end vertices and they are computed as described in [5]. Finally, the classification of $U$ is performed by the iterative energy minimization through max-flow min-cut algorithm described in [28]. To define the trimap representing the corresponding classes: $F, B$ and $U$, we take the segmentation map obtained from the previous section as $F$ class. Since, the false negatives usually lie in and around the object boundary we use *dilation* ($\oplus$) morphological operator to estimate the 'Unknown' region $U$. The segmentation map $\Omega$ is dilated with structuring element $s$ of size $w$:

$$\Omega_{dil} = \Omega \oplus s$$

The unknown region $U$ region is then computed as:

$$U = \Omega_{dil} \setminus \Omega$$

and region $B$ is computed by negating the dilated mask:

$$B = \texttt{NOT}(\Omega_{dil}) \tag{5}$$

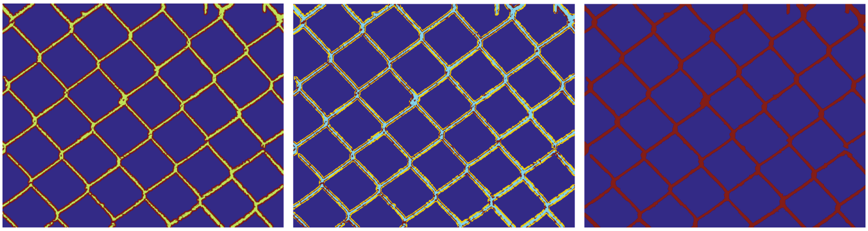

(a) (b) (c)

**Fig. 3.** (a) Trimap: Green color represents the $F$ region, Blue color shows the $B$ region and dark red represents the $U$ region. (b) The $U$ region is partitioned into $F$ (shown in dark red) and $B$ (shown in yellow) using graph-cuts, (c) Final segmentation results.

Figure 3a shows the trimap after dilation of mask (Fig. 2e) with structuring element of size 7 × 7. Figure 3b shows the improvement achieved by graph-cut based refinement. Final segmented object is shown in Fig. 3c.

## 3 Experimental Evaluation and Results

The proposed segmentation technique is tested on images with a variety of distributed objects e.g. fences, bars, grills, wires, roads, rivers, etc. In each experiment the user marks few pixels on the target object. The number of user input

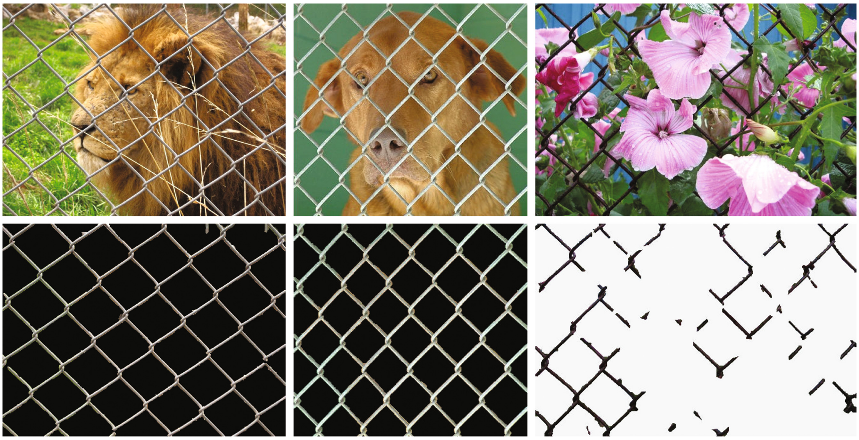


**Fig. 4.** Segmentation of fences.

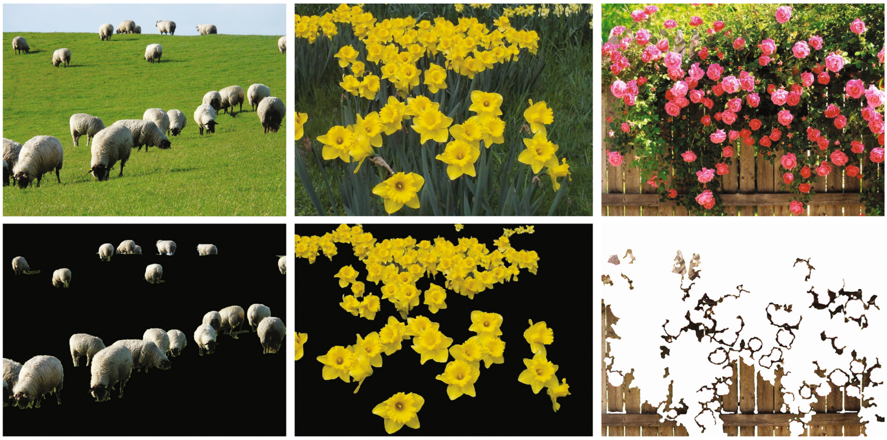


**Fig. 5.** Segmentation of scattered objects. Left: a flock of sheep, middle: a bed of flowers, right: a picket fence.

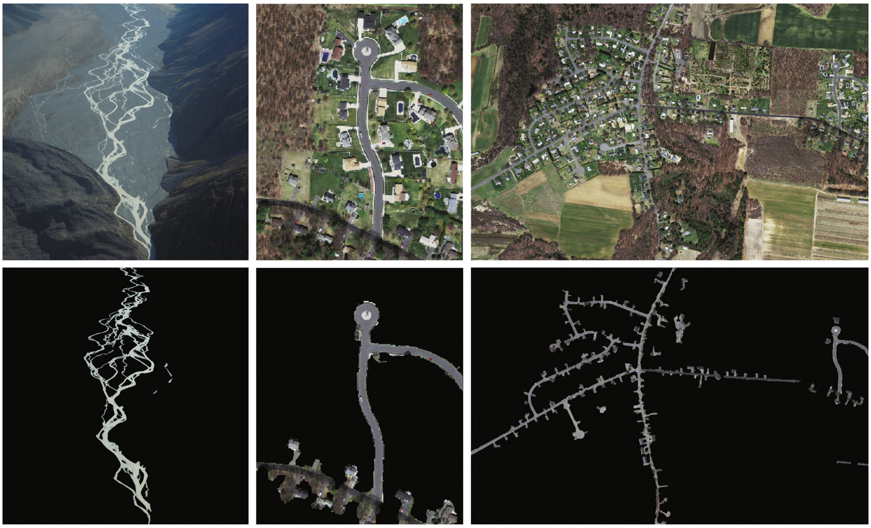


**Fig. 6.** Segmentation of river and road from aerial images. Left: segmentation of the Murchison river. Middle and right: segmentation of road from aerial images (the images are borrowed from Massachusetts Roads Dataset [19]).

points depends on the color characteristics of the target object. From experiment we found that 10 points are sufficient for objects with limited color variation whereas in case of large variation up to 20 clicks can be required. In all experiments the algorithm parameters are interactively tuned to optimize the segmentation results. In particular, $\kappa = 5$ neighboring pixels are selected to increase the sample data and the structuring element size $w$ was set to 7. The clustering is performed using CLUSTER library [4] and $K_0$ in model order estimation is set to 20. Figure 4 shows the segmentation of fence-like-objects from images. Figure 5 shows the segmentation of similar objects scattered in the image. Segmentation of roads and natural water sources e.g. rivers and streams from aerial images is an important problem in remote sensing. The proposed segmentation technique can be effectively used in such problems too. Figure 6 shows the segmentation results of road and rivers from aerial images.

## 4 Conclusions

In this paper an object segmentation technique is proposed to segment objects which are scattered over the whole image e.g. fences, railings, roads, rivers. Since such objects are usually very thin and distributed over a large portion of the image, the conventional segmentation techniques are not effective. The proposed segmentation technique provides accurate and precise segmentation of such objects with minimum and easy user interaction. The obtained results on various distributed objects show the effectiveness of the proposed technique.

## References

1. Adobe Photoshop: Lasso tool. http://helpx.adobe.com/photoshop/using/selecting-lasso-tools.html
2. Berman, A., Dadourian, A., Vlahos, P.: Method for removing from an image the background surrounding a selected object. US Patent 6,134,346, 17 October 2000
3. Beucher, S., Meyer, F.: The morphological approach to segmentation: the watershed transformation. Mathematical morphology in image processing. Opt. Eng. **34**, 433–481 (1993)
4. Bouman, C.: Cluster: An unsupervised algorithm for modeling Gaussian mixtures, April 1997. http://engineering.purdue.edu/bouman
5. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient N-D image segmentation. Int. J. Comput. Vis. **70**(2), 109–131 (2006)
6. Boykov, Y., Veksler, O.: Graph Cuts in Vision and Graphics: Theories and Applications. In: Handbook of Mathematical Models in Computer Vision, pp. 79–96. Springer, US (2006)
7. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23**(11), 1222–1239 (2001)
8. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. J. Royal Stat. Soc. Series B (Methodological) **39**, 1–38 (1977)
9. Farid, M.S., Mahmood, A., Grangetto, M.: Image de-fencing framework with hybrid inpainting algorithm. Sig. Image Video Process **10**(7), 1193–1201 (2016). http://dx.doi.org/10.1007/s11760-016-0876-7
10. Juan, O., Boykov, Y.: Active graph cuts. In: Proceedings of the IEEE Computer Society Conference Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 1023–1029, June 2006
11. Kang, H.: G-wire: a livewire segmentation algorithm based on a generalized graph formulation. Pattern Recognit. Lett. **26**(13), 2042–2051 (2005)
12. Kang, H., Shin, S.: Enhanced lane: interactive image segmentation by incremental path map construction. Graph. Models **64**(5), 282–303 (2002)
13. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? IEEE Trans. Pattern Anal. Mach. Intell. **26**(2), 147–159 (2004)
14. Kuntimad, G., Ranganath, H.: Perfect image segmentation using pulse coupled neural networks. IEEE Trans. Neural Netw. **10**(3), 591–598 (1999)
15. Li, Y., Sun, J., Tang, C.K., Shum, H.Y.: Lazy snapping. ACM Trans. Graph. **23**(3), 303–308 (2004)
16. Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data. Wiley Series in Probability and Statistics, 1st edn. Wiley, New York (1987)
17. Ma, W.Y., Manjunath, B.: EdgeFlow: a technique for boundary detection and image segmentation. IEEE Trans. Image Process. **9**(8), 1375–1388 (2000)
18. Mahalanobis, P.C.: On the generalised distance in statistics. Proc. Nat. Inst. Sci. (India) **2**(1), 49–55 (1936)
19. Mnih, V.: Machine learning for aerial image labeling. Ph.D. thesis, University of Toronto (2013)
20. Mortensen, E., Morse, B., Barrett, W., Udupa, J.: Adaptive boundary detection using 'live-wire' two-dimensional dynamic programming. In: Proceedings of Computers Cardiology, pp. 635–638, October 1992
21. Mortensen, E., Barrett, W.: Intelligent scissors for image composition. In: Proceedings of the SIGGRAPH, pp. 191–198 (1995)

22. Mortensen, E., Barrett, W.: Toboggan-based intelligent scissors with a four-parameter edge model. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR), vol. 2, pp. 452–458 (1999)
23. Mubasher, M.M., Farid, M.S., Khaliq, A., Yousaf, M.M.: A parallel algorithm for change detection. In: 15th International Multitopic Conference (INMIC), pp. 201–208, December 2012
24. Osher, S., Sethian, J.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys. **79**(1), 12–49 (1988)
25. Peng, B., Zhang, L., Zhang, D.: A survey of graph theoretical approaches to image segmentation. Pattern Recogn. **46**(3), 1020–1038 (2013). http://www.sciencedirect.com/science/article/pii/S0031320312004219
26. Peng, B., Zhang, L., Zhang, D., Yang, J.: Image segmentation by iterated region merging with localized graph cuts. Pattern Recogn. **44**(10–11), 2527–2538 (2011)
27. Rissanen, J.: A universal prior for integers and estimation by minimum description length. Ann. Stat. **11**(2), 416–431 (1983)
28. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. **23**(3), 309–314 (2004)
29. Ruzon, M., Tomasi, C.: Alpha estimation in natural images. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 18–25 (2000)
30. Vezhnevets, V., Konouchine, V.: "GrowCut": Interactive multi-label ND image segmentation by cellular automata. In: Proceedings of Graphicon, pp. 150–156 (2005)
31. Vicente, S., Kolmogorov, V., Rother, C.: Graph cut based image segmentation with connectivity priors. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8, June 2008
32. Yang, Q., et al.: Progressive cut: an image cutout algorithm that models user intentions. IEEE Multimedia **14**(3), 56–66 (2007)

# Playback Attack Detection: The Search for the Ultimate Set of Antispoof Features

Maciej Smiatacz(✉)

Faculty of Electronics, Telecommunications and Informatics,
Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland
slowhand@eti.pg.gda.pl

**Abstract.** Automatic speaker verification systems are vulnerable to several kinds of spoofing attacks. Some of them can be quite simple – for example, the playback of an eavesdropped recording does not require any specialized equipment nor knowledge, but still may pose a serious threat for a biometric identification module built into an e-banking application. In this paper we follow the recent approach and convert recordings to images, assuming that original voice can be distinguished from its played back version through the analysis of local texture patterns. We propose improvements to the state-of-the-art solution, but also show its severe limitations. This in turn leads to the fundamental question: is it possible to find one set of features which are characteristic for all playback recordings? We look for the answer by performing a series of optimization experiments, but in general the problem remains open.

**Keywords:** Playback detection · Antispoof algorithms · Biometrics

## 1 Introduction

The growing demand for automatic identity verification methods, which are necessary, for example, to speed up the client authentication process at call-centers providing essential services for banks or telecommunication companies, increased interest in the biometric analysis of human voice signals. Despite the very good performance of state-of-the-art speaker verification systems, their vulnerability to – sometimes quite straightforward – spoofing attacks, remains one of the key problems that prevent their wider use. Main types of such attacks include: impersonation, voice conversion, speech synthesis, the use of artificial signals or the playback of previously recorded speech. The latter does not require any extraordinary skills, specialized knowledge nor sophisticated equipment. It can be treated as a low-effort attack, very easy to implement with the help of ubiquitous mobile devices, and – as we will show in the experimental section – potentially very effective. The users of e-banking services are well aware of the threat related to playback spoofing, and therefore do not trust the new biometric protections. Surprisingly, only limited number of publications address the problem of playback detection. This is most probably caused by the lack of proper dataset, i.e.

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_13

the speech corpus containing both authentic recordings and their recaptured versions. The biggest publicly available database of speaker verification spoofing attempts (ASVspoof 2015 [1]) does not contain replay attacks. In the recent study [2], the playback was emulated using a random mix of three different loudspeaker impulse responses and three different replay environments. Collecting real-life examples of playback attacks is cumbersome and time-consuming; additionally, the infinite number of possible recording conditions and hardware configurations (different types of speakers, microphones, codecs etc.) that might be used by the impostor, usually makes the hard-earned data unrepresentative. As a result, many of the countermeasures lack generalization, and test results are optimistically biased due to overfitting. Application of the challenge-response scenario, in which the user is asked to utter a prompted phrase (for example a randomly generated sequence of digits) might seem to be a natural solution to the problem. This approach, however, affects usability of the system and requires a significant cooperation from the speaker. Moreover, the actual identity verification is more accurate when the voice sample represents the same phrase at each authentication attempt. Consequently, the aforementioned method of playback detection could deteriorate the performance of the biometric protection module. Most of the speaker verification systems process signals captured by telephones. In this case, the microphone is positioned very close to the user; recording from a distance, on the other hand, increases the noise and reverberation levels. Additionally, since the spoofed recording is usually played back through the low-quality speaker of a mobile device, the replayed signal usually contains effects of unwanted high-pass filtering. These two observations inspired the authors of the far-field replay attack detection system [3], one of the earliest countermeasures against this kind of spoofing. The authors of [4] followed a similar approach, trying to find the differences between channel noise patterns that could be used to distinguish between authentic and playback recordings. Another interesting solution relies on voice liveness detection based on the phenomenon of a pop noise, the distortion that appears when human breath reaches a microphone [5]. Unfortunately, many devices such as headsets or telephones eliminate this effect almost entirely. An alternative method utilizes the fact that every authentication attempt is slightly different than the previous ones. If the database already contains a sample that resembles the new recording too closely, we probably experience a playback attack. The authors of [6] proposed to use the landmark points detected in the amplitude spectrum of the speech signal to calculate the similarity score. First, the local maxima, representing peek candidates, are identified in the spectrogram. For each frame only the top-5 strongest spectral maxima are left, other frequency bins are set to zero. The frame is then convolved with a Gaussian window and thresholded. The remaining maxima serve as the so-called seed peaks. A target region is defined around each of them, and then each seed is combined with other peaks from the same region. The pattern representing the constellation of connected peaks is used as a description of the voice signal. The very high compression ratio provided by this representation makes it possible to store large number of authentication attempts effectively.

There are many methods that convert audio data to images – for example power spectral density or mel-frequency cepstral coefficients can be visualized in a form of a two-dimensional graph. Consequently, image processing or computer vision algorithms may be applied to process information represented in this way. The visual domain has been used to classify solo phrases of different musical instruments [7], recognize environmental sounds [8] or evaluate a singer's voice quality [9]. Recently, visual audio features proved to be relatively effective in detecting playback attacks. In the following sections we will propose significant improvements to the state-of-the-art solution based on this approach [2]. At the same time, however, we will point out that despite our ostensible success the problem remains open. The main thesis of this work is that there exists a unique set of texture features, which are detectable in the visual representation of every playback recording, but can rarely be found in the equivalent representation of genuine speech signals. If we were able to formulate even approximate definitions of such features, the training set could be used to tune the parameters of the detector only, and it would be much easier to obtain good generalization despite the small number of training samples. Deep learning techniques would not be helpful in this case, since it is impossible to create a representative and large enough collection of all possible playback attack examples. The feature engineering attempts reported in this paper, however, should only be treated as work-in-progress results.

## 2 Databases

Currently, there are no publicly available databases containing both authentic and playback recordings - the reference solution described in [2] was tested with the help of replay attack simulation realized through the convolution of a source signal with typical impulse responses representing different hardware components. Although it is very difficult to create a representative database of playback attacks, from the practical point of view even limited number of real recordings is, in our opinion, more valuable than the simulation results. In such circumstances independent testing and benchmarking of playback detection algorithms poses serious problems. Therefore, we decided use a new database, containing recordings obtained with different types of mobile phones. The database was prepared by Voicelab, the company providing commercial speech recognition solutions. We hope that in the nearest future we will be able to make the data available to other researchers in order to facilitate comparisons of different antispoofing methods. Courtesy of the authors of [6], we were also allowed to use the PAD corpus that they had prepared. The database contains recordings of 21 (10 male and 11 female) participants of the experiment who uttered the same Polish phrase (*Używam mojego głosu jako klucza*, meaning *I am using my voice as the key*) several times. The samples were captured using the telephone's built-in microphone (Gigaset C610 IP) as well as the two high-quality eavesdropping microphones: one of them (AKG C5) was located up to 10 cm from the mouth of the participant, and the other (Alphard ETP-280) about

50 cm from the speaking person. The eavesdropped recordings were then replayed through a good desktop computer speaker (Creative Inspire T10) to simulate the attack. Two distances (near/far) from the speaker to the telephone were considered. As a result the PAD corpus contains 81 authentic recordings and 4 sets ($4 \times 81 = 324$ samples) of the spoofed versions, related to four combinations of mouth-microphone and speaker-telephone distances. In general, due to the application of relatively advanced equipment the PAD database can be treated as a representation of high-effort attacks. Unfortunately, the quality of the authentic samples is probably too good to reflect real-life conditions, for example the situation when a legitimate customer tries to log into the bank account using a cheap telephone in a noisy environment. The new VL-Bio database contains 160 authentic recordings of 5 persons (4 male, 1 female), captured with 4 different smartphones (LG, Samsung, ZTE and Apple). Two different passphrases (one English and one Polish) were assigned to each device (see Table 1). Each passphrase was repeated 4 times by each participant. While one of the smartphones was capturing the true voice of a given person, transmitted over the GSM network and recorded at the receiver side, the other three used voice recorder applications for eavesdropping. In two out of four cases the impostor's devices were located only 5 cm from the smartphone capturing the genuine authorization data. In other two cases the distance between the eavesdropping devices and the speaking person was set to about 40 cm. Finally, the intercepted recordings were replayed (using the built-in speaker of the eavesdropping smartphone) to simulate the attack; the playback was captured by the same device that previously transmitted the genuine speech. As a result we obtained 480 examples of illegal access attempts. In our setup neither specialized microphones nor high-quality speakers were used, so the attacks are probably easier to detect than in the case of PAD database. On the other hand, the genuine signals are more variable because of the different characteristics of the microphones built into the mobile devices. Additionally, languages and passphrases varied from attack to attack, because our goal was to develop a universal, text-independent playback detector (English pronunciation is poor in some cases but it should not influence the antispoof algorithms performance). In order to study the generalization capabilities of different antispoofing methods, four training subsets were extracted from the VL-Bio database. All of them contained voices of only two persons – Male1 and Female1. The configuration of recording and replaying devices, however, varied from set to set:

- AZS set – genuine authorization attempts were collected using Apple smartphone, ZTE served as a eavesdropping device, Samsung transmitted the spoofed voice (notice that genuine and spoofed samples represent different passphrases),
- LSA set – LG used by the legitimate user, Samsung and Apple by the impostor,
- SLZ set – Samsung used by the legitimate user, LG and ZTE by the impostor,
- ZLS set – ZTE used by the legitimate user, LG and Samsung by the impostor.

**Table 1.** Passphrases assigned to different smartphones, which were used during the creation of our database of playback attacks

| Smartphone brand | Polish passphrase | English passphrase |
|---|---|---|
| LG | Mój głos jest moim hasłem | five, four, zero, seven, one |
| ZTE | Zaloguj mnie do banku | one, one, nine, six, five |
| Samsung | siedem, sześć, pięć, dwa, trzy | My voice is my password |
| Apple | sześć, osiem, dwa, pięć, dwa | Log me into my account |

## 3 Countermeasures

We decided to follow the computer vision approach to the playback detection by converting voice samples to images, which could be afterwards effectively described with the use of typical texture features, such as the well-known Local Binary Patterns [10]. The overall idea is not entirely new and has been recently applied by the authors of the algorithm described in [2], which we will treat as a reference solution and call it the Textrogram method. The method calculates the mel-scaled cepstrogram (time vs. the mel-frequency cepstrum representation of a signal), with appended dynamic features, to form a 2-dimensional image of a speech utterance. Technically, the power cepstrum of a signal is defined as: the squared magnitude of the inverse Fourier transform, of the logarithm of the squared magnitude of the Fourier transform, of a source signal. It is sometimes less formally referred to as spectrum of a spectrum. While the power spectrum (squared magnitude of the Fourier transform) represents the portion of a signal's power falling within given frequency bins and helps to detect main frequency components of a signal, the cepstrum describes the periodicity within the spectrum. For example, harmonics are periodic in the spectrum, and the period corresponds to the pitch. Therefore, investigation of the cepstrum allows for the fundamental frequency estimation, detection of voiced/unvoiced speech and so on. Mel-scale, on the other hand, was defined as a result of human perception analysis, which shows that human ear acts as a filter set concentrating only on certain frequency components. The filters are non-uniformly spaced on the frequency axis (lower frequencies are more important). Thus, to compute the mel-scaled cepstrum the set of triangular filters must be applied to the original power spectrum: the first filter is very narrow, and as the frequencies get higher the filters get wider. The typical mel filter bank contains 26 elements. The authors of the Textrogram method decided to keep 16 mel coefficients resulting from the inverse Fourier transform of the logarithm of the mel-filtered power spectrum. Additionally, they added the differences between mel coefficients from consecutive frames (the so-called *deltas* or differential coefficients), as well as differences of differences (*delta-deltas* or acceleration coefficients), in order to capture the dynamics of mel coefficient changes. The feature vector for each frame was supplemented with the frame energy, i.e. the logarithm of the sum of the power spectrum components. Once the visual representation of the sound is ready, the method applies the Local Binary Pattern transformation to the image. The LBP operator considers a small ($3 \times 3$ in the basic case) neighborhood of each pixel, and performs the binarization in this area, using the intensity of the central point as a threshold. Each of the 8 neighbors is assigned a binary value, 0 or 1, depending on the thresholding result. Finally, the intensity of every pixel is replaced with the 8-bit index of a local binary pattern. The Textrogram method computes a separate histogram for each line of the image and then concatenates the histograms into one feature vector. The so-called uniform codes, i.e. binary sequences containing at most one 0-1 and at most one 1-0 transition, are said to be especially important, thus one LBP histogram consists of 58 bins for individual uniform codes, and additional one for all non-uniform ones. We modified the Textrogram method in the three following ways. (1) We used regular spectrograms instead of mel-scaled cepstrograms, as we did not want to approximate human auditory system, whose playback detection capabilities are quite limited. Mel filtering or secondary Fourier transform possibly eliminate some important features of replayed speech. Our spectrograms are constructed from 8 kHz recordings with the use of 512-point Hamming windows and 256-element overlap. We convert them to 8-bit 257-line bitmaps. (2) We replaced Local Binary Patterns with Local Ternary Patterns (LTP) [11]. LTP operator assigns 0 values to all the neighbors that differ by less than $t$ from the central pixel ($t$ is a parameter), 1 values to neighbors with intensities larger by at least $t$ from the intensity of the central pixel, and $-1$ to all the rest. Our previous research indicates that in most of the cases LTP outperforms LBP [12]. (3) Instead of calculating the histogram of texture descriptors for each line of the image separately, we use 11 uniformly distributed horizontal regions (each is 23 pixels high) to create the feature vector containing $(59 \times 2) \times 11 = 1298$ features (the LTP histogram is twice as big as the LBP equivalent). Before the evaluation of the original Textrogram approach as well as our version of the algorithm (the LTPgram method), we performed initial experiments to test the performance of the antispoof solution based on the combination of far-field detection [3] and comparisons of spectrogram landmarks [6]. The false acceptance rate obtained for VL-Bio database was equal to almost 67%. It is not clear what classification method was used during the tests reported in [2]. The authors mentioned that feature vectors were compared using histogram intersection, and the resulting score was thresholded to make the final decision. They also declared the application Adaboost M1 meta classifier based on decision trees. Finally, no direct classification rates were provided, only the equal error rates and spoofing false acceptance rates obtained for the ASV system based on IV-PLDA and equipped with the Textrogram antispoofing module. In our experiments the Textrogram and LTPgram features were processed by the SVM classifier with $\chi^2$ kernel. The first round of tests involved 10-fold cross validation performed on PAD and VL-Bio databases. Results are summarized in Table 2. As we can see, the LTPgram method outperforms the Textrogram approach, which appeared to be particularly disappointing in the case of PAD database (the algorithm was not able at all to generalize its knowledge about genuine samples). Overall, the results might look like a proof of a major success

**Table 2.** Results of the 10-fold cross validation (FAR – false acceptance rate, FRR – false rejection rate)

| Database | Textrogram | | LTPgram | |
|---|---|---|---|---|
| | FAR (%) | FRR (%) | FAR (%) | FRR (%) |
| PAD | 0 | 100 | 0.31 | 5 |
| VL-Bio | 1.25 | 17.33 | 0.42 | 0 |

**Table 3.** Results of the tests of LTPgram method for training sets (indicated in column headers) substantially different from the testing data

| | AZS | LSA | SLZ | ZLS | VL-Bio | PAD |
|---|---|---|---|---|---|---|
| FAR (%) | 24.79 | 15.42 | 51.46 | 53.75 | 4.58 | 43.83 |
| FRR (%) | 1.94 | 58.71 | 9.68 | 14.84 | 78.71 | 23.45 |

of our modifications. Unfortunately, the second round of tests did not confirm that. This time we used the AZS, LSA, SLZ and ZLS subsets as the training data, and the whole VL-Bio database as the testing set. We also trained the system using the whole PAD database and tested it on entire VL-Bio dataset. Then the roles of PAD and VL-Bio databases were exchanged. Table 3 shows the results. Despite the initial success, the error rates obtained in the second round disqualify our method from being considered as a practical solution to the real-life problem. Even with the modifications that we introduced, the LTP-based computer vision approach to playback detection works properly only when the training set contains samples that closely resemble all kinds of potential attacks. In practice it would be infeasible to create sufficiently large and representative database of this kind.

## 4 The Search for Better Features

The failure reported in the previous section should be probably attributed to the inadequate set of features. Although LTPgrams proved to be significantly better than original Textrograms, their generalization abilities are nevertheless poor. We still claim, however, that it is possible to find a set of texture features, which are characteristic for most of the playback recordings, but not for the genuine speech signals. Better features would make the antispoof solution less dependent on the training set contents. To facilitate some kind of texture feature selection we used the well-known simulated annealing (SA) procedure, according to which one can start with a random solution (feature subset in this case) and then accept new candidates with the probability defined as

$$P(E_C, E_N, T) = 1/\left(1 + \exp((E_N - E_C)/T)\right), \tag{1}$$

where $E_C$ is the so-called energy of the current solution, $E_N$ is the energy of the new candidate (the lower the better), and $T$ denotes the parameter (called a temperature), the value of which decreases throughout the process. The following definitions of the energy are appropriate in our case:

$$\text{(a)}\quad E = \chi^2(\boldsymbol{\mu}_p, \boldsymbol{\mu}_g)^{-1} \qquad \text{(b)}\quad E = \left(\frac{\chi^2\left(\boldsymbol{\mu}_p, \boldsymbol{\mu}_g\right)}{\sigma^2_{\chi p} + \sigma^2_{\chi g}}\right)^{-1}, \tag{2}$$

where $\boldsymbol{\mu}_p$ and $\boldsymbol{\mu}_g$ denote the means of the two classes (playback recordings and genuine samples, respectively), while $\sigma^2_{\chi p}$ or $\sigma^2_{\chi g}$ is the variance of $\chi^2$ distances in the given class. The histogram distance metric, $\chi^2$, is defined as:

$$\chi^2(\mathbf{A}, \mathbf{B}) = \sum_{i,j} w_i \frac{(a_{i,j} - b_{i,j})^2}{a_{i,j} + b_{i,j}}, \tag{3}$$

where $a_{i,j}$ denotes the value of the $j$-th bin in the $i$-th histogram describing image $\mathbf{A}$ and $w_i$ is the weight that indicates the importance of the $i$-th region (by default all weights are equal). The minimization of (2a) means that we just try to push the class means away as far as possible. By using (2b) we look for the biggest ratio of the between-class variance to the sum of within-class variances, in other words we also try to keep the classes compacted. In the first experiment we used SA to adjust weights of the eleven horizontal regions (stripes), for which the LTP histograms were calculated. The procedure was repeated for different datasets, but each time we obtained the same results: all the weights were set to 0 except for the lowest stripe, which was assigned the highest value. On the other hand, when we tried to perform playback detection using this particular stripe only, the results were worse than before. This means that the differences between genuine and spoofed recordings reside mainly in the low frequency region, but they vary considerably between attacks. Next, we applied SA with (2b) energy to find the optimal set of LTP operator thresholds (the $t$ parameters). The resultant values were 15, 15, 10, 5, 3, 15, 15, 3, 3, 3, 5 (top to bottom). This means that larger tolerance is required in the high frequency region, while the analysis of the lower (and, surprisingly some mid-range) frequencies should be more detailed. Table 4 shows the improvements over the original solution, obtained with the new set of thresholds. Finally, we used the SA with (2a) energy to search for the most discriminant subset of 8-bit uniform LTP codes. As a results each histogram contained 30 bins instead of original 118 ones (the feature vectors were shortened to $30 \times 11 = 330$ elements). Improvements achieved this way are summarized in Table 5. In most of the cases the efforts presented above led to some improvements; in our opinion it is justified to say that antispoof systems using optimized thresholds, or the set of selected discriminant texture descriptors, will generalize the knowledge more effectively than the regular LTPgrams (which still perform much better than original Textrograms). As a result, the algorithm's sensitivity to the training set contents became slightly lower. However, bearing in mind very high error rates of the initial experiment (Table 3), we have to admit that we are still far away from finding the universal set of texture features that discriminate original voices from playback recordings.

**Table 4.** Improvements over the results reported in Table 3 after LTP threshold optimization. Negative value indicates that the error was higher than before

| | AZS | LSA | SLZ | ZLS | VL-Bio | PAD |
|---|---|---|---|---|---|---|
| $\Delta$FAR (%) | 7.71 | −5.21 | 3.54 | 5.62 | 20.06 | 0 |
| $\Delta$FRR (%) | 0.65 | 21.29 | 0 | 2.58 | 17.28 | 5.81 |

**Table 5.** Improvements over the results reported in Table 3 after selection of most discriminant LTP codes. Negative value indicates that the error was higher than before

| | AZS | LSA | SLZ | ZLS | VL-Bio | PAD |
|---|---|---|---|---|---|---|
| $\Delta$FAR (%) | −2.50 | 3.54 | 5.63 | 1.87 | 6.48 | 0.62 |
| $\Delta$FRR (%) | −0.64 | 7.74 | 0 | 10.32 | 13.57 | 3.87 |

## 5 Conclusions

The modifications proposed in this paper significantly improved the performance of Textrogram algorithm, the latest antispoof method using computer vision approach to address the problem of playback detection. More importantly, we prepared the new database of real replay attacks and showed that despite our success, the method of LTPgrams is currently not efficient enough to be treated as a truly reliable protection. Therefore we formulate the new challenge: to find the set of the texture features that would characterize most of the playback attack attempts, regardless of the particular type of recording equipment or the presence of environmental sounds. We will continue our research in this area.

**Acknowledgement.** The author would like to thank Tomasz Szwelnik and Jacek Kawalec from Voicelab for fruitful discussions, sharing the expertise and granting access to the VL-Bio database of playback attacks, which was created at the company's laboratories.

## References

1. Wu, Z., Kinnunen, T., Evans, N., Yamagishi, J., Hanilci, C., Sahidullah, M., Sizov, A.: ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In: Interspeech 2015, Dresden, pp. 2037–2041 (2015)
2. Janicki, A., Alegre, F., Evans, N.: An assessment of automatic speaker verification vulnerabilities to replay spoofing attacks. Secur. Comm. Netw. **9**, 3030–3044 (2016)
3. Villalba, J., Lleida, E.: Preventing replay attacks on speaker verification systems. In: IEEE International Carnahan Conference on Security Technology, Barcelona (2011)
4. Wang, Z.-F., Wei, G., He, Q.-H.: Channel pattern noise based playback attack detection algorithm for speaker recognition. In: International Conference on Machine Learning and Cybernetics, vol. 4, Guilin, pp. 1708–1713 (2011)

5. Shiota, S., Villavicencio, F., Yamagishi, J., Ono, N., Echizen, I., Matsui, T.: Voice liveness detection algorithms based on pop noise caused by human breath for automatic speaker verification. In: Interspeech 2015, Dresden, pp. 239–243 (2015)
6. Gałka, J., Grzywacz, M., Samborski, R.: Playback attack detection for text-dependent speaker verification over telephone channels. Speech Commun. **67**, 143–153 (2015)
7. Yu, G., Slotine, J.-J.: Audio classification from time-frequency texture. In: IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, pp. 1677–1680 (2009)
8. Maka, T., Forczmański, P.: Environmental sounds recognition based on image processing methods. In: Burduk, R., Jackowski, K., Kurzyński, M., Woźniak, M., Żołnierek, A. (eds.) Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015. AISC, vol. 403, pp. 723–732. Springer, Cham (2016). doi:10.1007/978-3-319-26227-7_68
9. Forczmański, P.: Evaluation of singer's voice quality by means of visual pattern recognition. J. Voice **30**(1), 127.e21–127.e30 (2016)
10. Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recogn. **29**, 51–59 (1996)
11. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Trans. Img. Proc. **19**, 1635–1650 (2011)
12. Smiatacz, M., Rumiński, J.: Local texture pattern selection for efficient face recognition and tracking. In: Burduk, R., Jackowski, K., Kurzyński, M., Woźniak, M., Żołnierek, A. (eds.) Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015. AISC, vol. 403, pp. 359–368. Springer, Cham (2016). doi:10.1007/978-3-319-26227-7_34

# A Vision-Based Method for Automatic Crack Detection in Railway Sleepers

Ahmad Delforouzi(✉), Amir Hossein Tabatabaei, Muhammad Hassan Khan, and Marcin Grzegorzek

Pattern Recognition Group, University of Siegen, Siegen, Germany
ahmad.delforouzi@uni-siegen.de
http://www.pr.informatik.uni-siegen.de

**Abstract.** In this paper, a method for automatic selection and classification of the sleeper cracks is presented. This method includes three main sequential steps of image pre-processing, sleeper detection and crack detection. Two approaches including rule-based method and template matching method in the frequency domain are proposed for the sleeper detection step. We utilize adaptive threshold binarization to handle challenging crack detection under non-uniform lightening condition and hierarchical structure for the decision making step. Two unsupervised classifiers are exploited to detect the cracks. The results show that the presented method has the overall detection rate with accuracy of at least 87 percent.

**Keywords:** Sleeper crack detection · Adaptive thresholding · Support vector machines · Template matching

## 1 Introduction

Conventionally, the railway sleepers are inspected manually which is an extremely time consuming and labor intensive maintenance task. However to automate this task an integrated system including hardware and software solutions based on a vehicle called "rail mapper" (http://www.igi-systems.com/railmapper.html) is employed to acquire data including capturing the sleeper frames in several locations and further on automatic detection of the sleepers and the possible cracks within. The work presented in this paper is mainly concerning the image processing and pattern recognition methods used in the task of detection and classification of the sleepers and cracks respectively. The presented method consists of three main steps in turn. The latter steps include performing image processing operations, detecting the sleepers and classifying the possible tracks. Throughout the first step, some basic image processing operations are performed to adjust and enhance the captured images. Two different approaches are proposed for the sleeper detection step based on the general rules and template matching. Also an adaptive threshold binarization is used to detect

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_14

some candidates for cracks. Finally an SVM classifier is exploited to automatic detection of cracks.

The structure of this paper is as follows. Section 2 reviews the main related works in the literature and necessary background. Section 3 discusses the approaches and methods used in detecting the sleepers. Section 4 presents the method for detecting and classifying the cracks. Section 5 presents the experimental results. Finally, Sect. 6 concludes the paper.

## 2 Related Work and Background

### 2.1 Related Work

Several approaches exist in the literature for crack detection in the structures like streets, pavements and bridges. For instance in [19] sequential similarity detection algorithm followed by active search (SSDA) is used to detect the cracks fast in high resolution images. In another approach presented in [16] a STRUM (Spatially Tuned Robust Multi-Feature) classifier is used for crack detection in concrete areas over the bridges. They have been able to improve the crack detection results significantly. They used 15 x 15 segmentation to extract spatial features. The features are then forwarded to the svm, random forest and adaboost classifier. Henrique, et. al [14] proposed a method in which a supervised classifier decides non-overlapping image blocks of 75 x 75 pixels whether they contain crack pixels. In the other proposed method, gray processing techniques including peak measurement, the mean value and the weighted average method are used [3]. Despite of several research work in the field of crack detection on different structures, just one is focused on the crack detection on railroad sleeper. To the best of our knowledge, the presented work in [11] is the only available resource which proposes a method for crack detection in wooden sleepers wherein a method based on a binary threshold technique for both sleeper and crack detection of wooden sleepers is presented. The main drawback of this method is that they use a constant binary threshold of 0.65. Thus, the algorithm is very vulnerable to the environmental illumination change. It can just detect very thick cracks within the wooden sleepers.

### 2.2 Background

One of the most commonly used image segmentation methods is image thresholding. The image thresholding methods for image segmentation can be categorized into three groups: 1. Global thresholding, 2. Local thresholding and 3. Adaptive thresholding. In adaptive thresholding which is used here, the desirable foreground image objects is separated from the background based on the difference in pixel intensities of each region. Local adaptive thresholding, selects an individual threshold for each pixel based on the range of intensity values in its local neighborhood. This allows for thresholding of an image whose global intensity histogram doesn't contain distinctive peaks. In adaptive thresholding, a threshold $T(x, y)$ is set individually for each pixel such that the binary image with binary values $b(x, y)$ is generated.

There are many proposed methods to calculate the threshold in the adaptive thresholding method. In local adaptive technique, a threshold is calculated for each pixel, based on some local statistics such as range, variance, or surface-fitting parameters of the neighborhood pixels. It can be approached in different ways such as background subtraction [9], water flow model [13], mean and standard derivation of pixel values [17], and local image contrast [18]. Some drawbacks of the local thresholding techniques are mentioned in the literature as region size dependent, individual image characteristics, and time consuming. Therefore, some researchers use a hybrid approaches that apply both in global and local thresholding methods [4] and some use morphological operations [8]. In other approaches like Niblack [12], and Sauvola and Pietikäinen [17] the local variance technique is used.

In the latter methods, the threshold is calculated based on the local mean $m(x, y)$ and standard deviation $\sigma(x, y)$ within a window of size $w \times w$. Sauvola and Pietaksinen method is an improvement on Niblack's method, especially for stained and badly illuminated document images. The local mean $m(x, y)$ and standard deviation $\sigma(x, y)$ adapt the value of the threshold according to the contrast in the local neighborhood of the pixel. J. Bernsen [2] uses the local gray range technique. In this technique the range between the maximum and minimum pixel gray range within the local window is used to determine the threshold value. In this method, the threshold is set at the midrange value, which is the mean of the maximum and minimum gray values in a local window of size $w \times w$.

## 3 Sleeper Detection

### 3.1 Pre-processing Step

Prior to apply any sleeper detection technique, few pre-processing steps are applied to remove the noise and the adjustment of brightness in the input images. In particular, we apply the contrast adjustment in combination with noise removal and image compression with minimal loss of quality. The pre-processing phase consists of two parts of image resizing by factor of 0.25 to speed up the algorithm and histogram equalization and low-pass filtering for image enhancement.

Two different detection techniques are proposed to detect and segment the sleeper from background. In the first technique, the sleeper is detected using a fast template matching algorithm. In the second technique, a generic model-based method is exploited for sleeper extraction using its geometrical and statistical features. The description of the aforementioned methods is given in the following subsections.

### 3.2 Sleeper Detection Using Template Matching

The proposed template matching detection technique exploits a fast template matching algorithm to locate the sleeper. For matching, we utilized the sum of square differences (SSD) and cross correlation (CC) based matching algorithms [7]. The 2D template used for the sleeper detection, is shown in Fig. 2a. The template is traversed over the entire input image from top-left to bottom-right, and the matching measure is determined on each pixel of the input image which lies under the template. Exhaustive search is performed only in the first frame of the video to find the location of the sleeper which is then used to restrict the search area in the succeeding frames of the video (Fig. 1).

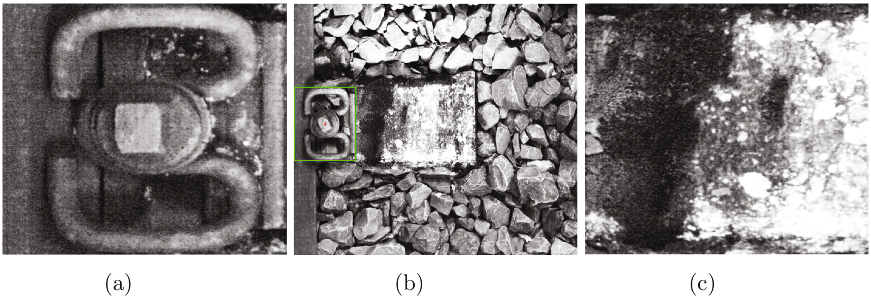


**Fig. 1.** Results of proposed template matching algorithm. (a) template used for sleeper detection, (b) detected sleeper location and (c) segmented sleeper.

Let the input image be as $I$ and template image $T$ with size $h \times w$. In template image, a pixel is represented as $T(i,j)$ where $i$ and $j$ are the coordinate elements of the pixel location. The SSD at $I(x,y)$ can be computed as follows.

$$S(x,y) = \sum_{i=1}^{h} \sum_{j=1}^{w} \big( I(x+i, y+j) - T(i,j) \big)^2 \tag{1}$$

$S$ can also be viewed as squared Euclidean distance between the image patch of $I$ and the template $T$. So, expanding Eq. 1 yields,

$$S(x,y) = \sum_{i=1}^{h} \sum_{j=1}^{w} I^2(x+i, y+j) + \sum_{i=1}^{h} \sum_{j=1}^{w} T^2(i,j) - 2 \sum_{i=1}^{h} \sum_{j=1}^{w} I(x+i, y+j) T(i,j) \tag{2}$$

In Eq. 2, the first term is the sum of squared values in the input image $I$, the second term belongs to the template image, and the third term is twice the correlation between the patch of input image $I$ and the template $T$. Note that the term $\sum_{i=1}^{h} \sum_{j=1}^{w} T^2(i,j)$ is constant and assuming that the term

$\sum_{i=1}^{h}\sum_{j=1}^{w} I^2(x+i, y+j)$ is approximately constant [5]. The remaining term (i.e., cross correlation) is:

$$C(x,y) = \sum_{i=1}^{h}\sum_{j=1}^{w} I(x+i, y+j)T(i,j), \tag{3}$$

The computation cost of cross correlation in spatial domain is quite expensive [10]. To cope with this problem, we compute the cross correlation in frequency domain using the Fourier Frequency Transformation (FFT) and correlation theorem. The theorem states that multiplying the Fourier transform of one function (i.e., input image) by the complex conjugate of the Fourier transform of the other (i.e., the template) gives the Fourier transform of their correlation [6]. That is,

$$C(u,v) = \mathcal{F}^{-1}\big(\mathcal{F}^*(I(x,y))\mathcal{F}(T(i,j))\big), \tag{4}$$

where $\mathcal{F}$ represents the Fourier transform, $\mathcal{F}^*$ represents the complex conjugate and $\mathcal{F}^{-1}$ is the inverse Fourier transform. Although there is no difference in the detection results of the proposed template matching method either computing correlation in spatial domain or frequency domain, it is shown in [10] that the correlation computation in frequency domain is 2.5 to 12 times faster than in spatial domain. The region of the image where the correlation is high, is the place where template and image match well. Figure 2b shows results of the proposed detection algorithm on a sample input image. The red dot in the green rectangle shows the best matching location. The entire sleeper is then extracted on detected location of sleeper using its geometrical features (e.g., height and width).

### 3.3 Sleeper Detection Using Rule-Based Method

In this method, we propose the sleeper detection using its geometrical features. Three different operations are performed sequentially on the images, as shown in the Fig. 2a. In the first step, the input image is converted into binary image. That is, the image pixels are converted to either 0 or 1 based on some threshold $T$. Specifically, Otsu's implementation [15] is exploited for this purpose. However this step can be done using an adaptive thresholding method as well.

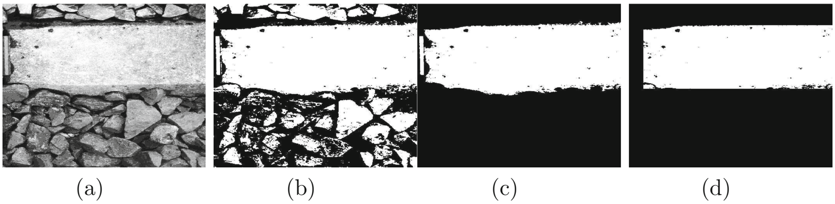


**Fig. 2.** Results of proposed rule-based sleeper detection technique. (a) Original image, (b) binary image, (c) sleeper extraction and (d) boundary specification.

In the second step, the biggest connected object in the binary image is located and extracted from the scene. Finally in the third step, The exact boundary of the sleeper is specified based on five different features to set the size and the shape of the sleeper. It is worth to mention that an optional sequence of morphological operations might be applied prior the second step depending on the image background texture.

## 4 Detection of Cracks on the Extracted Sleepers

Given a sleeper from last section, in this section the detailed description of the proposed crack detection method is presented. The main idea of crack detection is first to generate some crack candidates based on adaptive threshold and then train a supervised classifier to detect the cracks.

The selected adaptive threshold is implemented as follows [1]:

1. Convolving the image with a suitable statistical operator, i.e. the mean or median
2. Subtracting the original from the convolved image
3. Thresholding the difference image with C
4. Applying the threshold on the image and outputting the binary image

Sequentially for a given extracted sleeper image $I$, the proposed crack candidate selection is performed according to the following steps:

1. Applying image enhancement operations: $I_{enh} = F(I)$ ($F$ is the image enhancement function).
2. Applying adaptive thresholding operations: $I_{adp} = G(I_{enh})$ ($G$ is the adaptive threshold function which has been described above).
3. Applying morphological operations: $I_{mor} = H(I_{adp})$ ($H$ is the function for applying morphological opening operations).
4. Applying Connected Component Analysis (CCA) on $I_mor$ for object labeling.
5. Extract the features of $I_{mor}$ as vector $V^t_{I_{mor}} = (v_1, ...v_n)$ (The description of the features is given consequently).
6. Applying pre-defined supervised classifier (support vector machine) on $V_t$.

Let the mask image be given by $T(x, y)$, the list of features to be extracted is as follows.

- Location of the top-left point of each object on the sleeper
- Location of the right-down point of each object on the sleeper
- Area of the objects
- NPDI (Normalized Density Pixel Intensity) of the objects
- Aspect-Ratio of the objects
- Length of the objects

To calculate NPDI, the enhanced image $I_{enh}$ is covered by the mask image $T(x, y)$, then the NDPI of the enhanced image at the exactly same region of the objects in the correspondent mask image is calculated. The following equation, mathematically explains the NDPI for each object.

$$NDPL_i = \sum_i A_i \frac{E_i(x, y)}{A_i}, \quad \text{if} \quad T(x, y) = 1 \tag{5}$$

In the above equation, $E(x, y)$ is the enhanced image of sleeper and $A_i$ is the area of each object. Therefore, for each candidate in the binary image, we have a NDPI number calculated from the correspondent enhanced gray-scale image of sleeper. Sometimes the real crack has been connected to the some noisy candidates. In such a case, the aspect ratio, and the top and down coordinates of the crack can be mistaken. It may cause in rejecting the real crack during the decision process (false negative). To avoid this rejection, a morphological opening operand with the window size of $[10, 1]$ is used to detach the sticking no-crack components from the real crack in the vertical direction. Note that the values of the first two feature are normalized by the length and the width of the sleeper. For the sake of classification, a support vector machine (SVM) classifier is used. The SVM is considered as a powerful tool for solving non-linear classification problems. Due to the multi-dimensionality of the extracted features, the SVM classifier seems to be a proper choice.

## 5 Experimental Results

In this section the experimental results including the intermediate results corresponding to the main steps of the presented method are given. At first, the main sleeper image from the sleeper locating module has been enhanced by the same histogram equalization and low pass filtering with the same parameter as in the sleeper locating module. The result has been shown in Fig. 3(b). Then, the enhanced image is converted to a binary image by using the adaptive threshold method as shown in Fig. 3(c). Then tiny and noisy particles of the resulting binary image are removed by the area circumstances. The de-noised image has been shown in Fig. 3(d). The threshold of area can be set by the user according to the desired sensitivity and classification accuracy. We call the resulting binary image as the mask image. In fact, the mask image indicates some objects (group of white pixels) which can include a crack as well. For the classification purposes, two types of SVM including linear and radial-basis function (RBF) are used. We have used 1284 samples for training and 328 samples for testing purpose. Among the total 1612 samples, 932 samples belong to non-cracked class (negative set) and the rest to the cracked class (positive set). In this experiment the Gamma parameter of RBF-SVM is set to 0.1, which is the experimentally optimized parameter for achieving higher precision and accuracy.

The classification results have been shortlisted in Table 1 below. The maximum false positive rate is less than 2% which shows a satisfactory result.

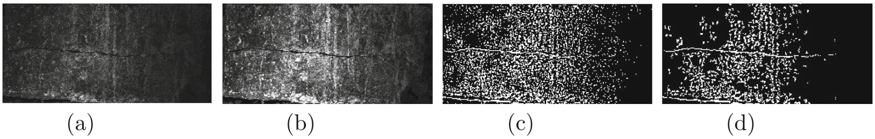


**Fig. 3.** The intermediate results of the candidate generation for crack detection

**Table 1.** Classification results of the supervised classifiers to detect the cracks

| Supervised classifier | True positive | False positive |
|---|---|---|
| Linear SVM | 63.03% | 1.99% |
| RFB-SVM | 96.70% | 1.99% |

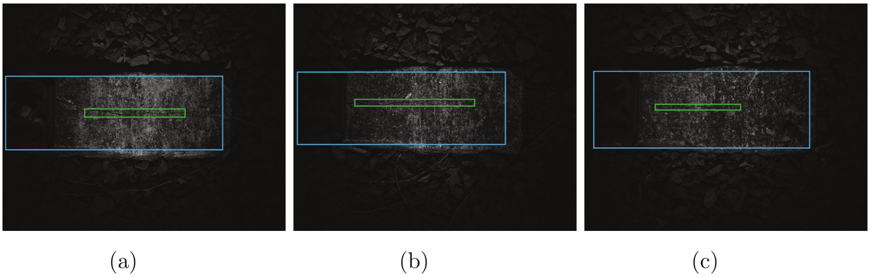


**Fig. 4.** Visualization results of the proposed sleeper and crack detection

According to the above Table 1 using SVM classifier with Radial Basis Function (RBF) as kernel functions will generate the most accurate result. The sleeper detection module can detect the sleepers with accuracy around 90%. Thus the overall accuracy is around 87.03 percent. The successful results of the proposed method on some cracked sleepers have been visualized in Fig. 4. The results of the proposed sleeper and crack detection have been shown by the blue and the green rectangles respectively.

## 6 Conclusion

In this paper an automatic method for crack detection in sleeper inspection system has been presented. The presented method is based on a joint research-industrial work. We exploited a template matching method in frequency domain which uses the sleeper fastener to locate the sleeper in the image. In the crack detection step, we employed an adaptive thresholding method to handle overcome the non-uniform illumination problem. By using a hierarchical structure the number of cracked candidates has decreased. Finally, an SVM classifier is exploited to distinguish the cracks. The proposed method results in an accurate recognition rate which suits the industrial applications in an appropriate way.

## References

1. http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm. Accessed 04 Oct 2016
2. Bernsen, J.: Dynamic thresholding of gray-level images. In: Proceedings of the 8th International Conference on Pattern Recognition, pp. 1251–1255 (1986)
3. Fujita, Y., Mitani, Y., Hamamoto, Y.: A method for crack detection on a concrete structure. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006), vol. 3, pp. 901–904 (2006)
4. Gatos, B., Pratikakis, I., Perantonis, S.J.: Adaptive degraded document image binarization. Pattern Recogn. **39**(3), 317–327 (2006). http://www.sciencedirect.com/science/article/pii/S0031320305003821
5. Goshtasby, A., Gage, S.H., Bartholic, J.F.: A two-stage cross correlation approach to template matching. IEEE Trans. Pattern Anal. Mach. Intell. **6**(3), 374–378 (1984)
6. Khan, M.H., Helsper, J., Yang, C., Grzegorzek, M.: An automatic vision-based monitoring system for accurate Vojta-Therapy. In: Proceedings of the 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pp. 1–6. IEEE (2016)
7. Khan, M.H., Shirahama, K., Farid, M.S., Grzegorzek, M.: Multiple human detection in depth images. In: International Workshop on Multimedia Signal Processing (MMSP). IEEE (2016)
8. Le, T.H.N., Bui, T.D., Suen, C.Y.: Ternary entropy-based binarization of degraded document images using morphological operators. In: 2011 International Conference on Document Analysis and Recognition, pp. 114–118, September 2011
9. Lu, S., Su, B., Tan, C.L.: Document image binarization using background estimation and stroke edges. Int. J. Doc. Anal. Recogn. (IJDAR) **13**(4), 303–314 (2010). http://dx.doi.org/10.1007/s10032-010-0130-8
10. Lyon, D.: The discrete fourier transform, part 6: cross-correlation. J. Object Technol. **9**(2), 17–22 (2010)
11. Mohammad, S.P.: Machine vision for automating visual inspection of wooden railway sleepers. Master's thesis, Department of Computer Science, Dalarna University (2008)
12. Niblack, W.: An Introduction to Digital Image Processing. Strandberg Publishing Company, Birkeroed (1985). pp. 115–116
13. Oh, H.H., Lim, K., Chien, S.I.: An improved binarization algorithm based on a water flow model for document image with inhomogeneous backgrounds. Pattern Recogn. **38**(12), 2612–2625 (2005). http://www.sciencedirect.com/science/article/pii/S0031320305001317
14. Oliveira, H., Correia, P.L.: Automatic road crack detection and characterization. IEEE Trans. Intell. Transp. Syst. **14**(1), 155–168 (2013)
15. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979). http://dx.doi.org/10.1109/TSMC.1979.4310076
16. Prasanna, P., Dana, K.J., Gucunski, N., Basily, B.B., La, H.M., Lim, R.S., Parvardeh, H.: Automated crack detection on concrete bridges. IEEE Trans. Autom. Sci. Eng. **13**(2), 591–599 (2016)
17. Sauvola, J., Pietikäinen, M.: Adaptive document image binarization. Pattern Recogn. **33**(2), 225–236 (2000). http://www.sciencedirect.com/science/article/pii/S0031320399000552

18. Su, B., Lu, S., Tan, C.L.: Binarization of historical document images using the local maximum and minimum. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, DAS 2010, NY, USA, pp. 159–166 (2010). http://doi.acm.org/10.1145/1815330.1815351
19. Yamaguchi, T., Nakamura, S., Hashimoto, S.: An efficient crack detection method using percolation-based image processing. In: 2008 3rd IEEE Conference on Industrial Electronics and Applications, pp. 1875–1880, June 2008

# Towards Privacy-Aware Keyboards

Krisztian Buza[1(✉)] and Piroska B. Kis[2]

[1] Knowledge Discovery and Machine Learning, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany
chrisbuza@yahoo.com
[2] University of Dunaujvaros, Dunaujvaros, Hungary
piros@uniduna.hu
http://www.biointelligence.hu/typing.html

**Abstract.** As shown by various studies, the dynamics of typing on a keyboard is characteristic to persons. On the one hand, this may allow for person identification based on keystroke dynamics in various applications. On the other hand, in certain situations, such as chat-based anonymous helplines, web search for sensitive topics, etc., users may not want to reveal their identity. In general, there are various methods to increase the protection of personal data. In this paper, we propose the concept of *privacy-aware keyboard*, i.e., a keyboard which transmits keyboard events (such as pressing or releasing of a key) with small random delays in order to ensure that the identity of the user is difficult to be inferred from her typing dynamics. We use real-world keystroke dynamics data in order to simulate privacy-aware keyboards with uniformly random delay and Gaussian delay. The experimental results indicate that the proposed techniques may have an important contribution to keeping the anonymity of users.

**Keywords:** Privacy · Keystroke dynamics · Machine learning · Web search

## 1 Introduction

The dynamics of typing is known to be characteristic to persons [1–4]. While person identification based on keystroke dynamics may be desired in many applications, such as internet banking or online tax declaration [5], however, it may happen that the users want to remain anonymous. As an example, let us consider the case of web search. The keywords used when searching the web, may reveal sensitive information about the users and their interests [6]. In particular, searching for particular diseases and symptoms may be an indication of health status, other keywords may allow to infer political or religious views of users, etc. We assume an attacker, who wants to gain access to such sensitive information and wants to link the pieces of information to persons. Obviously, IP-addresses linked to search queries may be highly informative, however, due to

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_15

shared usage of computers and dynamic allocation of IP-addresses, they may not allow to identify users uniquely. Nevertheless, it may be possible to distinguish different users of the same computer (or the same IP-address, respectively) based on their keystroke dynamics. We note that keystroke dynamics may simply be captured by scripts running in the web browser, and the user is not likely to notice the activation of such scripts. In order to contribute to the protection of the private information, we propose the concept of *privacy-aware keyboard*, by which we mean a keyboard that transmits keyboard events (such as pressing or releasing a key) with small random delays in order to ensure that the identity of the user is difficult to be inferred from her typing dynamics. We perform experiments using real-world keystroke dynamics data in order to simulate privacy-aware keyboards choosing the delay from various random distributions such as uniform and Gaussian. The experimental results indicate that the identity of the users is much more difficult to be recognized in case of the privacy-aware keyboard, and therefore the proposed techniques may substantially contribute to the keeping the anonymity of users.

## 2 Problem Formulation

We assume an attacker whose goal is to identify the user who typed a particular text (such as keywords in the aforementioned web search scenario, or sentences in case of chat-based helplines). The attacker is able

- to run a script in the web browser that captures keystroke dynamics, and
- to use similarity-based models in order to compare keystroke dynamics.

On the one hand, we want to prevent the attacker from achieving his goal by the usage of a keyboard that transmits keyboard events with small random delays, causing the captured keystroke dynamics data to become corrupted. On the other hand, we want the corrupted keystroke dynamics data to look "natural", so that the attacker believes that he might be able to identify the user based on that data, and therefore he will *not* use other person identification techniques.

## 3 Our Privacy-Aware Keyboard Models

Next, we define two privacy-aware keyboard models: the Uniformly Random Delay Keyboard and the Gaussian Keyboard. After each keyboard event $e$ (such as pressing or releasing of a key), both types of keyboards chose a random number $d_e$, and wait for $d_e$ milliseconds before transmitting the signal indicating that event $e$ occurred. The keyboards preserve the order of events: that is, in case if event $e_1$ occurs at time $t_1$ and event $e_2$ occurs at time $t_2$ and the random delay $d_1$ was generated for event $e_1$, then the random delay of $d_2$ to be generated for event $e_2$ must fulfill: $t_1 + d_1 < t_2 + d_2$. From which follows that $d_2 > t_1 - t_2 + d_1$. The uniformly random delay keyboard chooses the random number $d_e$ uniformly from the interval $[max(0, t_{pre} - t_e + d_{pre})\ ,\ d_{max}]$, where $t_e$ and $t_{pre}$ are the times (in milliseconds) when event $e$ and the previous event happened, $d_{pre}$ is the random number generated for the previous event and $d_{max}$ is the maximal delay which is a parameter of this keyboard model. The delay $d_e$ associated with event $e$ in case of the Gaussian Keyboard is $max(\mathcal{G}(\mu, \sigma)\ ,\ t_{pre} - t_e + d_{pre})$, where $\mathcal{G}(\mu, \sigma)$ is a random number generated from a Gaussian distribution with mean $\mu$ and standard deviation $\sigma$, which are parameters of this keyboard model. Symbols $t_e$, $t_{pre}$ and $d_{pre}$ denote the same as in case of the Uniformly Random Delay Keyboard.

## 4 Experiments

We begin this section by the introduction of our keystroke dynamics data which we used to simulate privacy-aware keyboards. This is followed by the description of the experimental settings and results.

### 4.1 Typing Dynamics Data

We collected keystroke dynamics data, or *typing patterns* for short, from 12 different users over several months, resulting in a collection of 548 typing patterns in total. In each of the typing sessions[1], the users were asked to type the following short text based on the English Wikipedia page about Neil Armstrong:

*That's one small step for a man, one giant leap for mankind. Armstrong prepared his famous epigram on his own. In a post-flight press conference, he said that he decided on the words just prior to leaving the lunar module.*

In each typing session, we measured the duration of each keystroke, i.e., the time between pressing and releasing a key. We used a self-made JavaScript application and a PHP script to capture the aforementioned time series and to save the data. We mention that the length of typing patterns varies slightly from session to session due to typing errors.

### 4.2 Experimental Settings

The primary goal of our experiments was to show that privacy-aware keyboards indeed make person identification difficult. In order to do that, we simulate privacy-aware keyboard by adding noise to the data according to the privacy-aware keyboard models. Specifically, we will show that the accuracy of person identification decreases dramatically in case of privacy-aware keyboards, concretely, the accuracy in case of privacy-aware keyboards is close to the accuracy of random guessing. Furthermore, in order to allow to draw more general conclusions that are valid to *any* similarity-based algorithm, we analyze the similarities

[1] The number of typing sessions was approximately the same for each user. Despite the fact that the data is balanced, the recognition of the user based on typing dynamics could lead to an imbalanced classification task, for example in case if binary classifiers are used according to the one-vs-rest schema.

between the typing patterns both in case of the same user, as well as in case of different users, both for the original data, and the data "corrupted" by the privacy-aware keyboard models. In principle, one could measure the accuracy of person identification in context of various classifiers, such as neural networks [7,8], Hidden Markov Models [9], ensembles [10–13], or classifiers designed for imbalanced data [14,15], see also [16] for a survey on data stream mining. However, we decided to use nearest neighbor classifiers in our experiments, because keystroke dynamics data are time series and, in case of time series data, the 1-nearest neighbor classifier (1NN) with dynamic time warping (DTW) as distance measure was shown to be competitive with complex models, such as neural networks, Hidden Markov Models or "super-kernel fusion scheme"[17,18]. These empirical results are justified by theoretical analysis as well [19,20]. Thus, 1NN with DTW can be considered as a representative of time-series classifiers. Furthermore, taking into account that 1NN is popular and simple to implement, we can assume that an attacker is likely to use this classifiers for person identification.

For the classification experiments, we used the *first* five typing patterns from each user as training data, and the remaining typing patterns were used as test data. This is consistent with the assumption that the attacker may have access to a few typing patterns from the *past*, while the attacker may not be able to observe the typing dynamics of a user for a very long time without being noticed. We note that the same data and train-test splits are used in the *Person Identification Challenge*.[2] Both for regular and privacy-aware keyboards, we measured classification accuracy, i.e., the proportion of typing patterns for which the user was correctly recognized based on the dynamics of her typing. Additionally to presenting results of 1NN classification, we examine the similarity of typing patterns as well. In particular we show the median, the 10th and 90th percentiles of DTW distances both in case of the original data, as well as in case of the data corrupted according to our privacy-aware keyboard models.

### 4.3 Experimental Results

Figures 1 and 2 show the accuracy of person identification, i.e., the proportion of correctly identified users, for our privacy-aware keyboard models as function of maximal delay $d_{max}$ (in case of the Uniformly Random Delay Keyboard) and $\sigma$ (in case of the Gaussian Keyboard). For comparison, the accuracy in case of a regular keyboard (labeled as "No delay") as well as the accuracy of random guessing[3] are shown as well. As expected, the accuracy of person identification

[2] http://biointelligence.hu/typing-challenge/.

[3] With *random guessing* we mean a naive classifier that works as follows: for each typing pattern $x$ of the test data, is selects one of the users randomly (each user has an equal probability to be selected), and this randomly selected user, denoted as $y_x^{(rnd)}$, is the prediction of the classifier. That is: according to the "guess" of this naive classifier, the typing pattern $x$ belongs to the randomly selected user $y_x^{(rnd)}$. As there are 12 users in our dataset, with a probability of 1/12 the randomly selected user will match the true user associated with the typing pattern, therefore, the accuracy of random guessing is 1/12.

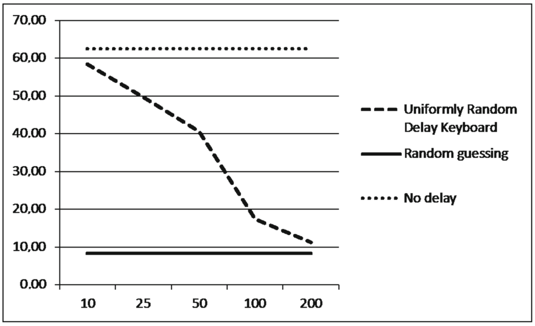


**Fig. 1.** Accuracy (in %) of person identification in case of Uniformly Random Delay Keyboard as function of the maximal delay $d_{max}$. The accuracy in case of a regular keyboard (labeled as "No delay") as well as the accuracy of random guessing are shown for comparison.

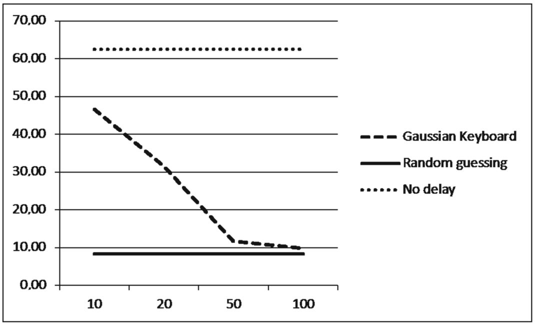


**Fig. 2.** Accuracy (in %) of person identification in case of Gaussian Keyboard as function of $\sigma$ when $\mu = 100$ ms. The accuracy in case of a regular keyboard (labeled as "No delay") as well as the accuracy of random guessing are shown for comparison.

decreases with increasing maximal delay in case of the Uniformly Random Delay Keyboard. Most importantly, already in case of a delay of 200 ms, the accuracy is close to that of random guessing. Taking the speed of typing into account, the delay of 200 ms seems to be acceptable for most of the users. Increasing $\sigma$ in case of the Gaussian Keyboard has a similar effect.

Figure 3 shows the median, 10th percentile and 90th percentile of DTW-distances between typing patterns of the same user and different users in the following cases: (i) regular keyboard, (ii) Uniformly Random Delay Keyboard with $d_{max} = 200$ and Gaussian Keyboard with $\mu = 100$ and $\sigma = 100$. While the DTW-distances are generally larger in case of the privacy-aware keyboards, from the point of view of privacy-aware keyboards, the most important is that the range of distances between typing patterns of the *same* user almost perfectly overlap with the range of distances between the typing patterns of *different* users.

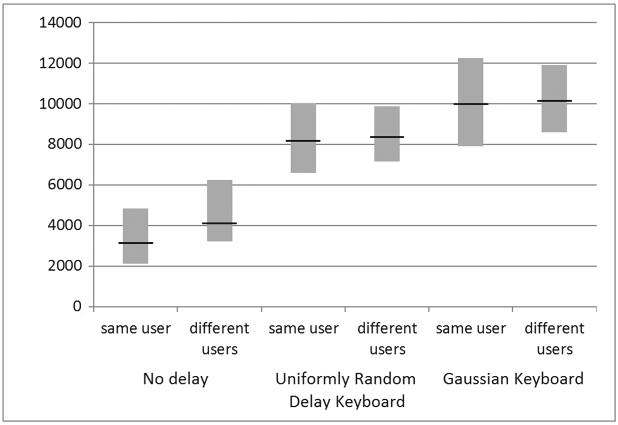


**Fig. 3.** Median, 10th percentile and 90th percentile of DTW-distances between typing patterns of the same user and different users in the following cases: regular keyboard (labeled as "No delay"), Uniformly Random Delay Keyboard and Gaussian Keyboard.

This makes person identification difficult (if not impossible) for *any* model that is based on the distances (or similarities) between typing patterns.

## 5 Conclusions and Outlook

In this paper, we proposed the concept of privacy-aware keyboards, and we discussed two privacy-aware keyboard models. In our experiments, we simulated privacy-aware keyboards by adding noise to real-world keystroke dynamics data. The analysis shows that privacy-aware keyboards indeed make different users' typing patterns to appear much more similar to each other and therefore it becomes difficult for *any* similarity-based algorithm to distinguish users when privacy-aware keyboards are used. We note that typing dynamics may be characterized by various features. For example, instead of measuring the duration of each keystroke (i.e., the time between pressing and releasing each key), one may measure the time between consecutive keystrokes (dwell times). In fact, we performed similar experiments in case of such data as well, and our observations are in accordance with the results reported in Sect. 4.3 for keystroke duration data. Furthermore, it has to be pointed out that increasing usage of smartphones and tablets underline the importance of the protection of personal information. We argue that a combination of various techniques may be necessary: for example, if the user forbids an app to read unique identifiers of the device, the app may still try to identify the user based on heuristics, such as the keystroke dynamics, on which we focused in this paper. While a detailed study of the realization of privacy-aware keyboards are out of scope of this paper, we note that most devices with Android and iOS systems have touch screens which measure pressure as well [1]. Therefore, on such systems, one should pay attention to add noise to the pressure information as well. Furthermore, when the privacy-aware keyboard is realized in a software, it is necessary that it captures keyboard events (or touch screen events, respectively) *before* any other application of the system. As this may be difficult to ensure, we believe that in cases of laptops and desktop computers, it may be more safe to realize a privacy-aware keyboard within the hardware, i.e., in the actual keyboard of the computer.

## References

1. Antal, M., Szabó, L.Z., László, I.: Keystroke dynamics on android platform. Procedia Technol. **19**, 820–826 (2015)
2. Monrose, F., Rubin, A.D.: Keystroke dynamics as a biometric for authentication. Future Gener. Comput. Syst. **16**(4), 351–359 (2000)
3. Doroz, R., Porwik, P., Safaverdi, H.: The new multilayer ensemble classifier for verifying users based on keystroke dynamics. In: Núñez, M., Nguyen, N.T., Camacho, D., Trawiński, B. (eds.) ICCCI 2015. LNCS (LNAI), vol. 9330, pp. 598–605. Springer, Cham (2015). doi:10.1007/978-3-319-24306-1_58
4. Buza, K., Neubrandt, D.: How you type is who you are. In: 11th IEEE International Symposium on Applied Computational Intelligence and Informatics, pp. 453–456 (2016)
5. Kozierkiewicz-Hetmanska, A., Marciniak, A., Pietranik, M.: Data evolution method in the procedure of user authentication using keystroke dynamics. In: Nguyen, N.-T., Manolopoulos, Y., Iliadis, L., Trawiński, B. (eds.) ICCCI 2016. LNCS (LNAI), vol. 9875, pp. 379–387. Springer, Cham (2016). doi:10.1007/978-3-319-45243-2_35
6. Korolova, A., Kenthapadi, K., Mishra, N., Ntoulas, A.: Releasing search queries and clicks privately. In: Proceedings of the 18th International Conference on World Wide Web, pp. 171–180 (2009)
7. Wong, F., Supian, A.S.M., Ismail, A.F., Kin, L.W., Soon, O.C.: Enhanced user authentication through typing biometrics with artificial neural networks and k-nearest neighbor algorithm. In: 35th IEEE Asilomar Conference on Signals, Systems and Computers, vol. 2, pp. 911–915 (2001)
8. Nanopoulos, A., Alcock, R., Manolopoulos, Y.: Feature-based classification of time-series data. Int. J. Comput. Res. **10**(3), 49–61 (2001)
9. Kim, S., Smyth, P., Luther, S.: Modeling waveform shapes with random effects segmental hidden Markov models. In: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, pp. 309–316 (2004)
10. Wozniak, M., Jackowski, K.: Fusers based on classifier response and discriminant function – comparative study. In: Corchado, E., Abraham, A., Pedrycz, W. (eds.) HAIS 2008. LNCS (LNAI), vol. 5271, pp. 361–368. Springer, Heidelberg (2008). doi:10.1007/978-3-540-87656-4_45
11. Krawczyk, B., Minku, L.L., Gama, J., Stefanowski, J., Woźniak, M.: Ensemble learning for data stream analysis: a survey. Inf. Fusion **37**, 132–156 (2017)
12. Buza, K., Nanopoulos, A., Horváth, T., Schmidt-Thieme, L.: GRAMOFON: general model-selection framework based on networks. Neurocomputing **75**(1), 163–170 (2012)
13. Buza, K.: Fusion Methods for Time-Series Classification. Peter Lang Verlag (2011)
14. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Prog. Artif. Intell. **5**(4), 221–232 (2016)

15. Saez, J.A., Krawczyk, B., Wozniak, M.: Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. Pattern Recogn. **57**, 164–178 (2016)
16. Ramírez-Gallego, S., Krawczyk, B., García, S., Woźniak, M., Herrera, F.: A survey on data preprocessing for data stream mining: current status and future directions. Neurocomputing **239**, 39–57 (2017)
17. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. In: Proceedings of the 23rd ACM International Conference on Machine Learning, pp. 1033–1040 (2006)
18. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. Proc. VLDB Endowment **1**(2), 1542–1552 (2008)
19. Chen, G.H., Nikolov, S., Shah, D.: A latent source model for nonparametric time series classification. Adv. Neural Inf. Proc. Syst. **26**, 1088–1096 (2013)
20. Devroye, L., Györfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer, New York (1996)

# Saliency-Based Optimization for the Histogram of Oriented Gradients-Based Detection Methods

Grzegorz Kurzejamski(✉) and Marcin Iwanowski

Institute of Control and Industrial Electronics, Warsaw University of Technology,
ul. Koszykowa 75, 00-662 Warsaw, Poland
{grzegorz.kurzejamski,marcin.iwanowski}@ee.pw.edu.pl

**Abstract.** The paper presents research on using graphical saliency cue for optimizing computation of the distance metrics for HoG features. Saliency values have been computed in the area of a HoG descriptor for macro and micro scale. Macro scale uses HoG features as a global descriptor for an image presenting a particular object, whereas micro scale consists of feature points and k-nearest neighbours approach to create similarity measure. Mechanism has been tested on a chosen database consisting of 20 000 images. Promising results have been achieved for macro scale approach.

**Keywords:** Saliency · HoG · SIFT · Feature points · Object detection · *k*-NN

## 1 Introduction

The local descriptor approach became one of the hottest topics in image processing within recent years. There have been proposed numerous methods that allow for producing the local descriptor vectors for detected feature points [2,6,15–17]. The output of such methods, for a given image, consists of a large number of descriptor vectors each of which consists of dozens of elements. The processing of such collections of vectors for multiple images became computationally complex time-consuming process. In addition, in most of the cases the descriptor vectors are computed not only for the meaningful image regions but also for other regions, less important. Consequently, the existence of additional vectors may lead to some errors in the proper descriptor matching when comparing (objects on) various images. In order to solve above problems, the selection or weighting or either feature points or elements of the descriptor is performed. In this paper we investigate the application of image saliency to achieve this goal for a descriptor based on the histogram of oriented gradients (HoG) [6]. Saliency has been a vastly researched topic. The salient regions in the image are of the utmost importance in computer vision, where we're looking for a most representative parts of visual data. Saliency can be characterized as the visual uniqueness. Visual uniqueness can be described in an organic or non-organic

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_16

manner. Organic approach uses human's visual system as a reference for creating a saliency map. One may use an eye tracking systems for this purpose. There comes many problems with changes to the human's perception in different time spot of the image analysis and learned, knowledge-based biases. Non-organic approach uses prior information for selection of the salient features. The prior may consist of a specific feature points algorithm or a selection of the images with common object present with proper annotations (for further learning). In both cases, saliency would present the cue for visual importance estimation. In theory salient regions should describe more class variation than non-salient regions, so It would be ideal for a feature selection. In practise saliency detectors are prone to variations in image quality and can give extremely different results with a change of background or scale. The reason for that is an assumption, that saliency uses the computed differences between the particular region and its surroundings. In a macro scale it means that saliency of an object is dependant on its ability to differentiate in a particular feature space from the background. Because of the variety in a background scenes and a visuals of a random objects, choice of the salient features is a hard task. In this paper we present research on using the Spectral Residual [11] approach for calculating a HoG descriptor for a restricted scale and rotation wise images presenting random object. The saliency values are used for filtering out unimportant parts of the image that may generate random noise in the distance calculation. Filtering is based on the HoG grid partitioning of the descriptor patch and may lead to smaller processing times without loss of the accuracy. The algorithm has been tested in different scale scenarios. The paper consists of 7 sections. Section 2 presents related works. In Sect. 3, the concept of an application of spectral residual approach to saliency estimation is described. Section 4 contains the description of the proposed idea of using saliency for HoG descriptors. In Sect. 5, the setup for experiments is presented, and in Sect. 6 – their results. Finally, Sect. 7 concludes the paper.

## 2 Related Work

Kadir *et al.* presented interesting approach to feature selection using spatial-scale information in [14]. The main author elaborated on the work in [24]. In [13] authors use graph-based activation maps and normalization for creation of the saliency map. In 2007 Hou *et al.* described saliency operator based on the analysis of the spectrum of natural images, giving promising results. Radhakrishna *et al.* used Difference of Gaussians for selecting the most visually stable areas in the image [19]. In later years there were many works concentrating on the fast computing of the saliency map, using differences in simple features as shape, edges and color [8,18,23]. They often used a segmentation for better salient object retrieval [1,20]. Erdem tried low-level saliency estimator using region covariances [7]. Toet made a comparison work for popular saliency detectors in [25]. Wang *et al.* presented saliency as an anomaly detector in [26]. Optimizing technique has been used in work of Jie Li *et al.* [12]. Cheng *et al.* use information in the group of images presenting the same object for background removal, that is one of the variations of the saliency estimation [3,5]. In further works the same author improves the GrabCut algorithm [4]. Some of the saliency algorithms have been built to be used for a video streams as in [22] or [27]. Rudoy *et al.* uses training data for saliency detector learning process in [21]. There are not many works using saliency for optimizing local or global descriptors using saliency. From the few one can cite the work of Gaurav *et al.* [9], where the saliency measure is incorporated in the classification process.

## 3 Saliency Based on the Spectral Residual Approach

Hou an Zhang [11] proposed a saliency detector based on a spectral residual model. They claim that natural images are not random and they obey highly predictable distributions. That distribution describes the prior knowledge about the scene that should be suppressed by a saliency detector. They show that all natural images have a common spectrum distribution obeying the rule:

$$E\{A(f)\} \propto 1/f, \tag{1}$$

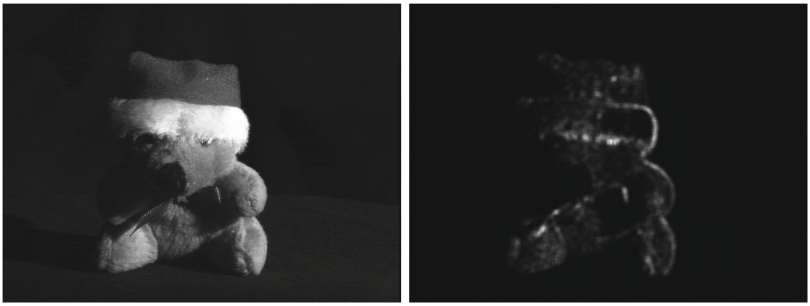
where the *A(f)* is the averaged amplitude of Fourier spectrum calculated on an ensemble of natural images. Log spectrum can be obtained by:

$$L(f) = \log(A(f)) \tag{2}$$

Salient area is then described as a non-linearity in the log spectrum in comparison to the model spectrum obtained from many natural images. These areas are suspected to be a proto-objects for a human's vision system. Residual approach is defined as:

$$R(f) = L(f) - h_n(f) * L(f), \tag{3}$$

where $h_n(f)$ is the averaging $\mathrm{n} \times \mathrm{n}$ matrix. Inverting the Fourier transform with phase data and spectral residual gives us a saliency map. Thanks to the straightforward and simple algorithm, Spectral Residual can be computed very fast. It is prone to optical interferences presented in the image. This characteristic is expected for a saliency detector, because most of the times the object in the scene has higher sharpness levels than the background. Saliency maps created from this algorithm are highly scale dependant, what may mimic the ability to notice the proto objects in the scene of different scales. If one can provide scale independence for the system, as through proper framing system, the saliency should be stable in a macro scale (full object's scale). Macro scale usually means resizing the scene to small resolution before the computation. That action boosts the processing greatly (Fig. 1).



(a) (b)

**Fig. 1.** Saliency computed for sample image: (a) Scene image, (b) Saliency in full resolution.

## 4 Saliency Optimized HoG

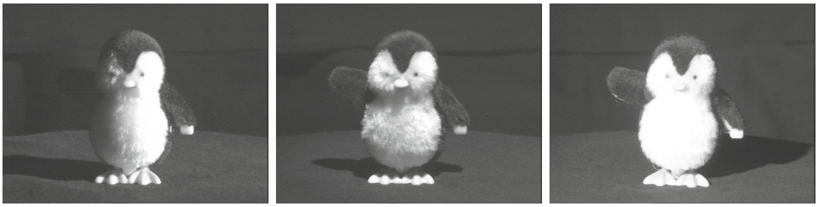
Histogram of Oriented Gradients presented by Dalal and Triggs in [6] is widely used descriptor. It uses a histograms of gradients normalized in blocks uniformly spread in the descriptor's area. Its rich description of the shape and gradients is robust to high luminance changes and small appearance shifts. Similar idea has been successfully used in SIFT descriptor [16] few years earlier. HOG descriptor consists of a vector of values of a fixed size. The size depends on the number of histogram bins in a cell and blocks number in a descriptor window. Histogram is build from pixels inside the cells. Cells are grouped in blocks, where the normalization of the histogram occurs. Blocks can overlap. Each block can be considered as an independent part of the feature vector. This characteristic of the HoG vector can be used for optimizing approaches, as boosting or bagging. Feature vector matching is often performed using the L2 distance. Optimizing single L2 matching computation can be achieved by lowering the dimensionality of the feature vectors. This step usually means computing a prior for a population of features, as space transformation in Principal Component Analysis algorithm. It can be also achieved through building the approximation data structure, as a decision *kd*-tree or hashing tables. The inventive part of the current work consist of using the saliency data to filter out some of the computations needed in metric computation in HoG feature space. We use a saliency computed in the scale of the HoG descriptor window. The saliency values for each pixel are summed in the blocks or cells, creating single value cue for each independent element of the HoG vector. Additional threshold parameter *Thr* is used to choose only the most salient elements of the vector as candidates. The L2 computation is done only with use of elements of the two vectors being the candidates. *Thr* is computed automatically for each descriptor from the saliency map as percentage of the average saliency sum in a block. As the gain in processing speed of the L2 metric is obvious, the overhead coming from saliency computation and matching quality has been researched. Results of there research are presented further in this paper. Saliency is not locally stable. It means, that if one removes an important visual feature from the object's appearance, the whole saliency map may change. Unfortunately the same works when background changes. For matching process though we assume, that the descriptor window contains only described object or the background is the minority of the scene. Owing to that, the usage of binary operator and threshold for accumulated saliency values stabilizes whole process. Since the binary vector of the thresholded saliency values per block has no enough discriminative power to be used as a feature, we use it only as the optimizing step. In the proposed approach, the matching process is built on the assumption, that if the objects match, the most salient regions will be placed in the same cells/blocks. Another assumption made is that the salient cells contain more contrast variations than non-salient regions. It is important to notice that the latter may not be true for some image categories, as the saliency algorithm computed in one scale favour proto-objects of a particular size, discriminating contrast variations in some frequencies. Using cells or blocks for thresholding depends on the context of HoG usage. In standard HoG a feature vector can be partitioned by blocks.

## 5 Experimental Setup

The experiment has been performed for two different approaches performed in macro and micro scale. These are described in the appropriate subsections below. Experiments in both cases have been performed on the same image database consisting of twenty thousand images.

### 5.1 Image Database

The image database used in tests is the ALOI [10] color image collection of one thousand small objects. Objects has been photographed on a black background from different framing angles, different illumination conditions and different color modifications. The database contains also wide-baseline stereo images. Whole database consist of more than one hundred thousands images but only the selected part has been used for the experiment. We use standard HOG descriptor, so we do not consider color data. We want to use reliable metric for results presentation so we do not use images with different viewing angle. Stereo images has been dropped as well. For testing we left one thousand classes containing twenty gray images per class. Each image presents the object with its default position but with different illumination direction. There is only one light source used during the shots so changes in the appearance of the same objects may differ substantially. This changes are presented in Fig. 2. Illumination direction changes are the most erroneous transformations of the object's appearance right beside viewing angle shift. The framing in the database is not ideal, so there are random shifts in the exact position of the object, that generates small error for the descriptors.



(a) (b) (c)

**Fig. 2.** Example of the change in the object's appearance in different illumination angle: (a) Scene image – 1st illumination pattern, (b) Scene image – 2nd illumination pattern, (c) Scene image – 3rd illumination pattern.

### 5.2 Macro Scale

For a macro scale we use HOG as a global descriptor spreading over whole object. At first, we perform a new framing for all of the images to minimize the background share in the scene. The region of interests covers about 50% of the center of the image and is constant for every class. It may results sometimes in a side effect – some parts of the objects are cut-out as well. Without framing the test is however ill conditioned, as we should not assume uniform background (ALOI database contain images which have more than 50% of uniform background). The saliency would then turn background blocks down and create additional shape-related information cue that would have no place in the real world scenario. HOG features have been calculated with appropriate saliency map for each image in the database. The saliency has been computed at the scale of the image. HOG features has been chosen to achieve the best result before using saliency information. We perform a 20-nearest neighbour brute-force search in the feature space for each descriptor. Number of neighbours has been chosen to match number of the images in a class. Then we calculate single accuracy measure from the correct matches in the query. The HoG parameters used for tests in a macro scale are the following:

- Blocks size: $32 \times 32$ px
- Cell grid size: $8 \times 8$ px
- Blocks overlapping: 1 cell $\times$ 1 cell
- Histogram size: 9

Each $k$-NN query consist of L2 metric computation between the feature vectors. Each computation has been performed according to the following algorithm:

```
Begin:
 Input: two feature vectors: v1 v2
        two saliency vectors: s1 s2
        block size: bs
        thresholds: thr1 thr2
        block count divisor: D
```

```
Begin
    sum = 0
    counter = 0
    for each (block index i):
        if (s1[i]>thr1 && s2[i]>thr2)
            sum += compute L2(v1[i*bs:i+1*bs],v2[i*bs:i+1*bs])
            counter++
    if (counter > blocks count / D)
        return sum/counter
    else
        return L2(v1, v2)
End.
```

The threshold can be calculated automatically for each L2 computation. In our tests we use the value: 25% of the average saliency sum in block. This parameter should be chosen to activate about half of the meaningful blocks (containing the object, excluding the background). $D$ parameter represents the factor how many L2 computations will be optimized with saliency filtering. $D$ has been chosen to maximize accuracy and has been set to 5. This algorithm can be optimized further during implementations with dynamic programming. Described procedure is performed with saliency cue and without it (performing standard L2 computation for whole set of feature vectors).

### 5.3 Micro Scale

For micro scale approach we use local features as a bag of words type descriptor. For tests we used implementation of the SIFT keypoints detector. We detected keypoints on each of the image in the database and computed HOG features on the area around each keypoint. We created the descriptor similar to original SIFT descriptor, keeping its appropriate scale, rotation and using blocks consisting of only one cell (and thus no overlapping). In fact, the difference lays only in the normalization procedure. HoG normalizes each block, whereas SIFT uses global descriptor normalization. For each patch on which the descriptor is calculated we computed a single level saliency map and its sums in each cell. The HOG parameters used for tests in micro scale are identical to SIFT descriptor and equal to:

- Patch (interpolated) size: $16 \times 16$ px
- Cell grid size: $4 \times 4$ px
- Blocks overlapping: no overlapping
- Histogram size: 9

Because objects have been presented in the same position, we can create a ground truth for keypoints matches between images in the same class. We perform full 1-nearest-neighbour search for all pairs of images in scope of each class for all of the keypoints. Resulting matches can be checked against the ground truth. The average percent of the correct matches is used as a metric for the experiment. The procedure of L2 optimization using a saliency sum over the blocks is the same as in macro scale approach. The procedure is performed both with and without saliency cue (performing standard L2 computation for whole feature vector).

## 6 Results

Results in both scenarios gave similar average accuracy values. In macro scale in Table 1 the accuracy percent points are given. Accuracy is calculated as number of all correct matches divided by number of matching tasks (times 100%). In micro scale in Table 2 only the average accuracy ratio per object class is given. The reason for that is high variance in base accuracy values for different object classes ranging from 4% to 86%. Average accuracy ratio is calculated with reference to average accuracy gained for simple 'HoG' approach. Accuracy here is calculated per image pair, then averaged. For each image pair in a class we have accuracy computed as a number of correct keypoint matches divided by number of possible correct matches (ground truth). Saliency augmented processing gave better results in macro scale by a 1.5% points. This gain can be substantial in some systems. Nevertheless stability of the accuracy ratio has not been tested on differently conditioned image databases. The difference in matching time is significant as is for descriptor building time. In Content Based Image Retrieval systems this approach can improve overall processing times. The fact that the accuracy gain occurred is an evidence that some non-significant areas of the HoG has low description capabilities for different object's areas. This assumption was a basis for the algorithm. Every parameter configuration tested yield no improvement in average accuracy per object class in a micro scale scenario. The best configuration led to decrement of accuracy by one percent in comparison to no-saliency approach. As we prioritized accuracy for this test, the matching time has been almost identical (difference $\leq$ 0.1%). Configurations that led to remarkable improvement in matching time decreased accuracy ratio

**Table 1.** Results for macro scale.

| | HOG+Saliency | HOG |
|---|---|---|
| Accuracy | 80.56% | 79.11% |
| Matching time ratio | 0.74 | 1.0 |
| Descriptor building time ratio | 2.13 | 1.0 |

**Table 2.** Results for micro scale.

| | HOG+Saliency | HOG |
|---|---|---|
| Average accuracy ratio | 0.99 | 1.0 |
| Matching time ratio | 1.0 | 1.0 |
| Descriptor building time ratio | 1.23 | 1.0 |

greatly. Descriptor building time increased by a 23%. Results have been achieved with manual parameters tweaking. As these has not been optimized fully, it is important to stress, that bad parameters choice could significantly lower the accuracy without any significant gain in processing time. It is thus important to take into consideration the cost of parametrization of new system configuration, as saliency adds complexity. The results show that saliency could be used in Content Based Image Retrieval and any other system that uses most processing time for matching tasks. The macro approach works well for images with excessive non-significant graphical data. Visual variance and significance measures are hard to define, when the whole variance of the signal drops, therefore micro scale approach is highly unstable and do not come with gains worth implementing.

## 7 Conclusions

It the paper, the application of Spectral Residual to optimization of the matching process based on a coherent patch-based descriptor has been presented. A prototype of the algorithm has been build. Implementation has been tested on two scenarios, differentiating in scale of the patches. Test has been performed on twenty thousands images presenting thousand object classes. Results showed that approach may optimize matching process in some application but its parametrization is demanding.

## References

1. Achanta, R., Süsstrunk, S.: Saliency detection using maximum symmetric surround. In: Proceedings of the International Conference on Image Processing, ICIP, pp. 2653–2656. Institute of Electrical and Electronics Engineers (IEEE), September 2010. http://dx.doi.org/10.1109/ICIP.2010.5652636
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Comput. Vis. Image Underst. **110**(3), 346–359 (2008). Similarity Matching in Computer Vision and Multimedia
3. Cheng, M.M., Mitra, N.J., Huang, X., Hu, S.M.: Salientshape: group saliency in image collections. Vis. Comput. **30**(4), 443–453 (2014). http://dx.doi.org/10.1007/s00371-013-0867-4
4. Cheng, M.M., Mitra, N.J., Huang, X., Torr, P.H.S., Hu, S.M.: Global contrast based salient region detection. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 569–582 (2015). http://dx.doi.org/10.1109/TPAMI.2014.2345401
5. Cheng, M.M., Zhang, Z., Lin, W.Y., Torr, P.H.S.: Bing: binarized normed gradients for objectness estimation at 300fps. In: IEEE CVPR. Institute of Electrical & Electronics Engineers (IEEE), June 2014. http://dx.doi.org/10.1109/CVPR.2014.414
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. http://dx.doi.org/10.1109/CVPR.2005.177
7. Erdem, E., Erdem, A.: Visual saliency estimation by nonlinearly integrating features using region covariances. J. Vis. **13**(4), 11 (2013). http://dx.doi.org/10.1167/13.4.11

8. Perazzi, F.: Saliency filters: contrast based filtering for salient region detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012. http://ieeexplore.ieee.org/servlet/opac?punumber=6235193
9. Gaurav, S., Frédéric, J., Cordelia, S.: Discriminative spatial saliency for image classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3506–3513 (2012). http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6248093
10. Geusebroek, J.M., Burghouts, G.J., Smeulders, A.W.: The Amsterdam library of object images, January 2005. http://dx.doi.org/10.1023/B:VISI.0000042993.50813.60
11. Hou, X., Zhang, L.: Saliency detection: a spectral residual approach. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. Institute of Electrical and Electronics Engineers (IEEE), June 2007. http://dx.doi.org/10.1109/CVPR.2007.383267
12. Li, J., Tian, Y., Duan, L., Huang, T.: Estimating visual saliency through single image optimization. IEEE Signal Process. Lett. **20**(9), 845–848 (2013). http://dx.doi.org/10.1109/LSP.2013.2268868
13. Harel, J.: Graph-based visual saliency. In: Neural Information Processing Systems (NIPS), vol. 2006, pp. 545–552 (2006). https://papers.nips.cc/paper/3095-graph-based-visual-saliency.pdf
14. Kadir, T., Brady, M.: Saliency, scale and image description. Int. J. Comput. Vis. **45**(2), 83–105 (2001). http://dx.doi.org/10.1023/A:1012460413855
15. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
16. Lowe, D.: Object recognition from local scale-invariant features (1999). http://dx.doi.org/10.1109/ICCV.1999.790410
17. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 128–142. Springer, Heidelberg (2002). doi:10.1007/3-540-47969-4_9
18. Ming, M.C., Guo, X.Z., Niloy, J.M., Xiaolei, H., Shi, M.H.: Global contrast based salient region detection. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 409–416 (2011). http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5995344
19. Radhakrishna, A., Sheila, H., Francisco, E., Sabine, S.: Frequency-tuned salient region detection. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 1597–1604 (2009). http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5206596
20. Valenti, R.: Image saliency by isocentric curvedness and color. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 2185–2192 (2009). http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5459240
21. Rudoy, D., Goldman, D.B., Shechtman, E., Zelnik-Manor, L.: Learning video saliency from human gaze using candidate selection. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1147–1154. Institute of Electrical and Electronics Engineers (IEEE), June 2013. http://dx.doi.org/10.1109/CVPR.2013.152
22. Seo, H.J., Milanfar, P.: Static and space-time visual saliency detection by self-resemblance. J. Vis. **9**(12), 1–27 (2009). http://dx.doi.org/10.1167/9.12.15

23. Shi, K., Wang, K., Lu, J., Lin, L.: Pisa: Pixelwise image saliency by aggregating complementary appearance contrast measures with spatial priors. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2115–2122. Institute of Electrical and Electronics Engineers (IEEE), June 2013. http://dx.doi.org/10.1109/CVPR.2013.275
24. Kadir, T.: Scale saliency: a novel approach to salient feature and scale selection. In: International Conference on Visual Information Engineering, VIE 2003 (2003)
25. Toet, A.: Computational versus psychophysical bottom-up image saliency: a comparative evaluation study. IEEE Trans. Pattern Anal. Mach. Intell. **33**(11), 2131–2146 (2011). http://dx.doi.org/10.1109/TPAMI.2011.53
26. Wang, M., Konrad, J., Ishwar, P., Jing, K., Rowley, H.: Image saliency: from intrinsic to extrinsic context. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 417–424. Institute of Electrical and Electronics Engineers (IEEE), June 2011. http://dx.doi.org/10.1109/CVPR.2011.5995743
27. Cui, X.: Temporal spectral residual: fast motion saliency detection. In: Proceedings of the 17th ACM International Conference on Multimedia (2009). http://dl.acm.org/citation.cfm?id=1631272

# Efficient Sketch Recognition Based on Shape Features and Multidimensional Indexing

Simone Buoncompagni, Annalisa Franco(✉), and Dario Maio

C.d.L. Ingegneria e Scienze Informatiche, University of Bologna,
Via Sacchi, 3, Cesena (FC), Italy
{simone.buoncompagni2,annalisa.franco,dario.maio}@unibo.it

**Abstract.** Face sketch recognition on real forensic mug shot photo galleries is a complex task since a large amount of images needs to be matched in few seconds to produce a useful outcome. Several effective solutions for sketch-based subject identification have been recently proposed, but the cost of linear search makes them not scalable when large databases have to be scanned. In this work we propose an approach which combines the use of efficient shape features for sketch-photo matching with a suitable indexing structure based on dimensionality reduction. The proposed method provides a preliminary set of candidate photos to be used as input for the final identification based on state-of-the-art techniques, offering scalability and time efficiency without noticeably compromising recognition accuracy, as confirmed by the experimental results.

## 1 Introduction

In forensic contexts face recognition from sketch represents a crucial step to identify culprits. The verbal description provided by one or more eyewitnesses coupled with the skill of a forensic artist could be of paramount importance in order to reproduce a drawing of the potential suspect's face thus guiding the recognition process.

As it occurs for classic face recognition tasks, also for sketch recognition a probe sketch (identikit) has to be matched against a gallery of mug shot photos in order to return a ranking of the most similar subjects. However, sketch and real photos are not directly comparable; the different richness of details and textures requires the adoption of novel and non-standard techniques to allow a direct comparison and overcome such modality gap. Furthermore, typical mug shot galleries like those owned by law and police enforcement in real contexts are composed by thousands of elements and their cardinality is expected to grow continuously as long as new subjects are registered. In other words, a face sketch recognition system is required not only to be accurate but also scalable with reference to the gallery size.

In the last decade, many automatic approaches have been proposed for sketch recognition, but the problem of time efficiency, critical in the presence of large

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_17

mug shot galleries, has been mostly neglected. Only few state-of-the-art methods provide performance results with large galleries (e.g. [4,11,16,22]), and often in these cases additional information (e.g. gender or ethnicity), manually labeled, are exploited to reduce the set of mugshots to analyze.

In this work the shape features proposed in our very preliminary work [5] are organized in an indexing structure in order to reduce the search space when a new sketch (query) has to be matched against a gallery of photos. The most similar photos to a given sketch can thus be retrieved in a time that keeps nearly constant while increasing the gallery size. After shape-based preselection, the recognition task is completed with a robust technique based on SURF descriptors; we thus demonstrate that combining our pre-selection method with state-of-the-art recognition techniques we can tear down the computational cost of recognition obtaining at the same time results comparable to the linear-search scenario, thus offering scalability and time efficiency.

The paper is organized as follows: Sect. 1 provides a review of the state-of-the-art, in Sect. 2 the proposed approach is described, with details about the shape features used and the indexing technique adopted; Sect. 3 reports the results of the experiments carried out to evaluate the algorithm and finally Sect. 4 draws some conclusions.

## 2 Related Works

### 2.1 Face Photo-Sketch Recognition

Face photo-sketch recognition has been widely investigated in the last two decades and two different types of approaches have been introduced with the aim of overcoming the previously mentioned modality gap thus allowing a comparison between a real photo and a sketch. On the one hand, generative approaches [9,20,24,25,27,30] perform a synthesis from photo to sketch (or vice versa) and then make a comparison photo-to-photos or sketch-to-sketches based on traditional face recognition techniques. On the other hand, discriminative approaches (e.g. [3,19,28]) exploit modality independent features, thus making possible a direct matching. Different kinds of descriptors have been adopted in the literature [4,11,15,16,18,29,31].

### 2.2 Face Photo Indexing

Despite great efforts have been made to create indexing techniques related to other biometrics features such as fingerprint [2,6], iris [7] and multibiometric patterns [10], to the best of our knowledge only few indexing approaches have been introduced for face recognition on large galleries. A robust method has been presented in [26] for celebrity face retrieval, where local facial features are employed for a first index-based scan of a web-scale photo gallery and then global features are used to re-rank the previous returned candidate set. Furthermore, hashing-based indexing [12] has been recently proposed for face recognition [13,14] on common available photo datasets, but as far as our knowledge no indexing techniques have been explored for face recognition from sketch where matching heterogeneous image modalities is required. In this work, shape features are combined with an indexing structure based on dimensionality reduction thus resulting in a general method for photo retrieval from sketches on large mug shot galleries.

## 3 Proposed Approach

An overview of the proposed approach for sketch recognition is given in Fig. 1. The input sketch is first processed in order to extract robust face edges, then different shape features are computed to capture general information related to edge orientation, overall distribution of the main facial components and relative displacements of the main facial elements. A feature-level fusion is then adopted to combine the different feature vectors into a single representation which is successively used for photo-candidate pre-selection. Such operation is carried out efficiently thanks to an ad hoc indexing structure, built starting from the gallery of mugshot photos. The output of the index-based search is a list of candidate photos which are finally processed by robust sketch recognition techniques; in this work we exploit Speeded-Up Robust Features (SURF) [1], proposed in the literature as an effective solution for sketch recognition, but in principle the proposed pre-selection approach can be coupled with any recognition technique at the state of the art. In particular, each image (sketch or photo) is partitioned in a set of disjoint patches of fixed size ($32 \times 32$ in our experiments) and for each patch a SURF descriptor, capturing salient local information about the dominant edge orientations within each patch, is computed. The ordered concatenation of the descriptors will provide a global representation of the face.

### 3.1 Shape Feature Extraction

Shape features are computed on an image representing the salient facial edges. To obtain an accurate representation of both global and local facial features a robust technique, based on the Image and Vector Processing Framework (IVPF) [17], is used. The result is a binary image from which the following shape features are computed (please refer to [5] for more details):

- *SM - Shape matrix* [21] computed by superimposing to the face shape a regular matrix, with $M$rows and $N$ columns. For each cell the foreground pixel coverage percentage is computed.
- *BAS - Beam Angle Statistics* [21] describing the shape of the external contour as a statistics of the angle generated by neighboring sampling points uniformly distributed along the shape contour.
- *FD - Fourier Descriptors* [21], computed in the frequency domain, give a description of the face boundary and are extracted from the shape signature representation of the face boundary.

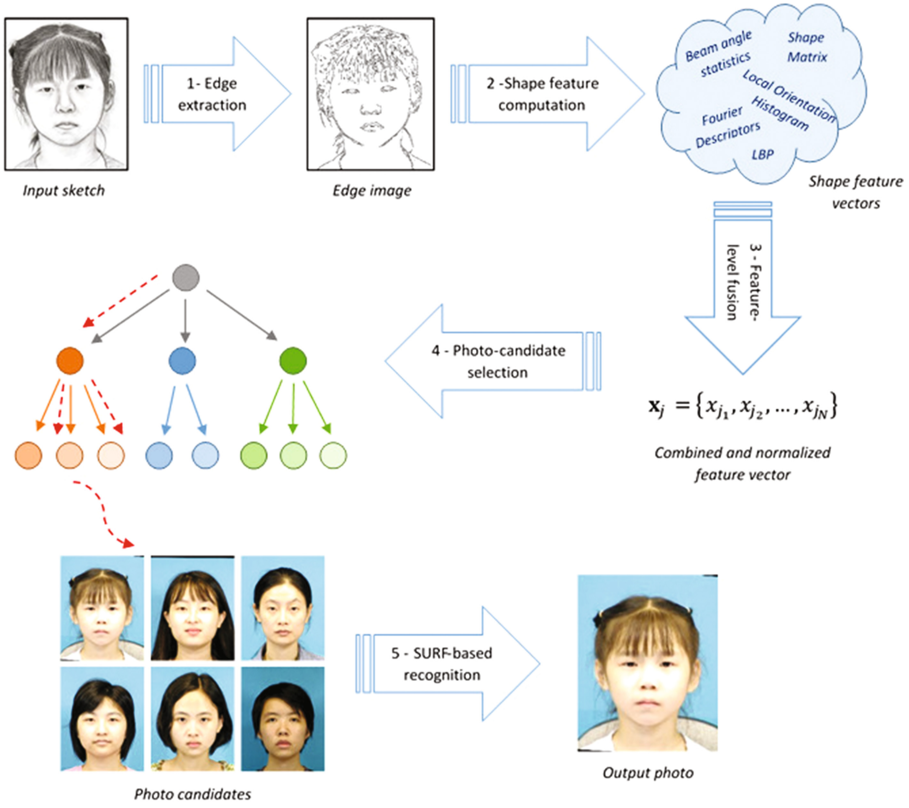

Input sketch
1- Edge extraction
Edge image
2 -Shape feature computation
Beam angle statistics
Shape Matrix
Local Orientation Histogram
Fourier Descriptors
LBP
Shape feature vectors
3 - Feature-level fusion
$\mathbf{x}_j = \{x_{j_1}, x_{j_2}, \ldots, x_{j_N}\}$
Combined and normalized feature vector
4 - Photo-candidate selection
Photo candidates
5 - SURF-based recognition
Output photo

**Fig. 1.** Overview of the proposed sketch recognition approach. The procedure consists of the following steps: (1) Edge extraction from the sketch image; (2) Shape feature computation; (3) Feature normalization and fusion; (4) Index-based photo candidate selection; (5) SURF-based recognition.

- *LBP - Local Binary Pattern*, well-known and effective descriptor originally proposed in [23]. Even though the edge image is not as rich of textures as full gray-scale images, the LPB operator well captures the local distribution of the face edges and increases the discriminating power of the previously mentioned features.

The design of a full indexing structure based on multiple features requires the definition of a strategy for combining them into the indexing scheme. A feature-level fusion approach is here adopted; it is worth noting that the different features are defined in specific, incompatible, ranges so that a normalization is needed to combine them into a single feature vector. In particular the single shape descriptors are first normalized in the range [0,1] according to their maximum theoretical value, then a weight is assigned to the normalized descriptors, depending on the efficacy of each feature for sketch recognition; finally the descriptors are then concatenated in order to create a unique feature vector. Combining multiple features into a single vector produces a high-dimensional pattern that makes indexing a very complex task and requires the adoption of ad-hoc techniques, as described in the following subsection.

## 3.2 MKL-Based Photo Indexing

MKL-tree [8] is a hierarchical and height-balanced structure where data are stored in the leaves, while internal nodes are used to guide the search. All the paths root-leaves have the same length $h$ (height of the tree) thus granting a constant search cost. The structure is built adopting a hierarchical clustering approach, i.e. each element of the internal nodes represents a generalization of its children. In particular, the tree consists of two kinds of node:

- *Internal nodes*: they contain a set of $q$ elements. Each element consists of a KL subspace $S_i$, representing the points in the related sub-tree, and a pointer $p_i$ to the child.
- *Leaves*: the nodes at the lower level of the structure contain a set of r elements. Each element refers to a single indexed pattern $x_j$ and contains a pointer to the disk page where the pattern is stored and the projection of the point $x_j$ into the KL subspace of the father $S_i$.

The MKL-tree is a dynamic structure; new insertions are efficiently managed using ad hoc incremental algorithms appositely studied for KL subspaces; the adoption of such techniques avoids the complete reconstruction of the indexing structure when new records have to be included, and this is a very important requirement in the context considered in this work. The search procedure, which has been specifically designed for face indexing, will be described in detail in the following paragraph; for the other managing techniques please refer to [8].

### 3.2.1 K-NN Searches for Candidate Photo Retrieval

The search procedure designed to retrieve the photo candidates is described in this section. Given the query feature vector (representation of the sketch to identify) and the number *knn* of photo candidates to retrieve, the algorithm visits the tree from the root to the leaves, gradually adding to the result the leaves' elements closest to the query. Starting from the root, the elements of each visited node are analyzed to evaluate the opportunity of following the related path (i.e. to visit its child). A heap queue $Q$ is maintained to store an ordered list of nodes to be visited. The heap initially contains only the root and is successively updated including the nodes fulfilling a probabilistic pruning criterion, adopted to select the most promising paths in the tree. In particular, a bounding box is derived from the subspace representations associated to the internal nodes as the hyper-rectangle centered in its origin and with semi-axes lengths of three times the values of the corresponding standard deviation. A path is followed only if the query fits its bounding box. Each time a leaf node is reached, all the elements whose distance from the query is lower than a threshold are inserted into the result set $R$; the others are stored in a "reserve" list $NQ$, whose elements are maintained in increasing order of distance from the query. A maximum capacity if fixed for the reserve list; if the limit is reached the search procedure is arrested. At the end of the search, if the set of results does not contain the desired number of candidates, the elements of the reserve list closest to the query will be picked to complete the result.

**Algorithm 1.** KNN-Search for candidate photo retrieval

```
input  : $\mathbf{v}$ shape feature vector representing the query sketch
input  : $knn$ number of photo candidates to retrieve
input  : $T$ pointer to the root of the tree structure
output: $R$ candidate photos
$R \leftarrow \emptyset; NQ \leftarrow (t, 0); RQ \leftarrow \emptyset; maxReserveCount \leftarrow knn \times f$;
repeat
    $N \leftarrow$ Extract the minimum element from $NQ$;
    if $N$ is an internal node then
        foreach element $e = (S_i, p_i) \in N$ do
            if $v$ is inside $BoundingBox(S_i)$ then
                $d_i \leftarrow$ Compute the distance of v from space $S_i$;
                $NQ \leftarrow NQ \cup \{(p_i, d_i)\}$;
            end
        end
    end
    else
        foreach element $e = (r_i, p_i) \in N$ do
            Calculate the projection $\mathbf{v'}$ of $\mathbf{v}$ in the subspace associated to $N$;
            $d_i \leftarrow$ Compute the Euclidean distance between $\mathbf{v'}$ and $r_i$;
            if $d_i < d_t$ then
                $R \leftarrow R \cup \{p_i\}$;
            end
            else
                $RQ \leftarrow RQ \cup \{(p_i, d_i)\}$;
            end
        end
    end
until $Q = \emptyset \vee |Q| = maxReserveCount$;
if $|R| < knn$ then
    Add to $R$ the $(knn - |R|)$ elements of $RQ$ with minimum distance from $\mathbf{v}$
end
Return $R$;
```

## 4 Experiments

It is very difficult to find public face datasets realistic in this scenario; we therefore collected well-controlled photos from various sources to build a database of

8221 mug shot images: 188 images from CUHK, 123 from AR face database, 65 from IIITD semi forensic, 114 from CVL, 100 from PUT, 1194 from Feret, 6387 from FRGC and 50 real mug shots collected from the web. The sketches used for testing are those associated to the 311 photos of CUHK + AR (*viewed*), 65 from the IIITD (*semi-forensic*) and 50 from the real web mug shots (*forensic*). The output of the proposed recognition system is a ranking of the photos in the database; the performance is measured in terms of $Rank - N$ recognition accuracy, computed as the percentage of testing sketches correctly identified within the first $N$ positions.

### 4.1 Results

A first set of experiments has been carried out to evaluate the effectiveness of shape features for candidate pre-selection, independently on the indexing scheme adopted. In Table 1 the baseline SURF recognition accuracy measured on the whole database (without pre-selection) is compared to the accuracy obtained by a pre-selection of only 10% of the database photos on the basis of shape features. At this stage the indexing structure is not used. The different nature of the sketches has a severe impact on the recognition accuracy: good performance can be reached with viewed sketches, whereas semi-forensic and forensic images are still very challenging. This behavior can easily be explained considering that viewed sketches provide a faithful representation of the facial appearance whereas semi-forensic and forensic sketches are characterized by significant differences (face geometry, main facial traits) that make difficult even for a human expert to perform a reliable recognition. However, in all cases, the shape-based pre-selection produces a significant improvement in terms of accuracy, noticeably reducing at the same time the recognition time.

**Table 1.** Recognition accuracy of the baseline SURF-based approach *without* (first value) and *with* (second value) shape feature-based preselection. The best result is highlighted in bold.

| Sketch | $Rank - 1$ | $Rank - 5$ | $Rank - 10$ | $Rank - 15$ | $Rank - 20$ |
|---|---|---|---|---|---|
| *Viewed* | 79.4 → **79.7** | 89.7 → **90.0** | 92.6 → **93.2** | **95.2** → 94.2 | **95.2** → **95.2** |
| *Semi-forensic* | 3.0 → **9.0** | 14.0 → **15.0** | 14.0 → **23.0** | 20.0 → **28.0** | 20.0 → **32.0** |
| *Forensic* | 12.0 → **18.0** | 18.0 → **24.0** | 22.0 → **26.0** | 24.0 → **30.0** | 26.0 → **32.0** |

Further experiments have been carried out to evaluate the effectiveness of the indexing structure; starting from the gallery we built to this aim a two-levels structure, and we used the sketches of the CUHK dataset as probes. In particular the Rank-N recognition accuracy values, with $N \in \{1, 5\}$, have been analyzed as a function of the percentage of retrieved candidates (see Fig. 2). The results confirm the efficacy of the indexing structure: a noticeable increment of the efficiency produces only a negligible reduction of the performance and the recognition accuracy measured is very close to the one obtained by the SURF baseline without candidate pre-selection. Moreover, it is clearly visible that even selecting only 5% of the photos as candidate set, the accuracy obtained is close to the maximum value measured over the different experiments, with a large reduction of the computational cost. To analyze the efficiency, the average processing time has been measure for the baseline SURF-based approach as well as for the two shape-based techniques (flat search and indexing). In this evaluation we supposed to pre-compute the features for the mugshot (gallery) images and to save them in the database, as it would be obvious in a real application; therefore, the time reported only includes the following steps: loading the features of the gallery, extracting the features from the test sketch and performing the comparison. The overall recognition time is reported separately for the two stages of the process: shape-based candidate pre-selection and SURF-based recognition. The complexity of the search based on SURF features only is clearly confirmed by the very high recognition time; on the average, about 38 seconds are needed to recognize a sketch. The shape-based candidate pre-selection procedure noticeably reduces the recognition time, in particular when the indexing structure is adopted. The flat search requires a total of about 8 s for recognizing a single sketch (4 s for pre-selection and 4 s for recognition), while the index-based pre-selection needs 4.6 s (0.6 for pre-selection and 4 s for recognition). In this last case the accuracy is slightly lower, but the average recognition time is significantly reduced, moreover, it is worth noting that the search time for the flat shape-based preselection grows linearly with the database size, while for the index-based approach the growth is only logarithmic; this aspect is very important to consider since the database size in this specific scenario is expected to grow continuously, thus making mandatory the adoption of fast indexing techniques.

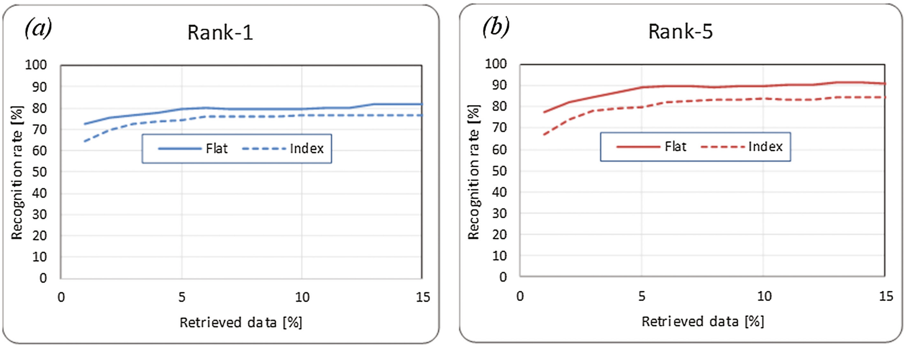


**Fig. 2.** Recognition accuracy obtained using the indexing structure. The $rank - 1$ $(a)$ and $rank-5$ $(b)$ accuracy values are given as a function of the percentage of pre-selected candidates.

## 5 Conclusions

This paper focused on the problem of mug shot photo indexing for sketch recognition. Real scenarios, where constantly increasing galleries of mug shot photos have to be scanned, clearly require efficient solutions to be designed to speed up the search process. A tree-like structure based on shape features has been adopted in this work; the experiments carried out show that a noticeable time reduction can be obtained with a slight performance drop. An analysis of the main causes of errors suggests possible future works. In particular, the preliminary edge extraction step has a major importance in the feature extraction process and often suffers for uneven illumination or other irregularities that might characterize mug shot images; more robust techniques, able to deal with such variations, will be investigated. Moreover, the efficacy of shape feature will be evaluated also for composite sketches which represent a further modality of representation.

## References

1. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (SURF). Comput. Vis. Image Underst. **110**(3), 346–359 (2008). http://www.sciencedirect.com/science/article/pii/S1077314207001555. Similarity Matching in Computer Vision and Multimedia
2. Bhanu, B., Tan, X.: Fingerprint indexing based on novel features of minutiae triplets. IEEE Trans. Pattern Anal. Mach. Intell. **25**(5), 616–622 (2003)
3. Bhatt, H.S., Bharadwaj, S., Singh, R., Vatsa, M.: On matching sketches with digital face images. In: 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), pp. 1–7, September 2010
4. Bhatt, H.S., Bharadwaj, S., Singh, R., Vatsa, M.: Memetically optimized MCWLD for matching sketches with digital face images. IEEE Trans. Inf. Forensics Secur. **7**(5), 1522–1535 (2012)
5. Buoncompagni, S., Franco, A., Maio, D.: Shape features for candidate photo selection in sketch recognition. In: 2014 22nd International Conference on Pattern Recognition, pp. 1728–1733, August 2014
6. Cappelli, R., Ferrara, M., Maltoni, D.: Fingerprint indexing based on minutia cylinder-code. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5), 1051–1057 (2011)
7. Dey, S., Samanta, D.: Iris data indexing method using Gabor energy features. IEEE Trans. Inf. Forensics Secur. **7**(4), 1192–1203 (2012)
8. Franco, A., Lumini, A., Maio, D.: Mkl-tree: an index structure for high-dimensional vector spaces. Multimedia Syst. **12**(6), 533–550 (2007). http://dx.doi.org/10.1007/s00530-006-0070-9
9. Gao, X., Zhong, J., Li, J., Tian, C.: Face sketch synthesis algorithm based on e-hmm and selective ensemble. IEEE Trans. Circ. Syst. Video Technol. **18**(4), 487–496 (2008)
10. Gyaourova, A., Ross, A.: Index codes for multibiometric pattern retrieval. IEEE Trans. Inf. Forensics Secur. **7**(2), 518–529 (2012)
11. Han, H., Klare, B., Bonnen, K., Jain, A.K.: Matching composite sketches to face photos: a component-based approach. IEEE Trans. Inf. Forensics Secur. **8**, 191–204 (2013)

12. He, J., Chang, S.F., Radhakrishnan, R., Bauer, C.: Compact hashing with joint optimization of search accuracy and time. In: CVPR 2011, pp. 753–760 (2011)
13. Kafai, M., Eshghi, K., Bhanu, B.: Discrete cosine transform locality-sensitive hashes for face retrieval. IEEE Trans. Multimedia **16**(4), 1090–1103 (2014)
14. Kaushik, V.D., Umarani, J., Gupta, A.K., Gupta, A.K., Gupta, P.: An efficient indexing scheme for face database using modified geometric hashing. Neurocomputing **116**, 208–221 (2013). http://www.sciencedirect.com/science/article/pii/S0925231212006753
15. Klare, B., Jain, A.K.: Sketch-to-photo matching: a feature-based approach. In: 2010 Biometric Technology for Human Identification, vol. 7667, p. 766702–10 (2010). http://dx.doi.org/10.1117/12.849821
16. Klare, B., Li, Z., Jain, A.K.: Matching forensic sketches to mug shot photos. IEEE Trans. Pattern Anal. Mach. Intell. **33**(3), 639–646 (2011). http://dx.doi.org/10.1109/TPAMI.2010.180
17. Klima, J.: Shape extraction framework for similarity search in image databases. In: Proceedings of the Dateso 2007 Annual International Workshop on DAtabases, TExts, Specifications and Objects, pp. 89–102 (2007)
18. Klum, S., Han, H., Jain, A.K., Klare, B.: Sketch based face recognition: forensic vs. composite sketches. In: 2013 International Conference on Biometrics (ICB), pp. 1–8 (2013)
19. Kukharev, G., Matveev, Y., Forczmański, P.: An approach to improve accuracy of photo–to–sketch matching. In: Campilho, A., Karray, F. (eds.) ICIAR 2016. LNCS, vol. 9730, pp. 385–393. Springer, Cham (2016). doi:10.1007/978-3-319-41501-7_44
20. Li, Y., Savvides, M., Bhagavatula, V.: Illumination tolerant face recognition using a novel face from sketch synthesis approach and advanced correlation filters. In: 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, vol. 2, p. II, May 2006
21. Mingqiang, Y., Kidiyo, K., Joseph, R.: A Survey of Shape Feature Extraction Techniques. INTECH Open Access Publisher (2008). https://books.google.it/books?id=BDDzoAEACAAJ
22. Mittal, P., Jain, A., Goswami, G., Singh, R., Vatsa, M.: Recognizing composite sketches with digital face images via SSD dictionary. In: IEEE International Joint Conference on Biometrics, pp. 1–6, September 2014
23. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 971–987 (2002)
24. Tang, X., Wang, X.: Face sketch recognition. IEEE Trans. Circ. Syst. Video Technol. **14**(1), 50–57 (2004)
25. Wang, X., Tang, X.: Face photo-sketch synthesis and recognition. IEEE Trans. Pattern Anal. Mach. Intell. **31**(11), 1955–1967 (2009)
26. Wu, Z., Ke, Q., Sun, J., Shum, H.Y.: Scalable face image retrieval with identity-based quantization and multireference reranking. IEEE Trans. Pattern Anal. Mach. Intell. **33**(10), 1991–2001 (2011)
27. Xiao, B., Gao, X., Tao, D., Li, X.: A new approach for face recognition by sketches in photos. Sig. Process. **89**(8), 1576–1588 (2009). http://www.sciencedirect.com/science/article/pii/S016516840900067X
28. Yuen, P.C., Man, C.H.: Human face image searching system using sketches. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **37**(4), 493–504 (2007)
29. Zhang, W., Wang, X., Tang, X.: Coupled information-theoretic encoding for face photo-sketch recognition. In: CVPR 2011, pp. 513–520 (2011)

30. Zhang, W., Wang, X., Tang, X.: Lighting and pose robust face sketch synthesis. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6316, pp. 420–433. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15567-3_31
31. Zhang, Y., McCullough, C., Sullins, J.R., Ross, C.R.: Hand-drawn face sketch recognition by humans and a PCA-based algorithm for forensic applications. Trans. Sys. Man Cyber. Part A **40**(3), 475–485 (2010). http://dx.doi.org/10.1109/TSMCA.2010.2041654

# Performance Evaluation of Selected Thermal Imaging-Based Human Face Detectors

Paweł Forczmański(✉)

Faculty of Computer Science and Information Technology,
West Pomeranian University of Technology, Szczecin,
Żołnierska Street 52, 71-210 Szczecin, Poland
pforczmanski@wi.zut.edu.pl

**Abstract.** The paper is devoted to the problem of face detection in thermal imagery. Its aim was to investigate several contemporary general-purpose object detectors known to be accurate when working in visible lighting conditions. Employed classifiers are based on AdaBoost learning method with three types of low-level descriptors, namely Haar–like features, Histogram of Oriented Gradients, and Local Binary Patterns. Additionally, the performance of recently proposed Max-Margin Object-Detection Algorithm joint with HOG feature extractor and Deep Neural Network-based approach have been investigated. Performed experiments, on images taken in controlled and uncontrolled conditions, gathered in our own benchmark database and in a few other databases support final observations and conclusions.

**Keywords:** Thermovision · Biometrics · Face detection · Haar–like features · Histogram of Oriented Gradients · Local Binary Patterns · AdaBoost · Max-Margin Object-Detection Algorithm · Deep Neural Network

## 1 Introduction

It can be observed, that biometric identification, especially methods based on facial portrait recognition, is getting more and more popular. Nowadays, not only top secret assets, but also low-end digital devices are guarded with algorithms capable of face recognition. It is because, human face is one of the most obvious biometric features, easy to capture, distinguish and identify [6]. However, it should be remembered, that there are still many situations when typical biometric approaches are not enough, e.g. when environmental conditions are not fully controlled [7] or there is a need of increased security level. In such case, thermal imaging seems to be a good choice [2], since images registered by infrared or thermal sensors can be used to perform face recognition without the necessity to properly illuminate the subject. Moreover, it is resistant to spoofing attempts (e.g. using a photo or video stream [19]).

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_18

### 1.1 Problem Definition

Locating human face in a static scene is equal to the determination of the image part containing the searched face. In such case it is important to select proper discriminative features used to build a face model. In a typical problem of face detection [20], the false positive rate must be close to $10^{-6}$. It is applicable for objects that are well represented, i.e. with similar imaging conditions, with standardized orientation etc. In case of thermographic imaging, the visual representation of faces depends highly on camera calibration, which may not be assumed. Some variations of facial portraits in terms of different camera calibration parameters are presented in Fig. 1. As it can be seen, different parts of faces are visible with different accuracy, hence it makes the discussed task particularly difficult, and highly dependant on training samples availability. The other important aspects of this problem are the mechanisms for feature matching and scanning of the source image. Hence, if there is no information about probable face position and size, the detection requires to perform search process in all possible locations, taking into consideration all probable window (or image) scales, which increases the overall computational overhead. In the paper we focus on feature extractors and classifiers that enable proper facial portrait detection. The algorithms were selected taking into consideration the computational complexity, the simplicity of implementation and the accuracy. Hence, we have selected several well-know and recently proposed approaches.

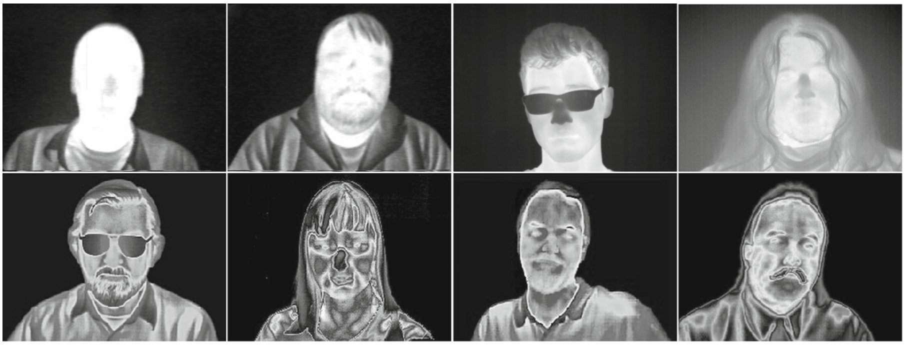


**Fig. 1.** Exemplary images taken with different camera calibration parameters

### 1.2 Employed Methods and Related Works

Despite the fact that the problems of human face detection and recognition in visible light have been investigated many times, the detection and recognition of faces in thermal spectrum are less represented [10]. The absence of standardized methods is caused by specific characteristics of thermal images which change the appearance of human faces depending on the surrounding temperature. In this context, the main disadvantage specific to the thermal sub-band images (thermograms) comes from the fact that the heat pattern emitted by the face is affected by a number of confounding factors, such as ambient temperature, air flow direction, physical exercises, individual metabolism, illness or drugs [18]. The methods of human face detection in both visible and thermal imagery can be classified as appearance-based and model-based [21]. It should be noted, however, that the second one is more popular [4]. Since there is a lack of established benchmarks, it makes the comparison of the algorithms very problematic. Nevertheless, there are some reports involving datasets of similar characteristics. Reported accuracy varies form approximately 67% to 96%, depending on the assumptions and imaging conditions. In this work we focus on certain well-researched approaches, employing some general-purpose features extractors and robust classifiers, namely Histogram of Oriented Gradients [3], Local Binary Patterns [17] and Haar-like features [20] combined with AdaBoost-based classifiers. Some previous investigations on these extractors are presented in [8]. Here, we additionally tested recently proposed Max-Margin Object-Detection Algorithm (MMOD) [14] joint with HOG and Deep Neural Network (DNN) as a combined feature extractor and classifier. The second element of the addressed detection algorithm is a classification. Most object detection methods operate by applying a binary classifier to sub-windows of an image, followed by a non-maximum suppression step where detections on overlapping sub-windows are removed. Since the number of possible sub-windows in even moderately sized image can be extremely large, the classifier is typically learned from only a subset of the windows. For several recent years, it has become a standard to employ an AdaBoost algorithm at the learning stage [1]. The derived feature vectors (HOG, LBP or Haar-like features) are used in the process of classification using e.g. Support Vector Machine or, in our case, AdaBoost [9]. One of the alternatives is Max-Margin Object-Detection Algorithm (MMOD) described in [14]. MMOD can be used to improve any object detection method which is linear in the learned parameters, such as HOG or bag-of-visual-word models. Its major advantage comes from the fact that it does not require to provide negative samples at the learning stage. It is enough to simply label positive samples in images. It also produces high quality detectors from relatively small amounts of training data. On the other hand, like the above mentioned feature detectors, it is not capable of learning to detect objects that exhibit complex pose variation or have a lot of other variability in how they appear. To overcome this limitation, the Convolutional Neural Networks were introduced, which are capable of dealing with all these issues within a single model. This alternative direction in research, associated with a use of deep learning methods, is aimed at creating robust classifiers without a need of carefully crafted low-level features [15]. Hence, in our research we also tested both above approaches in order to confirm their high robustness and accuracy.

## 2 Experiments

### 2.1 Benchmark Datasets

We evaluated the algorithm's performance on four different datasets: WIZUT database [12], Caltech (Courtesy NASA/JPL-Caltech) [11], OTCBVS Terravic Facial IR Database [16] and various uncontrolled images taken from the Internet. Collected faces include full frontal portraits (except WIZUT 'a', 'c', 'd', 'e' and UC), some of them include glasses, head cover, or both. The exemplary cropped faces are presented in Figs. 2 and 3. The details are given in Table 1. It should be noted that in case of OTCBVS Terravic Facial IR Database, from a total number of 23262 images, 12255 images containing faces rotated not more than 20° were selected, from which 2000 were randomly chosen for the evaluation. The images from WIZUT and OTCBVS are normalized in terms of temperature range, while the rest are images with variable temperature ranges. It leads to the very different representation of temperatures in the pixel's intensities. The images forming UC set were taken using various equipment, in variable conditions, indoor/outdoor, hence they are the most problematic data for the detection.

**Table 1.** Benchmark datasets characteristics

| Dataset | WIZUT | Caltech | OTCBVS | UC |
|---|---|---|---|---|
| No. images/faces | 505 | 64 | 2000 | 63/123 |
| No. subjects | 101 | 28 | 20 | 123 |
| Image width | 320 | 285 | 320 | 167–1920 |
| Image height | 240 | 210 | 240 | 129–1215 |
| Min. face size | 119 × 124 | 69 × 73 | 96 × 120 | 22 × 26 |
| Max. face size | 182 × 178 | 121 × 135 | 144 × 212 | 354 × 317 |
| Rotation angle [°] | ±45 | ±5 | ±20 | n/a |

### 2.2 Experimental Setup

To show that the algorithm is unbiased for the training dataset, we selected a training set, which is completely different from the testing set. The first experiment (Experiment No. 1) was devoted to the in-dept analysis of the performance of selected features/classifiers pairs on individual testing subsets. Hence, we selected WIZUT (b) as a training set. This set contains frontal faces only (101 images out of total number of 505 portraits). At that stage, each face has been manually cropped. No other pre-processing has been employed. All the faces used for training are presented in Fig. 2. It resembles a situation, where a limited number of training samples is available. Moreover, only frontal faces (like in identification documents) are provided, hence the detector can not tune to the orientation variations. The second experiment (Experiment No. 2) is devoted to the analysis of the performance of the detector trained on a significantly larger dataset: WIZUT ‘a’–‘e’, Caltech & OTCBVS), tested on UC dataset only. In such case, positive examples are formed from faces with different orientation and imaging conditions. We wanted to check the influence of the higher number and variability of positive samples on the performance on rather challenging testing images. Selected learning images are presented in Fig. 3. In each experiment we tested five approaches: cascading classifiers based on Haar-like features (Haar), Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP), together with HOG learned by Max-Margin Object-Detection Algorithm (HOG+MMOD) and deep-learning Convolutional Neural Networks (DNN). The first three classifiers were implemented using Open Computer Vision library (OpenCV), while the rest two classifiers were implemented using Deep Learning Library – dlib [13]. OpenCV implementation used Intel i7 processor, while dlib programs used NVidia GTX780 GPU. In both experiments, ground-truth bounding boxes were marked manually, covering most informative facial part. The negative samples were provided for cascading classifiers in an automatic manner. They were extracted from thermal images taken from the Internet, containing no faces. In case of cascading classifier, during learning, standard boosted cascade algorithm was applied, namely AdaBoost (Gentle AdaBoost variant), and varying values of the following parameters: number of positive/negative samples and the number of learning stages. The number of positive samples is often larger than the total number of the images in the dataset. It comes from the fact, that the images are artificially modified (rotated, brightened, darkened, noised, cropped) in order to increase their variability. In case of HOG+MMOD and DNN, only the number of positive samples is parametrized. The parameters for Experiment No. 1 are presented in Table 2. In case of Experiment No. 2 three parameters setups were tested - they are presented in Table 3. The number of positive samples varies, since the cascading classifiers uses a limited number of samples at each training stage, while DNN uses all the samples. HOG+MMOD uses positive samples doubled by using symmetrical images. The number of iterations depend on the learning rate and is automatically calculated.

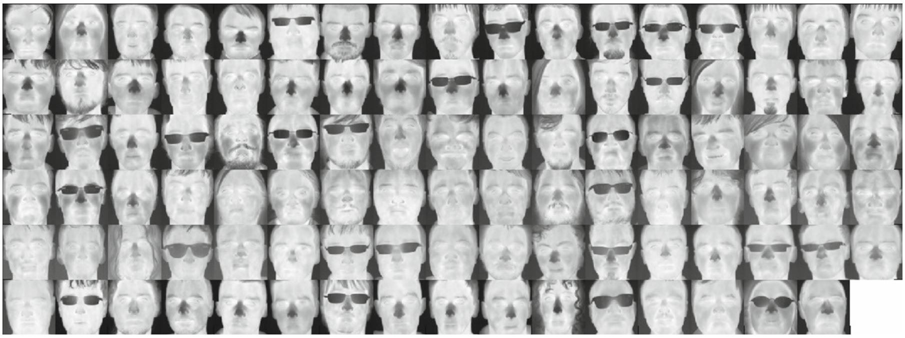


**Fig. 2.** Images used for learning (Experiment No. 1)

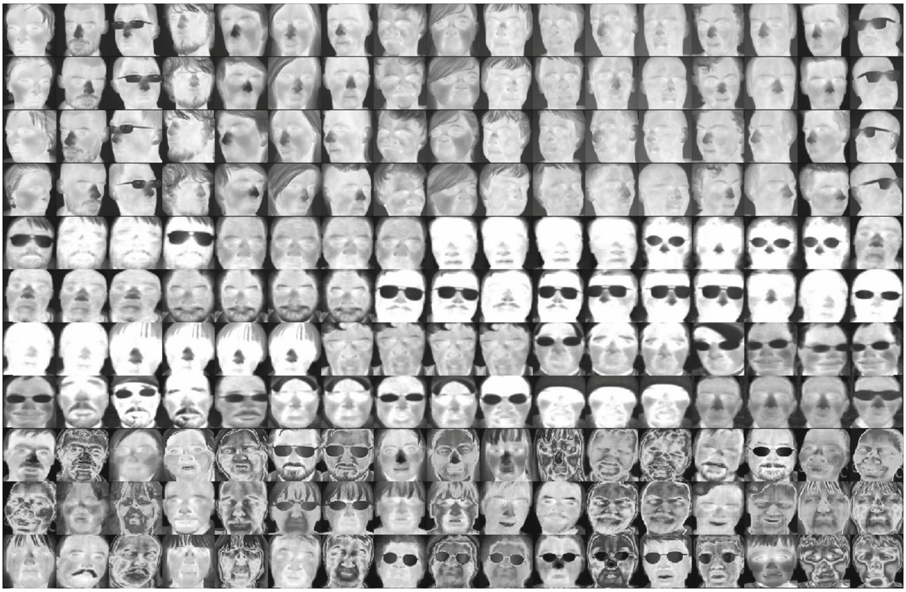


**Fig. 3.** Selected images used for learning (Experiment No. 2)

**Table 2.** Training parameters for Experiment No. 1

| Detector | Cascade | | | HOG+MMOD | DNN |
|---|---|---|---|---|---|
| | Haar | HOG | LBP | | |
| Window size | 32 × 32 | | | 40 × 40 | 40 × 40 |
| Positive samples | 800 | 800 | 800 | 202 | 101 |
| Negative samples | 2000 | 2000 | 3000 | – | – |
| No. stages/iterations | 18 | 18 | 16 | 63 | ~2500 |

**Table 3.** Training parameters for Experiment No. 2

| Detector | Cascade | | | | | | | | | HOG+MMOD | DNN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Haar | | | HOG | | | LBP | | | | |
| Setup no. | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | | |
| Pos. smpl. | 800 | 800 | 1000 | 800 | 1000 | 1500 | 500 | 800 | 1000 | 5138 | 2569 |
| Neg. smpl. | 2000 | 3000 | 5000 | 4000 | 4000 | 4000 | 2500 | 4000 | 2000 | – | – |
| No. stages | 11 | 12 | 12 | 15 | 17 | 17 | 11 | 11 | 13 | 54 | 3500 |

## 2.3 Accuracy Evaluation

In order to evaluate the accuracy of the detector, we calculate an Intersection over Union - $IoU$. It is often used in object detection challenges such as the popular PASCAL VOC challenge [5]. In order to apply Intersection over Union to evaluate an object detector we have to provide bounding boxes associated with the object(s) in the image. The bounding boxes are given as quadruples consisting of $(x, y)$-coordinates of the top-left corner of the box and its width $w$ and height $h$. With each image we associate:

- the ground-truth bounding box(es) of an area $A_{gt}$ representing known object(s),
- the predicted bounding box(es) from the model of an area $A_{det}$ representing detected object(s).

Hence, Intersection of Union is calculated by dividing the area of overlap between the bounding boxes by the area of union:

$$IoU = \frac{A_{gt} \cap A_{det}}{A_{gt} \cup A_{det}}. \tag{1}$$

An $IoU$ score higher than 0.5 is often considered a *good* prediction. However, in our case, we set the threshold to 0.3, in order to increase the recall accuracy. Exemplary values of $IoU$ for various detected faces are presented in Fig. 4. From the practical point of view, in the approach presented here, a matrix of IoUs is calculated for all combinations of ground-truth and detected bounding boxes, and only the highest scores are taken into consideration. As the performance metric, we selected precision and recall measures, taking into consideration $IoU$ values. The precision value is a number in the range 0–1 which measures the fraction of detector outputs which correspond to a real target. A value of 1 means the detector never produces any false alarms while a value of 0 means it only produces false alarms. On the other hand, the recall value is a number in the range 0–1 which measures the fraction of targets found by the detector. A value of 1 means the detector found all the non-ignore targets, while a value of 0 means the detector did not locate any of the targets.

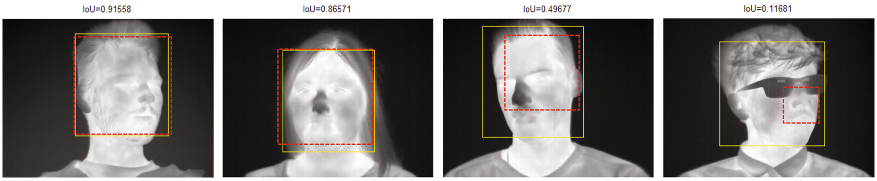


**Fig. 4.** A comparison of various $IoU$ scores: proper detection (two first cases), border line detection (third case) and unsuccessful detection (the last case)

## 2.4 Results

Several experiments have been carried out in order to verify the effectiveness of the selected detectors in the task of face detection in static scenes. The detection involved trained detectors with different parameters, presented in Table 4, namely size of search window, scaling step in the image pyramid and candidates rejection rule based on the number of adjacent detections (number of neighbouring detection indicating true detection, in case of cascading classifiers). The experiments were aimed at selecting the best classifier taking into consideration face detection rate (precision and recall rates) as well as stability of the results. To make the results more objective, we calculated weighted mean values of precision and recall indicators in case of Experiment No. 1 (the weights were associated with the number of testing images). The unweighted means were calculated for Experiment No. 2. As it can be seen from the presented tables (Tables 5 and 6), face detection in static scenes gives varying results, depending on the benchmark dataset. While the detection rate for rather easy images is very high (almost perfect), for images taken “in the wild” drops. When the main priority is safety, the precision should be considered as a main indicator. In such case, we detect faces, that may be successfully recognized at subsequent processing stages. Hence, DNN and HOG+MMOD give the best results. On the other hand, when we want a detector that locates most of the faces, recall is more important. In such case, we may get faces at orientations/variations, that are characterized by less probable recognition. Hence, all detectors based on cascading AdaBoost strategy are the choice. When considering the computational complexity, LBP/AdaBoost combination should be applied, thanks to its high-speed learning and fast classification (thanks to integer number calculations). The slowest, yet the most “safe” detector is build over DNN approach. The exemplary results are presented in Fig. 5. As it can be seen, most of the faces have been detected correctly. It is also interesting that in the case of a mannequin face captured in visible light, all, but Haar detector, actually found a face. There are some false detections, associated mostly with Haar, LBP and HOG detectors.

**Table 4.** Main parameters of evaluated detectors

| Detector | Haar/HOG/LBP | HOG+MMOD | DNN |
|---|---|---|---|
| Min window size | 16 × 16 | 40 × 40 | 40 × 40 |
| Max window size | 360 × 360 | 80 × 80 | 200 × 200 |
| Scaling factor | 1.02 | 1.2 | 1.2 |
| No. neighbors | 2 | – | – |

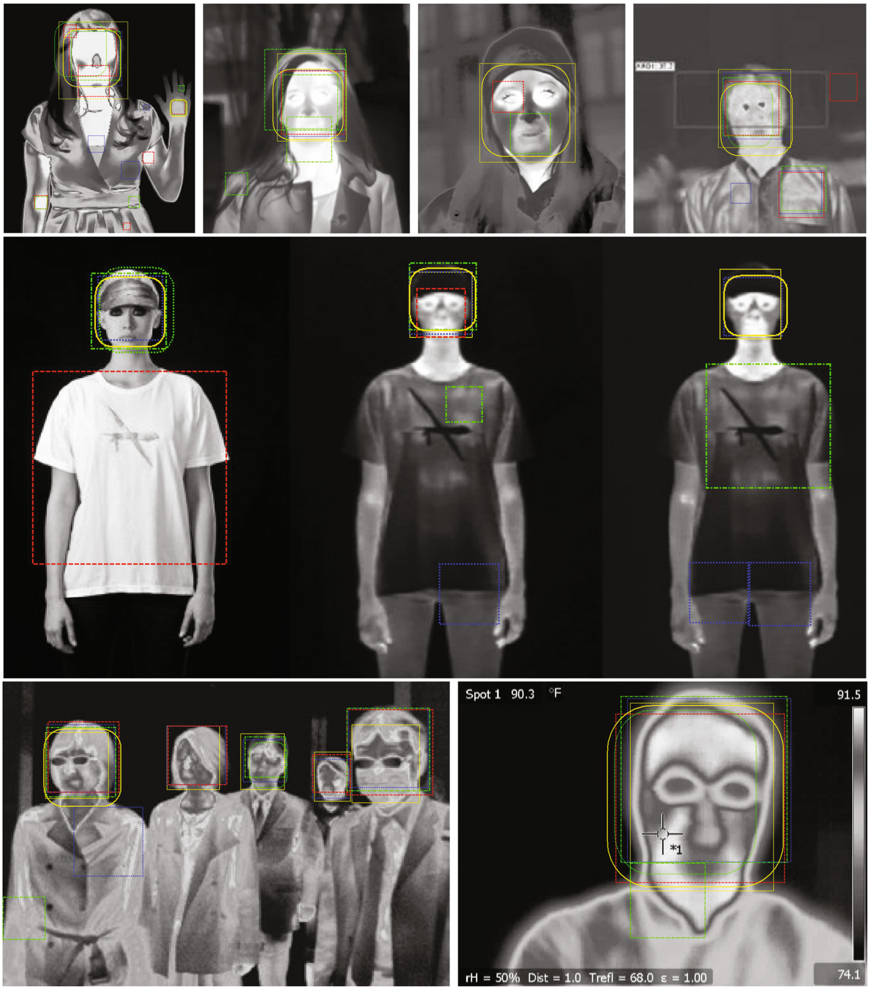


**Fig. 5.** Exemplary results from Experiment No. 2. The yellow solid line squares represent ground-truth, blue dotted line squares – HOG results, red dashed line – Haar results, green dash-dot line – LBP results, yellow solid line, rounded rectangles – HOG+MMOD results, and green rounded rectangles with dotted line – DNN results.

**Table 5.** Recall and precision rates for Experiment No. 1

| Detector | Haar | | HOG | | LBP | | HOG+MMOD | | DNN | |
|---|---|---|---|---|---|---|---|---|---|---|
| Testing set | Prec. | Rec. | Prec. | Rec. | Prec. | Rec. | Prec. | Rec. | Prec. | Rec. |
| WIZUT (a) | 0.59 | 1 | 0.82 | 0.93 | 0.66 | 0.99 | 1 | 0.97 | 1 | 1 |
| WIZUT (b) | 0.8 | 1 | 0.95 | 1 | 0.89 | 1 | 1 | 1 | 1 | 1 |
| WIZUT (c) | 0.74 | 0.99 | 0.89 | 0.97 | 0.72 | 1 | 1 | 1 | 0.99 | 0.99 |
| WIZUT (d) | 0.52 | 0.93 | 0.4 | 0.46 | 0.6 | 0.95 | 0.99 | 0.88 | 1 | 1 |
| WIZUT (e) | 0.64 | 0.84 | 0.52 | 0.32 | 0.62 | 0.98 | 1 | 0.9 | 0.99 | 0.99 |
| Caltech | 0.87 | 0.72 | 0.9 | 0.73 | 0.74 | 0.81 | 0.62 | 0.7 | 0.98 | 0.94 |
| OTCBVS | 0.75 | 0.97 | 0.93 | 0.96 | 0.86 | 0.99 | 1 | 0.91 | 1 | 1 |
| UC | 0.37 | 0.57 | 0.58 | 0.57 | 0.3 | 0.57 | 0.86 | 0.39 | 0.87 | 0.33 |
| Weighted mean | 0.81 | 0.76 | 0.86 | 0.75 | 0.72 | 0.83 | 0.7 | 0.73 | 0.98 | 0.92 |

**Table 6.** Recall and precision rates for Experiment No. 2

| Detect. | Haar | | | HOG | | | LBP | | | HOG+MMOD | | | DNN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Setup | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | – |
| Prec. | 0.47 | 0.5 | 0.4 | 0.69 | 0.62 | 0.58 | 0.46 | 0.44 | 0.5 | 0.9 | 0.85 | 0.85 | 0.88 |
| Rec. | 0.44 | 0.49 | 0.46 | 0.41 | 0.49 | 0.56 | 0.56 | 0.5 | 0.47 | 0.21 | 0.45 | 0.46 | 0.3 |
| $\overline{\text{Prec.}}$ | 0.46 | | | 0.63 | | | 0.47 | | | 0.86 | | | 0.88 |
| $\overline{\text{Rec.}}$ | 0.5 | | | 0.49 | | | 0.51 | | | 0.37 | | | 0.3 |

## 3 Summary

In the paper an evaluation of the selected thermal-imaging based face detectors was presented. The results of the experiments show that all of the presented classifiers can be applied to the presented task. Since the recognition rate is not perfect, some general advices have been given. The presence of a small number of false detections as well as some missed faces cause that the solution should be modified. One of the possible proposals is to perform an iterative detection involving the detectors with the highest recall rate (e.g. LBP-based), preceded by a detector with the highest precision (e.g. HOG, HOG+MMOD, DNN). The result showed also, that the Haar-like features, although very good in locating faces in visible lighting, are not so good in thermal imagery.

## References

1. Burduk, R.: The AdaBoost algorithm with the imprecision determine the weights of the observations. In: Nguyen, N.T., Attachoo, B., Trawiński, B., Somboonviwat, K. (eds.) ACIIDS 2014, Part II. LNCS (LNAI), vol. 8398, pp. 110–116. Springer, Cham (2014). doi:10.1007/978-3-319-05458-2_12
2. Chang, H., Koschan, A., Abidi, M., Kong, S.G., Won, C.-H.: Multispectral visible and infrared imaging for face recognition. In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–6 (2008)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 (2005)
4. Dowdall, J., Pavlidis, I., Bebis, G.: Face detection in the near-ir spectrum. Image Vis. Comput. **21**(7), 565–578 (2001)
5. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL visual object classes challenge: a retrospective. Int. J. Comput. Vis. **111**(1), 98–136 (2015)
6. Forczmański, P., Kukharev, G.: Comparative analysis of simple facial features extractors. J. Real-Time Image Process. **1**(4), 239–255 (2007)
7. Forczmański, P., Kukharev, G., Shchegoleva, N.: Simple and robust facial portraits recognition under variable lighting conditions based on two-dimensional orthogonal transformations. In: Petrosino, A. (ed.) ICIAP 2013. LNCS, vol. 8156, pp. 602–611. Springer, Heidelberg (2013). doi:10.1007/978-3-642-41181-6_61
8. Forczmański, P.: Human face detection in thermal images using an ensemble of cascading classifiers. In: Kobayashi, S., Piegat, A., Pejaś, J., El Fray, I., Kacprzyk, J. (eds.) Hard and Soft Computing for Artificial Intelligence, Multimedia and Security. AISC, vol. 534, pp. 205–215. Springer, Cham (2017). doi:10.1007/978-3-319-48429-7_19
9. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995). doi:10.1007/3-540-59119-2_166
10. Ghiass, R.S., Arandjelovic, O., Bendada, H., Maldague, X.: Infrared face recognition: a literature review. In: International Joint Conference on Neural Networks (cs.CV) (2013). arXiv:1306.1603
11. Hermans-Killam, L.: Cool Cosmos/IPAC website, Infrared Processing and Analysis Center. http://coolcosmos.ipac.caltech.edu/image_galleries/ir_portraits.html. Accessed 10 May 2016
12. Jasiński, P., Forczmański, P.: Combined imaging system for taking facial portraits in visible and thermal spectra. In: Choraś, R.S. (ed.) Image Processing and Communications Challenges 7. AISC, vol. 389, pp. 63–71. Springer, Cham (2016). doi:10.1007/978-3-319-23814-2_8
13. King, D.: Dlib 18.6 released: Make your own object detector! (2015). http://blog.dlib.net/2014/02/dlib-186-released-make-your-own-object.html. Accessed 27 Jan 2017
14. King, D.E.: Max-Margin Object Detection, Computer Vision and Pattern Recognition (cs.CV) (2015). arXiv:1502.00046
15. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25, pp. 1106–1114 (2012)

16. Miezianko, R.: IEEE OTCBVS WS Series Bench – Terravic Research Infrared Database. http://vcipl-okstate.org/pbvs/bench/. Accessed 20 May 2016
17. Ojala, T., Pietikinen, M., Harwood, D.: Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In: Proceedings of the 12th International Conference on Pattern Recognition, vol. 1, pp. 582–585 (1994)
18. Prokoski, F.J., Riedel, R.: Infrared Identification of Faces and Body Parts. In: BIOMETRICS: Personal Identification in Networked Society. Kluwer (1998)
19. Smiatacz, M.: Liveness measurements using optical flow for biometric person authentication. Metrol. Meas. Syst. **19**(2), 257–268 (2012)
20. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vis. **57**(2), 137–154 (2004)
21. Wong, W.K., Hui, J.H., Lama, J.A.K., Bin Md Desa, J., Izzati, N., Ishak, N.B., Bin Sulaiman, A., Nor, Y.B.M.: Face detection in thermal imaging using head curve geometry. In: 5th International Congress on Image and Signal Processing (CISP 2012), pp. 1038–1041 (2012)

# On a New Method of Dynamic Integration of Fuzzy Linear Regression Models

Jakub Kozerski(✉) and Marek Kurzynski

Wroclaw University of Science and Technology,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
jakub.kozerski@pwr.edu.pl

**Abstract.** In the study the problem of ensemble regression with fuzzy linear regression (FLR) models is considered. For this case a novel method of integration is proposed in which first fuzzy responses of base FLR models are integrated and next the fuzzy response of a common model is defuzzified. Four different operators are defined for integration procedure. The performance of proposed integration methods of FLR base models on the soft level were compared against state-of-the-art integration method on the crisp level using computer generated datasets with linear, 2-order and 3-order models and different variances of Gaussian disturbances. As a criterion of method quality the root mean square error was applied. The results of computer experiments clearly show that in many cases proposed methods significant outperform the reference approach.

**Keywords:** Fuzzy regression model · Regression ensemble · Integration method · Fuzzy integration

## 1 Introduction

Regression analysis is a statistical tool for the investigation of the dependence of a response variable on independent variables. Regression analysis is an empirical methodology, which quantitative specification of relationship between variables (called regression model) derives from observed data. Therefore, regression analysis is a central methodology in many research projects, mainly in disciplines where it is the lack of theory explaining, how the response varies as the values of independent variables change [3].

For a long time, the least square method was the only approach to create regression model. Widely available high-speed computing has expanded regression analysis methods with new techniques, such as nonparametric regression, neural networks, support vector machines, tree-based methods, to name only a few [13]. Within the last two decades, also the machine learning community has enclosed innovative ideas to the range of methods applied in the regression analysis. One of them is ensemble regression approach.

Nowadays, ensemble regression methods which combine responses of set of base regression models are intensively developed. The reason is that different

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_19

models offer complementary information about relationship between response and independent variables and therefore ensemble regression can achieved better approximation of this relationship, than any single base model [4,6,8–10].

In the procedure of combining base models for ensemble regression, two steps can be distinguished: ensemble selection or ensemble pruning and ensemble integration. In the first step, the original set is reduced by eliminating some of the base models, which – by a certain criterion – are less useful. The ensemble selection can be either static or dynamic. In the static selection scheme ensemble is selected for all test instances, whereas dynamic ensemble selection approach explores the use of different ensembles of base models for different test instances. In the ensemble integration, a method of aggregating responses of models is defined which is used to generating response of a common model based on the predictions of an ensemble members. There are many techniques of ensemble integration such as BEM and GEM methods [9], linear regression [5], stacked regression [1] or dynamic integration [7].

All abovementioned integration methods, although different in concept and procedures, have one characteristic in common: all methods integrate crisp responses of base regression models. In this study, we consider ensemble regression with fuzzy linear regression (FLR) models. For this case, a novel integration method is developed, in which first fuzzy responses of base FLR models are integrated and then fuzzy common response is defuzzified. Four different operators are proposed for the integration of fuzzy responses of FLR models. In a nutshell, originality of the proposed approach consists in integration of base regression models on the level of soft responses.

The paper is divided into five sections and organized as follows. In Sect. 2 the fuzzy linear model is presented and methods of its training are described. In Sect. 3 the original integration method of FLR base models is developed and different operators used in integration procedures are defined. The performance of proposed integration methods of FLR base models on the soft level were compared against state-of-the-art integration method on the crisp level using computer generated datasets with different models (linear, 2-order and 3-order) and different variances of Gaussian disturbances. The results of computer experiments are presented in Sect. 4 and Sect. 5 concludes the paper.

## 2 Fuzzy Regression Model

### 2.1 Preliminaries

Let

$$y = \Phi(x) + \varepsilon, \tag{1}$$

where $\Phi$ is an unknown function mapping the $N$-dimensional space $\mathcal{X}$ of input (independent, predictor) variable $x = (x^{(1)}, x^{(2)}, \ldots, x^{(N)})$ into one-dimensional space $\mathcal{Y}$ of output (dependent, response) variable $y$ and $\varepsilon$ is a random disturbance term with zero expected value $E(\varepsilon) = 0$ and a finite variance $Var(\varepsilon) < \infty$.

In the context of supervised learning, the parametric regression problem consists in approximation of function (1) by function $f(x, \alpha)$ which depends on a finite set of parameters $\alpha = (\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(p)})$ assuming that the learning set is given:

$$\mathcal{S} = \{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}, \tag{2}$$

where $y_i = \Phi(x_i) + \varepsilon$. Function $f(x, \alpha)$ built on the base of (2) is called regression model or simply regressor or predictor and typically its form is linear, polynomial or log-linear [13].

The aim of this study is to approximate the unknown function $\Phi(x)$ using a set of trained base models $\mathcal{F} = \{f_1, f_2, \ldots, f_L\}$ which must be integrated into one common model $f_{\mathcal{F}}$ using validation set:

$$\mathcal{V} = \{(x_1, y_1), (x_2, y_2), \ldots (x_m, y_m)\}. \tag{3}$$

In the next subsection we describe the training process of base models in a special case, when the regressors are linear and their parameters are fuzzy numbers.

### 2.2 Fuzzy Linear Regression Model

Standard approach for regression is linear modeling:

$$\hat{y} = f(x, \alpha) = \alpha^{(0)} x^{(0)} + \alpha^{(1)} x^{(1)} + \cdots + \alpha^{(j)} x^{(j)} + \cdots + \alpha^{(N)} x^{(N)} = \alpha \bar{x}', \tag{4}$$

where $\hat{y}$ is a model response, $x^{(0)} = 1$, $\bar{x} = (x^{(0)}, x)$ and $'$ denotes the transposition operator.

For the case of crisp (non-fuzzy) independent variables $x$, Tanaka in [12] proposed extension of model (4) to the fuzzy linear regression model (FLR):

$$\hat{Y}(x) = F(A, x) = A^{(0)} x^{(0)} + A^{(1)} x^{(1)} + A^{(2)} x^{(2)} + \cdots + A^{(N)} x^{(N)} = A \bar{x}', \tag{5}$$

where $A^{(i)}, i = 0, 1, 2, ..., N$ denote fuzzy parameters of the model and $\hat{Y}(x)$ is a model fuzzy response for $x$. The fuzzy parameters are assumed to be symmetric triangular fuzzy numbers (STFN) with membership function:

$$\mu_{A^{(i)}}(a) = \max\{1 - \frac{|a - a_i|}{c_i}, 0\}, \tag{6}$$

where $a_i$ is the center and $c_i$ is the spread as depicted in Fig. 1. In other words, center $a_i$ and spread $c_i$ clearly define STFN $A^{(i)}$, as we write $A^{(i)} = (a_i, c_i)$.

Similarly, from (6) by applying the extension principle [14] to the linear combination (5) of fuzzy numbers $A^{(i)} = (a_i, c_i)$ we get:

$$\mu_{\hat{Y}(x)}(y) = \max\{1 - \frac{|a\bar{x}' - y|}{c|\bar{x}'|}, 0\}, \tag{7}$$

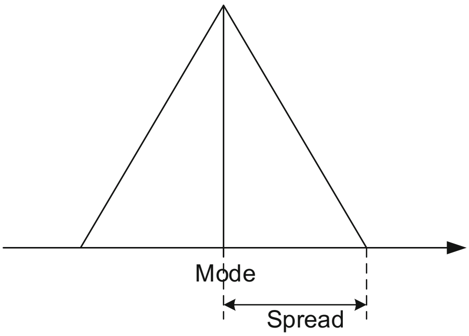


**Fig. 1.** Parameters of symmetric triangular fuzzy number.

where $a = (a_0, a_1, \ldots a_N), c = (c_0, c_1, \ldots c_N)$. It results that fuzzy model response $\hat{Y}(x)$ is also STFN with membership function (7) or equivalently:

$$\hat{Y}(x) = (\hat{y}(x), \hat{e}(x)) = (\sum_{i=0}^{N} a_i x^{(i)}, \sum_{i=0}^{N} c_i x^{(i)}). \tag{8}$$

In order to conduct learning procedure or fit the fuzzy regression model $\hat{Y} = F(A, x)$ to the output observations $\{y_1, y_2, \ldots y_n\}$ belonging to the training set (2) two different approaches are used [2]:

1. The possibilistic approach, which minimizes the fuzziness of the model by minimizing the total spreads of its fuzzy coefficient:

$$J[F(A = (a, c), x)] = nc_0 + \sum_{i=1}^{n} [c_1, c_2, \ldots c_n] \, x_i' \rightarrow \min_c, \tag{9}$$

   subject to including the data points of each learning object within a specific feasible data interval.
2. The least-square approach. Let $\gamma_i$ be a measure of compatibility between training data $y_i$ and the response $\hat{Y}(x_i)$ of FLR model. The value of $\gamma$ is between zero and unity. $\gamma_i = 1$ if $y_i = \hat{y}_i$ (center of $\hat{Y}(x_i)$) and $\gamma_i = 0$ if $\mu_{\hat{Y}(x_i)}(y_i) = 0$. The least-square approach minimizes the sum of squares of the deviations of $\gamma_i$ from unity, i.e.:

$$W[F(A, x)] = \sum_{i=1}^{n} (1 - \gamma_i)^2 \rightarrow \min_A . \tag{10}$$

   For the case of crisp observation and responses $\hat{Y}(x)$ in the form of STFN, measure $\gamma$ can be expressed as follows [2]:

$$\gamma_i = \mu_{\hat{Y}(x_i)}(y_i). \tag{11}$$

Details of both procedures can be found in [2]. In the next section we consider the problem of combining fuzzy linear base models in ensemble approaches for regression with dynamic integration scheme.

## 3 Dynamic Integration of Fuzzy Regression Models

Typically, the process of construction of multi-regressor system called also ensemble approach for regression or ensemble learning [6] can be divided on three phases. The first phase is *ensemble generation* which consists of generating a set of base regression models $\mathcal{F} = \{f_1, f_2, \ldots, f_L\}$ using learning set (2). In this study as base models the FLR models (5) are applied, i.e. we suppose that a set of trained fuzzy linear regression models is given:

$$\mathcal{F} = \{F_1(x, A_1), F_2(x, A_2), \ldots F_L(x, A_L)\}. \tag{12}$$

In the second phase called *ensemble selection* or ensemble *pruning* the ensemble $\mathcal{F}$ is reduced by eliminating some of the models generated in previous phase. Finally, in the *ensemble integration* phase a method of combining selected base models is defined, which is used to generating response of a common model $f_{\mathcal{F}}$ based on the predictions of base models.

In the next subsections two integration methods for FLR models will be presented. Former method is classical approach which is typically applied for crisp responses of base models. The latter method is our original proposition, which denotes integration on fuzzy level of model responses.

### 3.1 Crisp Integration (CI)

General form of integration procedure of base regression models from the set $\mathcal{F}$ is following [9]:

$$f_{\mathcal{F}}(x) = \sum_{l=1}^{L} h_l(x) f_l(x), \tag{13}$$

where $h_l(x)$ denotes the weighting function of the $l$th model. The simplest approach, called Base Ensemble Method (BEM) supposes that weights for all base models have the same value equal to $1/L$. In the more advanced approach (Generalized Ensemble Method - GEM) weights are calculated so as to minimize mean square error for validation data. In the method proposed in [8] and detailed described in [6,11], first we create the *performance matrix* $PM = [p_{il}]_{M \times L}$ which contains an error produced by $l$th model for $i$th validation object. Next, the $k$ best models for validation objects belonging to the defined neighborhood of $x$ get weight value $h_l(x) = 1$ and remaining models get $h_l(x) = 0$.

In considered case with FLR models, fuzzy responses of base models must be defuzified before integration phase conducted according to formula (13). Typically, the defuzzification procedure consists in determination for a given test point $x$ center value of fuzzy response $\hat{Y}_l(x)$, which – for STFN – is equivalent to the center of gravity (CoG) method of defuzzification [15].

### 3.2 Fuzzy Integration (FI)

The proposed original method of integration, unlike the crisp integration (13), combines fuzzy (soft) responses of base FLR models. It means, that first a fuzzy response $\hat{Y}_{\mathcal{F}}(x)$ of an ensemble (12) is calculated and next after defuzzification procedure we get final crisp result $\hat{y}(x) = f_{\mathcal{F}}(x)$.

We propose the following integration methods at fuzzy level which are defined as $L$-variate functions $[0,1] \times [0,1] \times \ldots \times [0,1] \rightarrow [0,1]$ ($\mu_{\hat{Y}_l(x)}(y)$ and $\mu_{\hat{Y}_{\mathcal{F}}(x)}(y)$ denote membership functions of $\hat{Y}_l(x)$ and $\hat{Y}_{\mathcal{F}}(x)$, respectively):

1. The maximum integration (MAX):

$$\mu_{\hat{Y}_{\mathcal{F}}(x)}(y) = \max\{\mu_{\hat{Y}_1(x)}(y), \mu_{\hat{Y}_2(x)}(y), \ldots \mu_{\hat{Y}_L(x)}(y)\}. \tag{14}$$

2. The bounded sum integration (BS):

$$\mu_{\hat{Y}_{\mathcal{F}}(x)}(y) = \min\{1, \sum_l \mu_{\hat{Y}_l(x)}(y)\}. \tag{15}$$

3. The sum integration (S):

$$\mu_{\hat{Y}_{\mathcal{F}}(x)}(y) = \sum_l \mu_{\hat{Y}_l(x)}(y) / \max_y \sum_l \mu_{\hat{Y}_l(x)}(y). \tag{16}$$

4. The weighted sum integration (WS):

$$\mu_{\hat{Y}_{\mathcal{F}}(x)}(y) = \sum_l \lambda_l(x) \mu_{\hat{Y}_l(x)}(y) / \max_y \sum_l \lambda_l(x) \mu_{\hat{Y}_l(x)}(y), \tag{17}$$

   where $\lambda_l(x)$ denotes weight at a point $x$ connected with the $l$th model. In the further experimental investigations, as weight we adopt the inverse spread of response of $l$th model at point $x$, viz.

$$\lambda_l(x) = 1/e_l(x). \tag{18}$$

   It means, that models with crisper response have greater weights than models with uncertain (less precise) response.

All abovementioned methods are formulated according to the BEM approach. Applying the GEM method denotes, that membership functions of base FLR models are multiplied by additional weight $h_l(x)$.

## 4 Experiments

### 4.1 Experimental Setup

In order to study the performance of the proposed fuzzy integration methods of base fuzzy linear regression models some computer experiments were made. The experiments were conducted in Python environment on the computer generated data.

As a reference method the crisp integration procedure (13) was applied. In both crisp (CI) and fuzzy integration (FI) approaches the GEM method was applied with $0-1$ weights calculated according to the performance matrix concept.

**Datasets.** Experiments were conducted on the computer generated data. Each dataset contained 3000 pairs $(x, y)$, where $x$ and $y$ denote scalar value of input and output variable, respectively. Value $x$ was randomly chosen from uniform distribution on $[-4, +4]$ interval and $y = \Phi + \varepsilon$, where $\Phi$ was linear function, 2nd-order function and 3th-order function and $\varepsilon$ was normally distributed random variable with zero mean value and variance $\sigma^2$. For each database all objects were randomly divided into 3 equinumerous disjoint sets: learning set $\mathcal{S}$ for training base FLR models, validation set $\mathcal{V}$ for creating performance matrix and determining the integration methods and testing set $\mathcal{T}$ for evaluation of integration methods. Table 1 summarizes parameters of datasets used in experiments.

**Base Models.** The experiments were conducted using homogeneous ensemble of base models $\mathcal{F}$, the same for both tested integration methods (CI and FI). Ensemble $\mathcal{F}$ consisted of 100 linear fuzzy regression models, which were trained using the possibilistic approach and bootstrapping of the training set.

**Table 1.** Datasets used in experiments.

| Dataset | Function $\Phi$ | Variance $\sigma^2$ |
|---|---|---|
| 1 | $0.1x + 0.1$ | 0.1 |
| 2 | $0.1x + 0.1$ | 0.2 |
| 3 | $0.1x + 0.1$ | 0.5 |
| 4 | $-0.2x^2 + 0.1x - 0.1$ | 0.1 |
| 5 | $-0.2x^2 + 0.1x - 0.1$ | 0.2 |
| 6 | $-0.2x^2 + 0.1x - 0.1$ | 0.5 |
| 7 | $10x^3 - 2x^2 - 1.1x + 0.12$ | 0.1 |
| 8 | $10x^3 - 2x^2 - 1.1x + 0.12$ | 0.2 |
| 9 | $10x^3 - 2x^2 - 1.1x + 0.12$ | 0.5 |

### 4.2 Results and Discussion

Accuracies of CI and FI integrated models evaluated by means of root mean-square error (RMSE) for testing set are presented in Table 2. The results are averaged RMSE values obtained over 40 replications for each experiment. Arrows ↑/↓ mean that the fuzzy integration method is statistically significant better/worse then the reference integration method (CI). The Wilcoxon tests were applied in which the level of $p < 0.05$ was considered as statistically significant.

**Table 2.** Averaged RMSE ± standard deviation for the integration methods tested (description in the text). The best result for each dataset is highlighted in boldface.

| Dataset | Crisp integration | Fuzzy integration | | | |
|---|---|---|---|---|---|
| | | MAX | BS | S | WS |
| 1 | **0.507** ± 0.006 | 0.508 ± 0.017 | 0.519 ± 0.013 ↓ | **0.507** ± 0.012 | 0.508 ± 0.016 |
| 2 | **0.748** ± 0.038 | 0.762 ± 0.021 ↓ | 0.800 ± 0.011 ↓ | **0.748** ± 0.011 | 0.754 ± 0.001 |
| 3 | 3.291 ± 0.121 | 3.111 ± 0.126 ↑ | 3.141 ± 0.142 ↑ | 3.183 ± 0.074 ↑ | **3.103** ± 0.165 ↑ |
| 4 | 0.703 ± 0.021 | 0.708 ± 0.012 ↓ | **0.701** ± 0.013 | 0.703 ± 0.016 | 0.708 ± 0.011 ↓ |
| 5 | 0.973 ± 0.021 | 0.974 ± 0.023 | **0.960** ± 0.012 ↑ | 0.975 ± 0.013 | 0.972 ± 0.016 |
| 6 | 4.034 ± 0.088 | 3.912 ± 0.100 ↑ | **3.467** ± 0.088 ↑ | 3.957 ± 0.130 ↑ | 3.942 ± 0.110 |
| 7 | **0.801** ± 0.021 | 0.804 ± 0.017 ↓ | 0.802 ± 0.017 | 0.802 ± 0.015 | **0.801** ± 0.021 |
| 8 | 1.538 ± 0.022 | 1.539 ± 0.024 | 1.538 ± 0.019 | **1.537** ± 0.018 | 1.538 ± 0.020 |
| 9 | 6.295 ± 0.099 | **5.513** ± 0.101 ↑ | 6.098 ± 0.091 ↑ | 6.171 ± 0.154 ↑ | 6.296 ± 0.088 |

These results imply the following conclusions:

1. The FI methods produced statistically significant better (worse) scores in 11 (6) out of 36 pairwise tests (9 datasets × 4 methods tested);
2. The FI method with bounded sum (BI) operator achieved significant better result in 4 out of 9 pairwise test while FI with MAX and sum (S) operators in 3 pairwise tests;
3. There are no statistically significant differences between RMSE of CI and FI methods or FI methods are significant worse for small (0.1) and medium (0.2) values of variance $\sigma^2$ and linear and 3rd order function $\Phi$;
4. For high value of variance $\sigma^2$ (0.5) and for all forms of function $\Phi$, FI methods with different operators are statistically significant better than CI method.

## 5 Final Remarks

In this study we have demonstrated that the ensemble of FLR models can be integrated using operators defined on fuzzy sets (maximum, sum, weighted sum) which are typically applied in the Mamdani inference system for aggregating conclusions of active rules. Such an approach states an interesting and effective alternative to the traditional methods in which integration is performed on the level of crisp responses of base regression models.

Conducted experimental investigations have preliminary character. In order to fully identify cases in which the fuzzy integration of FLR models is more effective than crisp integration further experimental studies are needed, including multivariate models, non-linear models and distribution of disturbances dependent on input variable.

**Acknowledgments.** This work was supported by the statutory funds of the Department of Systems and Computer Networks, Wroclaw University of Science and Technology.

## References

1. Breiman, L.: Stacked regressions. Mach. Learn. **24**(1), 49–64 (1996)
2. Chang, Y.H.O., Ayyub, B.M.: Fuzzy regression methods - a comparative assessment. Fuzzy Sets Syst. **119**(2), 187–203 (2001)
3. Chatterjee, S., Simonoff, J.S.: Multiple Linear Regression, pp. 1–21. Wiley, New York (2013)
4. Hashem, S.: Optimal linear combinations of neural networks. Neural Netw. **10**(4), 599–614 (1997)
5. Leblanc, M., Tibshirani, R.: Combining estimates in regression and classification. J. Am. Stat. Assoc. **91**(436), 1641–1650 (1996)
6. Mendes-Moreira, J., Soares, C., Jorge, A.M., De Sousa, J.F.: Ensemble approaches for regression: a survey. ACM Comput. Surv. **45**(1), 10:1–10:40 (2012)
7. Merz, C.J.: Dynamical Selection of Learning Algorithms. In: Fisher, D., Lenz, H.-J. (eds.) Learning from Data. Lecture Notes in Statistics, vol. 112, pp. 281–290. Springer, New York (1996)
8. Merz, C.J.: Classification and Regression by Combining Models. Ph.D. thesis (1998). aAI9821450
9. Perrone, M.P., Cooper, L.N.: When networks disagree: ensemble methods for hybrid neural networks. Technical report, DTIC Document (1992)
10. Perrone, M.P.: Improving regression estimation: averaging methods for variance reduction with extensions to general convex measure optimization (1993)
11. Rooney, N., Patterson, D., Anand, S., Tsymbal, A.: Dynamic integration of regression models. In: Roli, F., Kittler, J., Windeatt, T. (eds.) MCS 2004. LNCS, vol. 3077, pp. 164–173. Springer, Heidelberg (2004). doi:10.1007/978-3-540-25966-4_16
12. Tanaka, H., Uejima, S., Asai, K.: Linear regression analysis with fuzzy model. IEEE Trans. Syst. Man Cybern. **12**(6), 903–907 (1982)
13. Weisberg, S.: Applied Linear Regression, 3rd edn. Springer Science+Business Media, Heidelberg (2001)
14. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning–i. Inf. Sci. **8**(3), 199–249 (1975)
15. Zimmermann, H.: Fuzzy Set Theory and Its Applications, 4th edn. Wiley-Interscience, New York (2005)

# Ensemble Machine Learning Approach for Android Malware Classification Using Hybrid Features

Abdurrahman Pektaş and Tankut Acarman(✉)

Computer Engineering Department, Galatasaray University,
Ortaköy, 34349 İstanbul, Turkey
apektas@yandex.com, tacarman@gsu.edu.tr
http://gsu.edu.tr

**Abstract.** Feature-based learning plays a crucial role at building and sustaining the security. Determination of a software based on its extracted features whether a benign or malign process, and particularly classification into a correct malware family improves the security of the operating system and protects critical user's information. In this paper, we present a novel hybrid feature-based classification system for Android malware samples. Static features such as permissions requested by mobile applications, hidden payload, and dynamic features such as API calls, installed services, network connections are extracted for classification. We apply machine learning and evaluate the level in classification accuracy of different classifiers by extracting Android malware features using a fairly large set of 3339 samples belonging to 20 malware families. The evaluation study has been scalable with 5 guest machines and took 8 days of processing. The testing accuracy is reached at 92%.

**Keywords:** Malware · Classification · Feature · Ensemble machine learning

## 1 Introduction

Mobile is rising with all aspects including high-end phones and tablets with powerful processors and Internet high-bandwidth connectivity. From the perspective of end users, mobile applications and services request different authorisations for access to a set of resources like database, preference, files in Android operating system (OS). In consequence, a malware may be deployed to perform malicious activities while using the privileges granted by Android OS. Android market share was 83,3% and 1.4 billion Android smartphones were sold in 2015, [16]. The growing market in smartphones and diversity in mobile applications has lead to smartphones becoming attractive target for online criminals. Again in 2015, 3.3 million applications were identified as malware targeting to steal valuable personal information. Malware writers can simply release variant of a malware

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_20

by using obfuscation techniques. But since malware features are unique and variant of a malware belongs to a specific class, learning based malware classification subject to a large volume of samples is crucial for enhancing the OS security and protecting safety-critical user's information. A large variety of information-rich features is used for accurate classification of a malware sample and depending on the variety and volume of a sample set, different feature extraction and learning, decision algorithms are studied. A survey on Android security is presented in [7]. Although signature based security solutions, or namely static analysis based methods, are vulnerable to obfuscation, behavior based and dynamic analysis solutions are more reliable and accurate at detection and classification of a malware variant. In [6], N-gram searching over the static code of malware samples is used to create a classifier and a learning database is developed to determine whether an application is a malware or benign. In [12] system functions such as API calls and permissions are discovered by static analysis from applications' profile and then, classification of a sample into a malware or goodware is evaluated with a set of 1200 malware and 1200 benign samples. Permissions and events requested by Android applications are monitored, clustering and classification by K-means algorithm and a decision tree learning algorithm is studied with a set 500 sample Android applications in [4]. One-Class Support Vector Machine is trained by using the extracted permissions and a control flow graph of the input applications in [13]. The One-Class Support Vector Machine was adequate to reach at the level 80% in accuracy due to a larger set of 2081 benign sample in comparison to 91 malicious Android applications used for training. A set of 1000 malware samples belonging to 49 families and 1000 benign applications is statically analysed in [10]. Then, the Bayesian-based classifier is developed by using the features extracted from this set of known malware and benign Android applications. In [15], text mining and information retrieval is applied to a dataset of Android OS malware families for discovering similarity between those samples and an automated system is deployed to classify malware samples. In [17], a platform deployment and testing is presented. The API calls and event flows of users are collected and 15 types of features are extracted. Support Vector Machine (SVM), Decision Tree (DTree) and Random Forest algorithm is applied for classification of 666 Android applications. In our study, we apply machine learning and evaluate the level in classification accuracy of different classifiers by using the extracted Android malware features from 20 malware families with a fairly large set of 3339 samples. The analysis platform is scalable with 5 guest machines and analysis study took 8 days of processing. The testing accuracy is reached at 92%. The paper is organized as follows: In Sect. 2, we propose a novel hybrid feature-based classification system for Android malware. In Sect. 3, we define the dataset, malware families and features. Then, we apply and evaluate online machine learning algorithm to classify malware samples subject to given performance metrics. Finally, some conclusions are given.

## 2 Methodology

In this section, we propose a novel hybrid feature-based, i.e., static and dynamic features, classification system for Android malware. The proposed system considers static features, including permissions, contains embedded APK and hidden payload, etc. and dynamic features, including API calls, installed services, network connections to model android applications. The proposed methodology, as shown in Fig. 1, consists of five major steps. The first step is dynamic analysis of Android malware samples by Cuckoo Framework. In this step, cuckoo run samples on a virtualized environment and report their activities and some static features of the samples for example, requested permission list extracted from AndroidManifest.xml file in APK file. The second step consists of extracting appropriate features from analysis reports. In this step, we also choose some text-based features to characterize application such as methods (API calls), permissions and HTTP connections. The entire feature set is given in Table 1. Since these features need to be pre-processed before the stage of machine learning, the bag of words approach is used along with normalizing features by the means of Tf-idf transformer. After this pre-processing stage, textual features are represented with sparse matrix. In the next step, feature selection is applied on sparse matrix. We use a meta-transformer (specifically SelectFromModel in scikit-learn) along with Extra Trees Classifier [8] to select the best subset of features from the malware dataset. As a result of this process, each application is represented as feature matrix and a class label indicating Android application family. The fourth part includes the building classification model based on feature set. As Android applications are vectorised into a sparse matrix, one can simply feed this matrix into any machine learning algorithm to derive classification model. However, in our experiments, we examine three classification methods, which are more suitable to high dimensional feature space, including Logistic Regression, Naive Bayes and meta-classifier Ran-

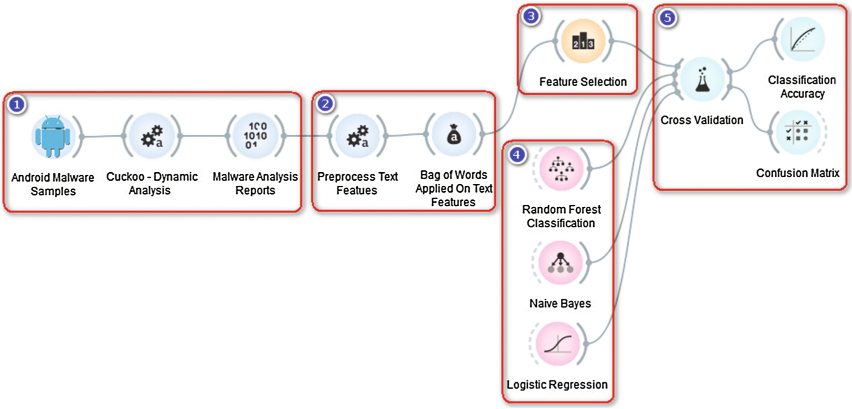


**Fig. 1.** Overview of the proposed methodology

**Table 1.** Features and their types for malware with MD5 value equals f6b60dfdab6558e26f0e92972573d0c8

| Feature category | Type | Value |
|---|---|---|
| installed service count | Integer | 5 |
| fingerprint | String | getSimCountryIso, getDeviceId, getSimOperatorName, getLine1Number |
| methods | String | isVisibleToUser, getFrameTime, performAction, ... |
| permissions | String | Internet, Access_Network_State, Read_Phone_State, Get_Account |
| HTTP connections | String | http://d.applovin.com<br>http://houseads.eu |
| send_sms | Boolean | False |
| receive_sms | Boolean | False |
| read_sms | Boolean | False |
| call_phone | Boolean | False |
| app_execute_shell_commands | Boolean | True |
| app_queried_account_info | Boolean | True |
| app_queried_installed_apps | Boolean | False |
| app_queried_phone_number | Boolean | True |
| app_queried_private_info | Boolean | False |
| app_recording_audio | Boolean | False |
| app_registered_receiver_runtime | Boolean | True |
| app_uses_location | Boolean | False |
| embedded_apk | Boolean | False |
| is_dynamic_code | Boolean | True |
| hidden_payload | Boolean | False |
| is_native_code | Boolean | False |
| is_reflection_code | Boolean | True |

dom Forest Classification. In Sect. 3, we briefly describe these algorithms and elaborate the performance metrics, then plot confusion matrix for better illustration of levels in class-wise classification accuracy.

## 3 Experiments

In this section, we elaborate the prediction accuracy of algorithms subject to the feature set. Our aim is to demonstrate that combining static and dynamic features of Android application can improve classification accuracy. For this purpose, we evaluate our approach on real world malware samples. Classification experiments are carried out on a 2.5GHz Intel 4-Core i-7 processor with 8GB physical memory, using scikit learn [1,11] and MS Windows 10.

### 3.1 Dataset

The benchmark malware dataset is obtained from "VirusShare Malware Sharing Platform" [2]. This platform provides up-to-date malware samples from various types including windows executable (exe, dll), javascript, Android samples (APK), Java, PDF. The Ubuntu 16.04 desktop operating system installed on a hardware Intel(R) Core(TM) i5-2410M@2.30 GHz processor and 2 GB of memory is used to conduct the experimental study. The analysis was scalable with 5 guest machines and took 8 days of processing 3339 samples. To label malware samples, Virustotal as an online web-based multi anti-virus scanner is used [3]. The malware classes and class-specific measures are written in Table 2.

**Table 2.** Malware families and their class-specific measures

| Class | Code | Count | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| android.adware.adwo | 0 | 97 | 0.75 | 0.82 | 0.78 |
| android.adware.appquanta | 1 | 70 | 1.00 | 0.67 | 0.80 |
| android.adware.dowgin | 2 | 867 | 0.86 | 0.99 | 0.92 |
| android.adware.gingermaster | 3 | 89 | 0.83 | 0.62 | 0.71 |
| android.adware.kuguo | 4 | 63 | 1.00 | 0.25 | 0.40 |
| android.adware.plankton | 5 | 42 | 1.00 | 1.00 | 1.00 |
| android.adware.utchi | 6 | 45 | 1.00 | 1.00 | 1.00 |
| android.adware.wapsx | 7 | 127 | 0.75 | 0.82 | 0.78 |
| android.adware.youmi | 8 | 125 | 0.86 | 0.75 | 0.80 |
| android.exploit.gingerbreak | 9 | 37 | 0.50 | 1.00 | 0.67 |
| android.exploit.psn | 10 | 41 | 0.00 | 0.00 | 0.00 |
| android.riskware.agent | 11 | 93 | 0.50 | 0.10 | 0.17 |
| android.riskware.smspay | 12 | 311 | 0.82 | 0.88 | 0.85 |
| android.riskware.smsreg | 13 | 133 | 0.73 | 0.47 | 0.57 |
| android.trojan.agent | 14 | 84 | 0.70 | 0.78 | 0.74 |
| android.trojan.clicker | 15 | 50 | 1.00 | 1.00 | 1.00 |
| android.trojan.fakeinst | 16 | 207 | 0.96 | 1.00 | 0.98 |
| android.trojan.smskey | 17 | 75 | 0.67 | 0.33 | 0.44 |
| android.trojan.smssend | 18 | 81 | 0.88 | 0.70 | 0.78 |
| trojan.java.smssend | 19 | 702 | 0.81 | 1.00 | 0.90 |
| **Average** | | | 0.82 | 0.84 | 0.81 |

### 3.2 Evaluation Metrics

The proposed classification method is evaluated by using the following metrics: ***precision***, ***recall*** (a.k.a. sensitivity), ***F1-score***, ***classification accuracy*** (the overall correctness of the model). In binary classification (positive and negative classes), true positives (*tp*) refer to the correctly predicted positive samples, while true negatives (*tn*) are the number of the correctly predicted negative samples. False positives (*fp*) refer to the incorrectly classified positive samples. Similarly, false negatives (*fn*) are the number of incorrectly classified negative samples. On one hand, the terms positive and negative denote the classifier's success, on the other hand true and false determines whether or not the prediction is matched with the actual (i.e., ground truth) label. The precision is the proportion of the sum of true positives versus the sum of positive instances. For instance, it is the probability for a positive sample to be classified correctly. The recall is the proportion of instances that are predicted positive and are also actually positive (i.e., *tp*) of all the instances that are positive. The F1-score, also known as F-measure or F-score, is the weighted harmonic mean of the precision and recall. F1-score reaches at its best value 1 and the worst score at 0. In binary classification problem, the precision and recall contribute equally to F1-score. However, in the multi-class problems, overall F1-score is calculated by taking the weighted mean of the F1-score of each class. The F-score is a popular measure used in the natural language processing tasks. The metrics are given as follows:

$$precision = \frac{tp}{tp + fp} \tag{1}$$

$$recall = \frac{tp}{tp + fn} \tag{2}$$

$$F1 - score = 2 \times \frac{precision \times recall}{precison + recall} \tag{3}$$

$$accuracy = \frac{correctly\ classified\ instances}{total\ number\ of\ instances} \tag{4}$$

### 3.3 Classification Methods

We evaluate different classifiers, including logistic regression, naive bayes, random forest. The objective is to assess whether combining static and dynamic feature can provide sufficient information in describing Android application. Additionally, we also aim to determine the best classification method in terms of classification accuracy. In machine learning, multinomial Logistic Regression algorithm ([14,18]) is a classification technique that learns and fits the data based on logit function (or logistic curve). In order words, multinomial logistic regression is aimed to predict the outcomes of a multi-class problem according to given a set of feature including real-valued, binary-valued or categorical-valued. It is capable of dealing with a large number of features. The Multinomial Naive Bayes (MNB) classifier ([9]) is a variation of the probabilistic classification algorithm based on the Bayes' Rule. The algorithm assumes that the attributes, namely features, are independent from each other given the class. MNB classifier is highly scalable method for high dimensional dataset and also it takes linear time to make a prediction. As a result, it is a suitable method for text classification problem such as spam detection, information retrieval and text categorisation. Random Forest ([5]) is an ensemble method designed to increase the accuracy of the decision tree by using collection of decision tree. Each tree is trained on randomly selected features, and each tree votes for the most popular class. Then, the output of the classifier is determined by integrating the votes of trees. Consequently, the Random Forest algorithm can handle high dimensional feature space while being computationally less expensive when compared to other ensemble methods. Moreover, employing a set of trees leads a significant increase in classification accuracy.

### 3.4 Results

We used 10-fold cross-validation approach to measure accuracy. However, as our dataset is imbalanced in terms of the number of samples in each class, we adopted Stratified K-Folds method through the evaluation process. Stratified K-Folds validator splits the data into train and test sets by preserving the percentage of the samples for each class. Table 3 shows general classification accuracy and average recall, precision and F1-score for the tested machine learning algorithm. According to our evaluation, meta-classifier Random Forest outperforms the other two classifiers and achieves the highest accuracy with 84%. We also consider a base class that is constituted by trojan, riskware and adware targeting Android OS. This base class is primarily the largest set of variants that need to be identified at day-0. The classification accuracy of Random Forest applied to the base class is reached at 92%. The accuracy of the classifier at recognising instances of different classes is illustrated with the confusion matrix as plotted in Fig. 2. The confusion matrix compares the number of correct and incorrect predictions of the classifier with respect to the ground truth (actual classes). From confusion matrix, it can be seen that the model is successful in determining the majority of the malware families. However, one can remark that, android.exploit.psn (numbered as 10 for the sake of appearance) family was always wrongly classified as trojan.java.smssend (numbered as 19). Another remark is that due to the usage of imbalanced dataset, the classifier tends to classified malware samples

**Table 3.** Classification accuracies of the tested machine learning algorithms subject to all classes and base class

| Algorithm | Accuracy | | Precision | | Recall | | F1-score | |
|---|---|---|---|---|---|---|---|---|
| | All classes | Base classes | All | Base | All | Base | All | Base |
| Random forest | 0.84 | 0.92 | 0.82 | 0.92 | 0.84 | 0.92 | 0.81 | 0.92 |
| Logistic regression | 0.81 | 0.91 | 0.81 | 0.91 | 0.81 | 0.91 | 0.78 | 0.91 |
| Naive Bayes | 0.64 | 0.83 | 0.53 | 0.83 | 0.64 | 0.82 | 0.53 | 0.82 |

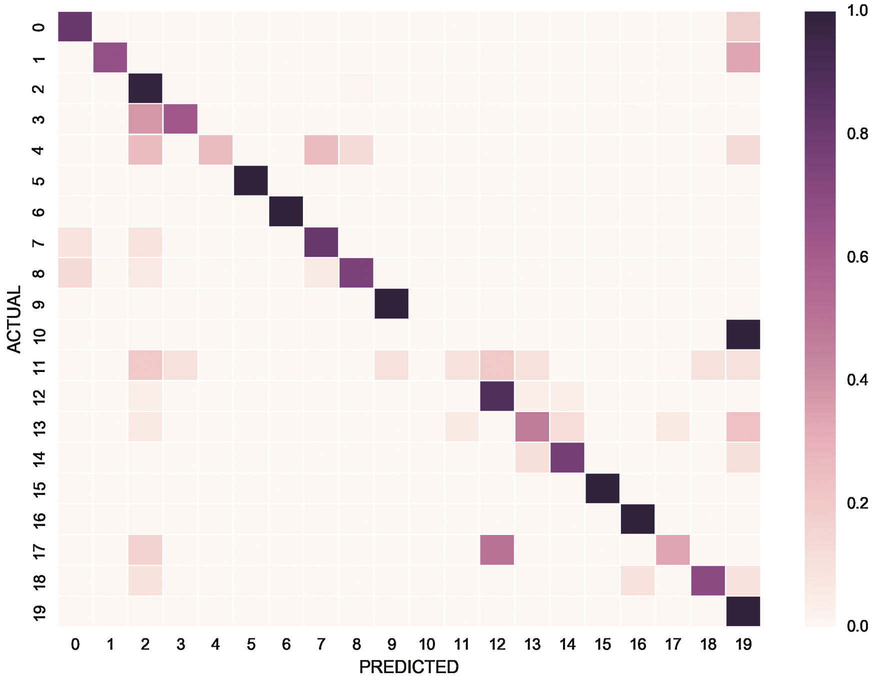


**Fig. 2.** Normalized confusion matrix for all classes

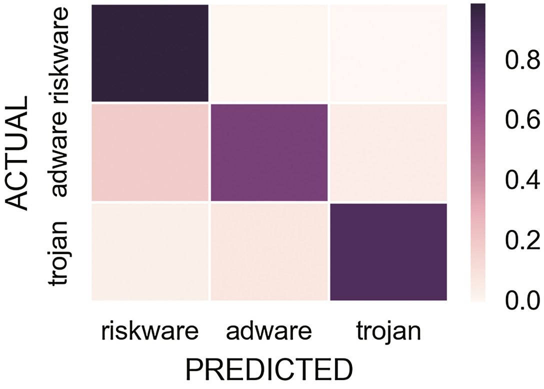


**Fig. 3.** Normalized confusion matrix for base classes

into android.adware.dowgin and trojan.java.smssend families, which are the first two classes that contain the most members in our dataset. Figure 3 shows the class-wise classification accuracy for the base classes, and the model is capable of predicting each base class with high accuracy.

## 4 Conclusions

In this paper, we propose a novel classification method for Android malware samples according to their static and dynamic features (i.e., hybrid features). The proposed framework executes the given samples in a virtualized environment and extracts its activities such as API calls, network connections, and so on. Then, that information is further combined with static features particularly with a permission list provided by the application. By applying learning algorithms to the benchmark datasets, the Random Forest algorithm achieved the highest accuracy at 84% while classifying applications into their respective families. Our experiments on real-world Android malware samples verify the contributions of the proposed method at identifying the Android families on a basis of their extracted hybrid features.

**Acknowledgements.** The authors gratefully acknowledge the support of Galatasaray University, scientific research support program under grant #16.401.004.

## References

1. Scikit-learn: Machine Learning in Python. http://scikit-learn.org/stable/index.html. Accessed 15 Jan 2017
2. Virusshare: Malware Sharing Platform. https://virusshare.com/. Accessed 15 Jan 2017
3. Virustotal: Free Online Virus, Malware and URL Scanner. https://www.virustotal.com/. Accessed 15 Jan 2017
4. Aung, Z., Zaw, W.: Permission-based android malware detection. Int. J. Sci. Technol. Res. **2**, 228–234 (2013)
5. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
6. Dhaya, R., Poongodi, M.: Detecting software vulnerabilities in android using static analysis. In: 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, pp. 915–918, May 2014
7. Faruki, P., Bharmal, A., Laxmi, V., Ganmoor, V., Gaur, M.S., Conti, M., Rajarajan, M.: Android security: a survey of issues, malware penetration, and defenses. IEEE Commun. Surv. Tutorials **17**(2), 998–1022 (2015). (Secondquarter)
8. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**(1), 3–42 (2006)
9. John, G.H., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, pp. 338–345. Morgan Kaufmann Publishers Inc. (1995)
10. McWilliams, G.: Analysis of Bayesian classification-based approaches for android malware detection. IET Inf. Secur. **8**(1), 25–36 (2014). http://digital-library.theiet.org/content/journals/10.1049/iet-ifs.2013.0095
11. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
12. Peiravian, N., Zhu, X.: Machine learning for android malware detection using permission and API calls. In: Proceedings of the 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, ICTAI 2013, pp. 300–305 (2013). http://dx.doi.org/10.1109/ICTAI.2013.53

13. Sahs, J., Khan, L.: A machine learning approach to android malware detection. In: 2012 European Intelligence and Security Informatics Conference, pp. 141–147, August 2012
14. Schmidt, M., Le Roux, N., Bach, F.: Minimizing finite sums with the stochastic average gradient. Math. Program. **162**(1), 83–112 (2017). http://dx.doi.org/10.1007/s10107-016-1030-6
15. Suarez-Tangil, G., Tapiador, J.E., Peris-Lopez, P., Blasco, J.: Dendroid: a text mining approach to analyzing and classifying code structures in android malware families. Expert Syst. Appl. **41**(4), 1104–1117 (2014). http://dx.doi.org/10.1016/j.eswa.2013.07.106
16. Symantec: Internet security threat report (2016). https://www.symantec.com/content-/dam/symantec/docs/reports/istr-21-2016-en.pdf
17. Yang, Y., Wei, Z., Xu, Y., He, H., Wang, W.: Droidward: an effective dynamic analysis method for vetting android applications. Cluster Comput. **19**, 1–11 (2016)
18. Yu, H.F., Huang, F.L., Lin, C.J.: Dual coordinate descent methods for logistic regression and maximum entropy models. Mach. Learn. **85**(1–2), 41–75 (2011)

# An Ensemble of Weak Classifiers for Pattern Recognition in Motion Capture Clouds of Points

Juan Luis Jiménez-Bascones[1,2] and Manuel Graña[2(✉)]

[1] Motion Capture Unit, STT-Systems, San Sebastian, Spain
[2] Computational Intelligence Group,
University of the Basque Country (UPV/EHU), Leioa, Spain
manuel.grana@ehu.es

**Abstract.** This paper deals with the problem of labeling a cloud of points as a classification problem, proposing an ensemble of weak classifiers. First, we define a set of geometrical features over small subsets of the cloud of points. Then, we apply an Adaboost like strategy to select a collection of features achieving a target accuracy in the detection of correct labeling as a whole. Furthermore, we use these features to generate the labeling of the points in the cloud. We demonstrate the approach on a real dataset obtained from the measurement of gait motion of persons, for which the ground truth labeling has been carried out manually. Results are encouraging, achieving high accuracy in both tasks (correct label detection and label generation) at a reduced computational cost.

## 1 Introduction

The process of recording the movement of objects (very often human bodies) so that they are digitized into a computer model is known as *motion capture* (MoCap). This technology is widely employed in many scientific and industrial fields like entertainment, clinical analysis, rehab and sports. A paradigmatical application is gait analysis [10,11]. Motion capture has diverse technological solutions. Capture systems can be optical (where a set of cameras is used to record the movement) and non-optical (those that use inertial, magnetic or mechanical devices [10,11]). Optical systems capture the movement by means of a set of calibrated and synchronized cameras deployed around the *scene*, recording images at a constant frame rate. Frame by frame, a set of 2D points (or markers in the MoCap slang) are extracted from the camera images and sent to the computer for further processing. The 3D coordinates of a point cloud are computed by photogrammetric techniques [1–3] but at this stage their individual ID remains unknown. The next task is to uniquely identify the candidates points one by one, frame by frame, what makes further biomechanical calculations possible. We have to point out here that there is no other information than cartesian coordinates: no color codes, surrounding image or fiducial schemes. The main motivation of this paper is focused on solving the labeling process. The labeling process is a crucial task in the motion capture flow and it is not possible to get

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_21

away without it. While a fully automatic and reliable error free mechanism is desirable, very often manual intervention has to be carried out to correct wrong marker identification. Such verification happens to be a time consuming and tedious task and requires skilled users willing to spend some time figuring out who's who. Not to mention that this slows down the attainment of biomechanical data messing up applications where real time is a requirement. Our main motivation is to provide new tools to ease the marker labeling. In addition we try to tackle the problem using machine learning inspired techniques that are supposed to fit better to the nature of the problem. Finally, our proposal differ from the common solutions found in the existing literature, offering an alternative approach. The main contributions in this paper are the following: (1) Formulation of the labeling correctness as a classification problem; (2) Proposal of a way of computing geometrical features over the cloud of points which allow to define weak classifiers; (3) An Adaboost approach to build the ensemble classifier from a collection of weak classifiers; (4) Label generator by using the weak classifiers to guide the process; (5) We demonstrate the validity of approach on a large dataset obtained from the real industrial practice of the first author.

## 2 Related Work

The first natural approach [4,5] is to try keeping the labeling through the time using trajectory estimators, appraising next markers positions from their positions in the previous frames. In some cases, such prediction is achieved by means of a Kalman filter tuned to fit the each particular marker behavior. Given an estimation on the movement, an energy or strength goal function is formulated between the predicted trajectory and the provided point cloud and some kind of minimization algorithm is performed to assign the labels [12,13]. The value to optimize is very often the mean distance between the candidates and the predicted marker positions [14–16] while the minimization algorithm is a tailored implementation of the well known Hungarian method [17]. However this strategy turns out to be error prone when it comes to deal with marker occlusions (points kept out of sight of the cameras) lasting several consecutive frames. In absence of a reliable trajectory estimation the goal function becomes untrustworthy to assess the right labeling. On the other hand, the appraisal of future marker movement based in its recent trajectory is simply too weak for abrupt movements. As it has been pointed out, it is like *trying to drive your car while staring at the rear view mirror*. So as to strengthen the marker labeling recovery after a long lasting occlusion, some authors take advantage of the underlying human skeleton by the identification of the markers belonging to the same body limb. The markers can be clustered analyzing the pairwise distance along the time keeping in mind that the skin movement and other artifacts prevents us from using classical rigid body restrictions. The identification of a reappeared maker is backed up by those sharing the same limb. This method may fail in case of massive occlusions where nearly all markers from the same limb have been hidden for too long. Some authors [14,16] face the most adverse situations going beyond the marker clustering, exploiting the fact the markers are placed over a articulated mechanism. Not only do the markers belong to the same rigid bodies and therefore the distances among them are supposed to remain the same along the time [6,7], but also the limbs are linked between them by means of physical joints. Hence, the overall range of movements is limited. In other words, they suggest to make use of kinematic (direct or inverse depending on the author) calculation techniques. Thanks to them and in terms of mechanical engineering, the number of *degrees of freedom* (DOF) of the underlying mechanism is restricted and so is the feasible marker labeling. This contributes to identify markers with a time dragged occlusion. In addition to the academic approaches mentioned above, there is a number of commercial solutions available for marker tracking such as *Cortex* (developed by Motion Analysis), *Track Manager* (from Qualisys) or *Clima* (by STT Systems). By and large, little has been written about the details of the internal tracking mechanism they implement due to the proprietary nature of these packages. The only information is provided by descriptive brochures or fliers. For example, it is known that the Qualisys software uses an tracking algorithm (named AIM which stands for Automatic Identification of Markers). Basically it is a marker classifier that learns from each manually verified track. What this means in practice is that after labeling each marker the underlying model is updated. When a new track is provided to the system, it applies the model and attempts to automatically label the markers over the whole trial, automatically filling in gaps of certain sizes. Apparently, huge benefits are obtained when markers flicker or disappear for short periods of the movement since the AIM model automatically labels them when they reappear.

## 3 Problem Statement

Motion Capture systems using passive markers require to place a set of $n$ reflective points over the object whose movement has to be tracked. Each marker has a predefined and constant position over the body and a unique ID or name is given. Hence, we have the set of markers that build the *model* as $\{M\} = \{M_1, M_2, ..., M_n\}$, where each element is uniquely labeled with anatomical informative names as '*right-shoulder*', '*left-knee*' or '*left-humerus-lateral-epicondyle*'. At some stage of the motion capture process, a set of unlabeled 3D points is provided to the system. These points, known as *candidates* are gathered in $\{C_t\} = \{C_1^t, C_2^t, ..., C_m^t\}$, where $t = \{0, 1, 2, ..., T\}$ makes reference to the frame or time from where they were extracted. When $m \neq n$ some real marker is hidden to the cameras (occlusion) or *ghost* points make their appearance on the scene (a ghost is a 3D point built from a wrong 2D matching, so that it does not correspond to a real marker – a kind of mirage). The challenge we undertake at this stage is to correctly match the elements from $M$ and $C$ with the only help of geometric information. A labeling of a given frame cloud of points $C_t$ is a correspondence, such as shown in Fig. 1, of the cloud into the model that can be coded as the integer vector $L_t = \{l_1^t, l_2^t, ..., l_n^t\}$ where:

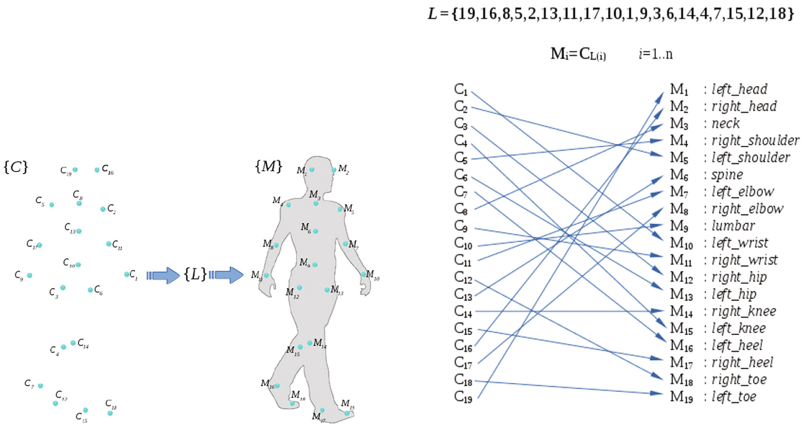


**Fig. 1.** Example of a humanoid model labeling $L$.

$$l_i^t \in \{\mathbb{N}, 0\}\,,\; 0 \leq l_i^t \leq m \tag{1}$$

$$\left(l_i^t \neq 0\right) \Rightarrow \left(l_i^t \neq l_j^t \;\forall j \in \{1,\ldots,n\} - \{i\}\right) \tag{2}$$

So that a numeric non-zero value of $l_i^t$ connects the marker $M_i$ with candidate point $C_{l_i^t}^t$, whereas a zero value means that marker $M_i$ has no match among the candidate points (i.e. it has been occluded). No two elements of $L$ contain the same non-zero mapping since a given candidate cannot be simultaneously assigned to more than one marker.

**Correct labeling detection**

Given a marker model and a set of candidate points, the challenge is to decide whether a given labeling $L$ is correct or not as a whole, i.e. if one label is incorrect the whole labeling is incorrect. We try to build a classifier in two classes, class 1 is the correct labeling

$$\phi\left(M, C_t, L_t\right) = \begin{cases} L_t \text{ correct} \longrightarrow 1 \\ \text{not correct} \longrightarrow 0 \end{cases}$$

**Labeling generation**

The challenge is to generate the correct labeling of the candidate points using the weak classifiers that have been developed for the detection of correct labelings. Here the decision is independent for each point, so we can have an incomplete labeling. We assume that the cloud of points corresponds to the same class of objects upon which the classifier has been trained, for example gait analysis sequences as used in this paper for validation.

## 4 Geometric Features and Weak Classifiers

Given the XYZ coordinates of the points, we define geometric function $g$ yielding scalar values. Examples of geometric functions are listed in the Table 1,

**Table 1.** Several geometric operations

| Geometric property | g | # points | Points | Expression |
|---|---|---|---|---|
| Angle between consecutive angles | $g_1$ | 3 | A, B, C | $\arccos\left(\frac{AB\cdot AC}{\lvert AB\rvert\cdot\lvert AC\rvert}\right)$ |
| Distance between points | $g_2$ | 2 | A, B | $\lvert AB\rvert$ |
| Similarity ratio between segments | $g_3$ | 4 | A, B, C, D | $2\frac{\lvert AB\rvert-\lvert CD\rvert}{\lvert AB\rvert+\lvert CD\rvert}$ |
| Height difference between two points | $g_4$ | 2 | A, B | $A_y - B_y$ |
| Distance ratio between consecutive segments | $g_5$ | 3 | A, B, C | $\frac{\lvert AB\rvert}{\lvert AC\rvert}$ |
| Angle between two segments | $g_6$ | 4 | A, B, C, D | $\arccos\left(\frac{AB\cdot CD}{\lvert AB\rvert\cdot\lvert CD\rvert}\right)$ |
| Angle between a segment and the vertical | $g_7$ | 2 | A, B | $\arccos\left(\frac{AB\cdot Y}{\lvert AB\rvert}\right)$ |
| Triangle area | $g_8$ | 3 | A, B, C | $\frac{1}{2}\lvert AB\times AC\rvert$ |
| $Y$ component of cross vector | $g_9$ | 3 | A, B, C | $\lvert AB\times AC\rvert\cdot\{0,1,0\}$ |

each corresponding to a geometric property of the polygon defined by the set of points. Given a cloud of 3D points, we define a feature as a geometric functions of a subset of points. For instance, if we consider {*left_elbow, left_wrist, right_elbow, right_wrist*} $= \{M_7, M_{10}, M_8, M_{11}\}$ from Fig. 1, a feature would measure the length similarity of forearms segments $g_3(M_7, M_{10}, M_8, M_{11}) = \frac{|M_7-M_{10}|-|M_8-M_{11}|}{|M_7-M_{10}|+|M_8-M_{11}|}$. The number of features grows combinatorially with the size of the cloud, it is of the order of $\begin{pmatrix} n \\ k \end{pmatrix}$, where $n$ is the number of points in the cloud, and $k$ the number of points considered by the feature. Therefore, the possible geometrical functions must be limited, and the effective features must be selected from the pool of all potential features. From the point of view of classification of the entire labeling of the cloud of points, each feature can be taken as a weak classifier, following the approach of Adaboost [9]. In our approach, we consider that each feature has a range of natural values $[\alpha, \beta]$ when the labeling of the cloud of points is correct. For instance $g_3(M_7, M_{10}, M_8, M_{11})$ defined above must have a value near zero, so that $[\alpha, \beta] = [-0.25, 0.25]$ is a feasible interval. Therefore, a weak classifier checks if the feature value is within the specified interval, i.e.

$$h\left(f_k^S\left(M, L_t, C_t\right), \alpha, \beta\right) = \begin{cases} 1 \text{ if } \alpha < f_k^S\left(M, L_t, C_t\right) < \beta \\ 0 \qquad\qquad \text{otherwise} \end{cases}, \tag{3}$$

where $f_k^S$ is a feature built with geometric function $g_k\left(\right)$ applied to a subset of points $S \subset M$ selected from the cloud $C_t$, $[\alpha, \beta]$ its natural interval, and class 1 denotes correct labeling of the cloud of points. Building a strong classifier has the following steps:

1. Generate all the possible features produced by applications of geometric functions to subsets of the cloud of points.
2. Determine the natural interval of values for each feature, thus defining the weak classifiers, as $\alpha_k^S = \min_C f_k^S(C)$ and $\beta_k^S = \max_C f_k^S(C)$, where all clouds $C$ are correctly labeled.
3. Select the minimal collection of features that ensures a given accuracy level of the ensemble of weak classifiers. Since the weak classifiers are trained on the correct labeling, it is easy to see that any collection of them will provide very high sensitivity (accuracy on the target class relative to all examples of the target class) but very likely a large number of false positives, i.e. a very low specificity. Hence our process is a greedy selection of the weak classifier providing the biggest increase of accuracy by decreasing the number of false positives.

## 5 The Ensemble of Weak Classifiers

Let us denote $\mathbf{O} = \{O_i\}$ the set of learning observations $O_i = \{C_i, L_i, b_i\}$ corresponding to a common model $M$. We can easily generalize to the case of several models. Each observation has a cloud of points $C_i$ and the labeling $L_i$ that maps it into the model. The vector $b_i$ encodes the correctness of the mapping, so that $b_{ij} = 0$ if the label of the $j$-th cloud point is incorrect, thus the whole labeling is incorrect. The training algorithm generates incorrect labeled observations by permutation of the labels in a selected correct observation. The number of permuted elements (from 2 to n, the number of markers) is an index of the severity of the labeling error. We denote $\mathbf{O}^* = \{O_i^*\}$ the incorrect samples, retaining $\mathbf{O} = \{O_i\}$ for the correct ground truth observations. The ensemble of classifiers consists of a collection of features whose corresponding weak classifier is weighted by its accuracy gain relative to the remaining weak classifiers. The output of the ensembles is computed as:

$$\phi_J(M, C, L) = \frac{\sum\limits_{j=1}^{J} w_j h_j(f_k^S(C), \alpha_k^S, \beta_k^S)}{\sum\limits_{j=1}^{J} w_j}, \tag{4}$$

where the index j refers to the order of selection of the feature for inclusion in the ensemble, i.e. $J$ is the size of the ensemble. Our method borrows from Adaboost [9] approach the strategy of greedy selection of the weak classifier that maximize the increase in accuracy, in our case the number of wrong labeling detection. Initially, all weights are initialized to zero and the set of selected weak classifiers is empty. In a loop we feed all classifiers with observations of different error severity obtained by permutations of labels in the correct observations. If the actual version of $\phi_J(M, C, L)$ does reject the incorrect sample no further process is done. If not, the weights of unselected weak classifiers that reject it are updated according to the error severity. After a number of incorrect observations is processed, the ensemble is engrossed with the weak classifier having the greatest weight. The whole process eventually ends up when a given threshold on the accuracy of the strong classifier is reached. Finally, the elite of classifiers is stored together with the weight they got during the learning process as kind of score.

# 6 Generating Labels from the Ensemble of Weak Classifiers

Previous section deals with the decision "is the whole labeling correct?", while in this section we aim to generate the labels for the cloud points using the previously trained weak classifiers and the ensemble classifier. Given an ensemble of weak classifiers $\phi_J(M, C, L)$ trained as described above, the number of weak classifiers giving positive outcome can be interpreted as a measure of how well the vector of integers $L$ links the model points $M$ and the candidate points $C$. Therefore, the labeling of a cloud of points can be stated as looking for the value of L that maximizes the number of weak positive classifications up to achieve $\phi_J(M, C, L) = 1$. For the sake of simplicity we will assume that the number of elements $n$ of $M$ and $C$ do match. In other words, no marker is occluded and no points other than the ones to be labeled are present in the input data. In this scenario $L$ can be any of the permutations of the integers between 1 and $n$ and therefore the number of possible configurations for $L$ is $n!$ . Fortunately, we can exploit the structure of the strong classifier in a branch and bound strategy using the following properties:

- classifier $\phi$ can be evaluated over a *partial solution* where only a subset of elements of $L$ as meaningful labels. Weak classifiers using unassigned labels are simply ignored;
- a single weak classifier rejecting a permutation definitively rules it out, so not all the elements of $\phi$ must be computed, hence the approach is a branch-and-bound process;
- a single weak classifier can be computed from a handful of points (usually from 2 to 6) which represents a subset of the vector $L$.

# 7 Experimental Results

## 7.1 Experimental Data and Ground Truth Labelings

The experimental data has been recorded and labeled manually for its use in the training and validation process. It is made up of 70 sequences of non-pathological gait analysis movement involving 14 different people from different ages and body shapes walking at random paces. Each sequence was recorded at 100 Hz. The average duration of the sequences is three seconds, so that we have more than 20.000 point cloud frames. These labeled clouds are the data of class 1 corresponding to correct labeling. Point clouds with incorrect labeling corresponding to class 0 data items are generated by random permutations of the labels of correct labeling data. The point cloud sequence capture follows the Helen Hayes lower train protocol [8], widely applied in clinical gait analysis, with the use of a *STT's* proprietary optical motion tracking system, including 6 infrared synchronized cameras of $800 \times 800$ pixel resolution specially designed for the detection of reflective markers. The proprietary *CLIMA* software[1] automates the whole process of motion capture from camera management, camera calibration, 3D reconstruction, marker tracking and further biomechanical analysis.

### 7.2 Detection of Correct Labeling Results

A set of features have been built using on geometrical functions $g_2$, $g_4$ and $g_9$ from Table 1 so we have 665 weak classifiers. The training algorithm selects 40 weak classifiers. The ensemble classifier achieves an accuracy over 99% after the presentation of more than $10^7$ negative samples with diverse error severity. Table 2 summarizes the best selected weak classifiers that achieve over 93% accuracy. Apparently the strongest weak classifier is the one that prevents the triangle *right asis - left asis - sacrum* from standing far from a horizontal plane. Indeed, the set of training data involves people walking. No bending over or lying on the floor movements are being exposed to the learning process so this restriction is full of meaning. After it, the strong classifier relies on distance features between consecutive marker. This is another way of saying that the length of humans limbs – or consecutive joints – is more limited than the distance among arbitrary parts such as the toes and the hands.

**Table 2.** First selected weak classifiers

| | Weak classifier | Score (%) | Cum score (%) |
|---|---|---|---|
| 1 | *TriangleNormal_Y(R_asis,L_asis,sacrum)* | *18.82* | *18.82* |
| 2 | *Dist(R_malleolus,R_heel)* | *12.91* | *31.74* |
| 3 | *Dist(L_malleolus,L_heel)* | *12.84* | *44.59* |
| 4 | *Dist(R_femoral_epicondyle,R_tibial_band)* | *11.85* | *56.45* |
| 5 | *Dist(L_femoral_wand,L_femoral_epicondyle)* | *11.51* | *67.96* |
| 6 | *Dist(L_tibial_wand,L_meta_h)* | *10.87* | *78.84* |
| 7 | *CoordDiff_Y(R_femoral_wand,R_meta_h)* | *10.43* | *89.28* |
| 8 | *TriangleNormal_Y(sacrum,R_meta_h,L_meta_h)* | *1.90* | *91.17* |
| 9 | *Dist(R_femoral_wand,R_femoral_epicondyle)* | *1.84* | *93.02* |

### 7.3 Label Generation

To validate the labeling algorithm, we apply a 10-fold cross validation strategy. In addition, we focus on the number of required node evaluation as an index of

[1] http://www.stt-systems.com/products/3d-optical-motion-capture/clima/.

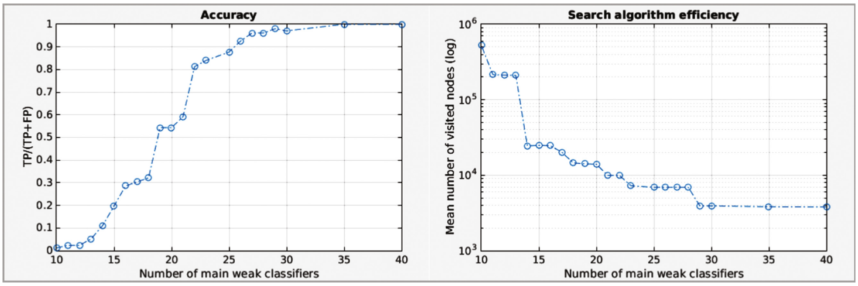


**Fig. 2.** Accuracy and efficiency assessment depending on the number of weak classifiers.

the efficiency of the algorithm: the less the numbers of evaluations, the less is the number of branches to be explored and thus the faster the search. Figure 2 shows how the true positive ratio grows with the number of weak classifiers used for the test of the labeling. However, as mentioned before all classifiers up to the 40th are useful to rule out nearly all the false positives. Regarding the efficiency, the number of node evaluations apparently gets stable around the 30th classifier, requiring a mean of 3500 evaluations before running into the right labeling. This number is rather small compared with $15! > 10^{12}$ (one trillion), the required number of tests in a brute-force search. Note that the graph that relates the number of node evaluations with the number of classifiers is using a logarithmic vertical axis: less than 14 classifiers still require more than 100,000 evaluations.

## 8 Conclusions and Further Work

We have dealt with the problem of labeling a cloud of points according to a given model using a simple classification approach that does not use any semantic prior structural information, such as anatomical locations or graphs of expected relative positions. We define geometrically based features that are evaluated over the cloud of candidate points. Corresponding weak classifiers are defined and trained on the available data. The selection of the minimal collection of weak classifiers achieving a given accuracy threshold is carried out by a greedy selection similar to Adaboost. The experimental validation is carried out over a dataset obtained from the real industrial experience of gait analysis. We achieve encouraging results in both correct labeling detection and label generation tasks. Actually, we have found that the labeling of a cloud of points of the size in this paper can be carried out in times of the order of $10^{-3}$ s, which raises expectation for its use in real time. Future work will be addressed to deal with occlusions, and to make the approach simultaneously valid for several models, i.e. the ensemble may be able to detect which model is best fit for the cloud of points.

## References

1. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000)
2. Faugeras, O.: Three-Dimensional Computer Vision: A Geometric Viewpoint. MIT Press, Cambridge (1993)
3. Heikkila, J., Silven, O.: A four-step camera calibration procedure with implicit image correction. In: Proceeding CVPR 1997 Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR 1997) p. 1106 (1997)
4. Lunardi Flam, D., Pacheco de Queiroz, D., Louise Alves de Souza Ramos, T., de Albuquerque Araújo, A., Victor Boechat Gomide, J.: OpenMoCap: an open source software for optical motion capture. In: 2009 VIII Brazilian Symposium on Games and Digital Entertainment, vol. 0, pp. 151–161 (2009)
5. Guerra-Filho, P.G.: Optical motion capture: theory and implementation. J. Theor. Appl. Inform. (RITA) **12**(2), 61–89 (2005)
6. Yu, Q., Li, Q., Deng, Z.: Online motion capture marker labeling for multiple interacting articulated targets. In: Computer Graphics Forum, vol. 26, no. 3, pp. 477–483. Blackwell Publishing Ltd., September 2007
7. Mehling, M.: Implementation of a low cost marker based infrared light optical tracking system. Kaufmann, H., Institute for Software Technology & Interactive Systems (2006)
8. Vaughan, C.L., Davis, B.L., Connor, J.C.: Dynamics of Human Gait. Kiboho Publishers, Cape Town (1999)
9. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vis. **57**(2), 137–154 (2004)
10. Cutti, A.G., Ferrari, A., Garofalo, P., Raggi, M., Cappello, A., Ferrari, A.: 'Outwalk': a protocol for clinical gait analysis based on inertial and magnetic sensors. Med. Biol. Eng. Comput. **48**(1), 17–25 (2010)
11. Ferrari, A., Cutti, A.G., Garofalo, P., Raggi, M., Heijboer, M., Cappello, A., Davalli, A.: First in vivo assessment of "Outwalk": a novel protocol for clinical gait analysis based on inertial and magnetic sensors. Med. Biol. Eng. Comput. **48**(1), 1–15 (2010)
12. Qiao, M., Cheng, J., Bian, W., Tao, D.: Biview learning for human posture segmentation from 3D points cloud. PLoS ONE **9**(1), e85811 (2014). doi:10.1371/journal.pone.0085811
13. Liu, A.-A., Su, Y.-T., Nie, W.-Z., Yang, Z.-X.: Jointly learning multiple sequential dynamics for human action recognition. PLoS ONE **10**(7), e0130884 (2015). doi:10.1371/journal.pone.0130884
14. Meyer, J., Kuderer, M., Muller, J., Burgard, W.: Online marker labeling for fully automatic skeleton tracking in optical motion capture. In: International Conference on Robotics and Automation (ICRA), pp. 5652–5657 (2014)
15. Schubert, T., Gkogkidis, A., Ball, T., Burgard, W.: Automatic initialization for skeleton tracking in optical motion capture. In: International Conference on Robotics and Automation (ICRA), pp. 734–739 (2015)
16. Maycock, J., Rohlig, T., Schröder, M., Botsch, M., Ritter, H.J.: Fully automatic optical motion tracking using an inverse kinematics approach. In: International Conference on Humanoids 2015, pp. 461–466 (2015)
17. Kuhn, H.W.: The Hungarian method for the assignment problem. Nav. Res. Logistics Q. **2**(1–2), 83–97 (1955)

# Portable Dynamic Malware Analysis with an Improved Scalability and Automatisation

Abdurrahman Pektaş and Tankut Acarman(✉)

Computer Engineering Department, Galatasaray University,
Ortaköy, 34349 İstanbul, Turkey
apektas@yandex.com, tacarman@gsu.edu.tr
http://gsu.edu.tr

**Abstract.** A malware is deployed ubiquitously to steal safety or liability-critical information and damage the compromised systems. In this paper, we present a portable, scalable and transparent system for dynamic analysis of malware targeting Windows OS. The portability feature is enabled by introducing a driver capable of collecting the behavioural activities of analysed samples in low kernel level and detection of a new malware in the latest version of Windows OS is guaranteed without waiting for its signature update. A large volume and variety of malicious behaviour is monitored and analysed by the presented virtual, scalable and automated system deployment. End-to-end design is presented and functional tests of portability feature are conducted by compiling the developed kernel driver component in the analysis machine. Evaluation is performed by using recently captured malware samples that are automatically analysed and detected on a Windows 8 Ultimate 64-bit and Windows 10 OS.

**Keywords:** Malware · Dynamic analysis · Portability · Detection

## 1 Introduction

A malicious software, or malware for short, is a software used or created by an attacker to perform his bad intentions on a computer system without authorisation and knowledge of its user. Stealing safety or liability-critical information and damaging the compromised system is mainly targeted. The recent developments in the field of computation systems and the proliferation of systems such as smart phones, tablet, Internet of Things (IoT), cloud computing have led to an increased interest in malware development. According to [1], more than 430 million new malware samples were released in 2015 with an increase of 36% from the previous year, and a new zero-day vulnerability was discovered at each week on average with an doubled release frequency in comparison with the previous year. Security tools compare suspicious files with their malware definition database,

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_22

which are constructed based on known security issues by analysts. Based on definition database, these tools check whether a given file is malware or not. For example, anti-viruses running on end-user computers are mainly signature-based solutions. From an abstract point of view, they read the suspicious file in a binary format and look for a match with their signature database. According to [2], anti-virus solutions are fast and effective for known malware but their accuracy of analysis and adequacy are easily degraded by slightly changed malware. Although signature-based solutions are subject to delay on identifying new and obfuscated version of malware and updating their databases, they are convenient over previously registered attacks. Subject to an increasing number of sophisticated, advanced, and targeted attacks (a.k.a., advanced persistent threat, APT), security researchers and practitioners have explored more robust and timely solutions to new and unknown threats. Generally, the security community prefers sandboxing where malware activities are monitored during their execution in an isolated and controlled environment. These systems track and inform about file, registry, network, and process activities. To successfully detect malware and take appropriate counter measures, dynamic analysis can be considered as an integrated scenario and solution of an environment provided to malware for being deployed and performing its tasks. CWSandbox leverages API hooking technique in user mode to track malware's activities [3]. Once the sample is loaded into memory, API hooking is performed by in-line code overwriting. The sample is executed in a suspend mode and then all loaded DLL's API functions are overwritten. Hence, CWSandbox collects all called functions and their related parameters. Then, it generates a high-level report about activities and malware analyst can quickly follow them. Since it collects data in user mode, low level operations and undocumented function calls can not be captured. Cuckoo is an open source analysis system and relies on virtualization technology to run a given file [4]. It can analyze both executable and non-executable files. These activities including pre-defined Win API functions and their parameters are monitored and captured by its user-space API hooking technique. Owing to the fact that it runs at user level, malware can easily notice presence of the analysis attempt causing to change its behaviour. Capture-BAT is another dynamic analysis tool developed by New Zealand chapter of honeynet.org [5]. Capture-BAT monitors process, registry, and file activities at kernel level, and it captures network traffic using winpcap library. Furthermore, it offers selection of events through its filtering interface that can be used by the analyst to prevent noisy events to be captured. Since Capture-BAT is not an automated malware analysis system, serious concerns exist on whether it can efficiently handle the high penetration of new and existing malware. To the best of our knowledge, dynamic analysis systems still use old versions of Windows OS, for instance Cuckoo merely employs Windows XP and 7, as an underlying analysis environment. However, computer owners generally prefer to upgrade their OS to the newest version. Therefore, existing systems may fail at analysing malware targeting new versions of Windows OS. Reliability and availability of malware analysis scheme may be a major concern subject to the release of new Windows OS version. Consequently, the next generation of dynamic malware analysis solutions should be adaptable to the future versions of OS. This chapter is an extended version of [6]. In this chapter, kernel callback mechanism is integrated which allows detailed view of run-time events based on defined conditions on a system basis and new malware samples with their targeted features are elaborated and evaluation study is enriched. This chapter is organized as follows: The implementation details of VirMon components and their functionalities are elaborated in Sect. 2. In Sect. 3, two real-world malware samples captured in the wild are used to illustrate the effectiveness and analysis results about monitoring the malware activities. Some conclusions are given in Sect. 4.

## 2 Design of VirMon: System Components and Functionalities

Analysis machine components include mini-filter driver and driver manager. They are responsible for reporting host-based process, registry, and file system activities performed by the analysed file, see Fig. 1. Activities of the process (or processes) initiated by that executable and (if any) child process (i.e., additional

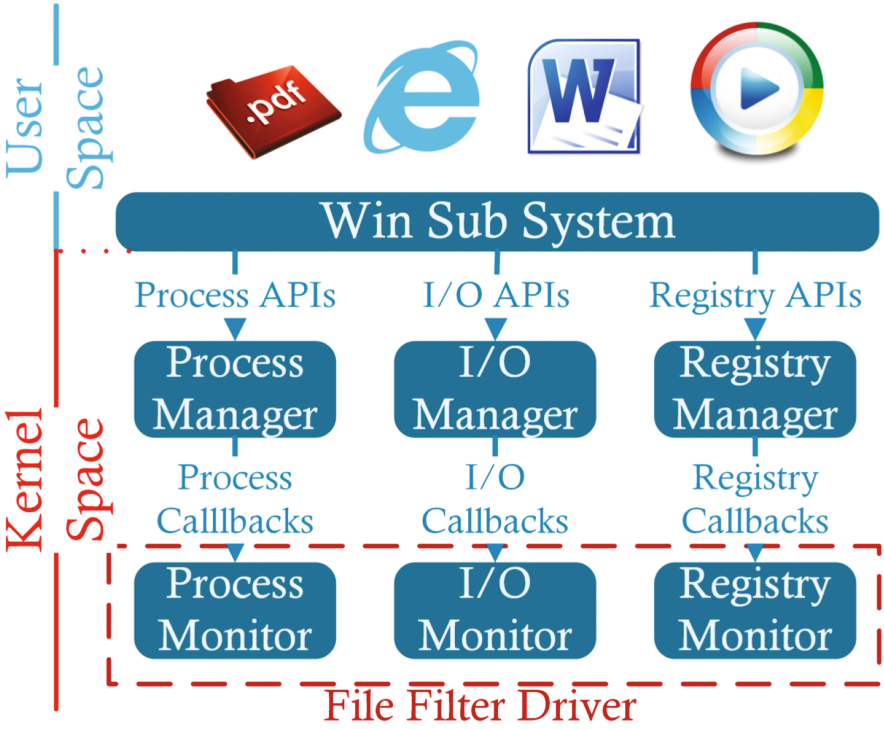


**Fig. 1.** Overview of the analysis machine components (e.g., filter drivers)

process created by some of these processes) are monitored by means of the kernel callback functions being embedded in kernel driver. API hooking is one of the preferred methods for dynamic malware analysis. We use kernel callback mechanism, which provides detailed view of run-time events based on defined conditions on a system basis [7]. To be able to use callbacks, a kernel driver needs to be built. Basically, this driver, also known as mini-filter driver, intercepts all IRP requests made by an application and decides whether allowing or refusing these operations according to the given rule set. The portability is followed by the claim of Microsoft, which states that kernel-callback mechanism is reliable and compatible with all versions of Windows including their 64 bit versions. Malware creates new process or changes an existing one in order to run its actions without being detected. To obtain the information about run-time events provided by the callback mechanism, related functions need to be called with their relevant parameters. For process monitoring, the "PsSetCreateProcessNotifyRoutine" function allows the mini-filter driver to monitor changes applied to the running processes. Many malware uses Windows registry to gain persistent access to the system. Attackers can use these registry keys to gain authority over the OS and persistent access to it. When an event is occurred on the registry, "CmRegisterCallback" function can provide related information to the mini-filter driver. To track registry events, one needs to identify which actions to be monitored in the driver via some self-explanatory constant values (e.g., *RegNtPostCreateKey*, *RegNtPreDeleteKey*, *RegNtEnumerateKey*). In VirMon, registry operations about *OpenKey*, *CreateKey*, *DeleteKey*, *SetValueKey*, *DeleteValueKey*, *QueryValueKey* and *EnumerateKey* are monitored. Malware copies itself or its variants to various locations in the file system and then adds a registry key to start automatically while booting. "FltRegisterFilter" function along with its callback actions can be used to monitor file system activities on the system. Like registry monitoring, the actions to be tracked have to be addressed in the driver accompanied by some constant values (e.g., *IRP_MJ_WRITE*, *IRP_MJ_READ*, *IRP_MJ_QUERY_INFORMATION*). In VirMon, to avoid redundant and distracting file operations, we consider only read, write, and delete events performed by the tracked processes. Network components are responsible for reporting network activities of the analysed file. Malware needs to connect very often to the C&C servers to send confidential information collected from compromised machines or to receive C&C servers' commands. This bi-directional communication makes analysis of the network activities of malware as an inevitable requirement to be fulfilled by malware researchers. In VirMon, we use different network solutions, such as VLAN, VPN, IPDS, and firewall to monitor network activities of suspicious files. Intrusion Prevention and Detection System (IPDS) is a network security solution monitoring network traffic and system activities [8]. In VirMon, an IPDS is introduced to prevent possible networks attacks caused by suspicious files in the system. This secure scheme fulfils the analysis requirement of the malware analyzer by giving an opportunity to acquire all network events including the requested web pages, downloaded files by malware. Suricata, an open source IPDS solution, can prevent malicious attacks such as distributed denial of service (DDoS), port scanning and shell codes [9]. It can also extract files and HTTP requests from live network traffic. In order to circumvent network attacks caused by malware sample under dynamic analysis, we use Suricata as one of the IPDS component. Bro [10] is different from the typical IPDS since it can not block the attacks and does not rely on network signatures but it enables monitoring all network traffic. It supports well-known network protocols, extracts related information from network packets and exposes network activities at high-level [11]. In the hierarchy of VirMon, Bro runs on IPDS server and reads network interface. To extract files from live network traffic, a custom script compatible with HTTP, FTP, IRC, SMTP protocol is created. The developed script logs hostname, URL, filename, file type, and transport layer information (e.g., IPs and ports) to a file which is parsed periodically for storing these information in database. Meanwhile, if the files extracted by Bro Engine have not already been dissected beforehand, they are queued into application server's priority queue. The open source Oracle Virtual-Box [12] is chosen as the virtualization infrastructure to host and to deploy malware analysis machines. Virtual-Box supports both 32 and 64 bit CPU. Virtual-Box also provides accessibility features such as remote machine management, display of multiple remote machines via web interface, and offers command-line interface (VBoxManage) for automated tasks. The application server is responsible for the management of the malware analysis processes. It assigns an analysis machine to the submitted file. It collects the file activities from analysis system components. Then, it formats the collected data and stores them in a database. After the analysis operations are completed, the application server commands the analysis machines to be restored to a clean state.

## 3 Evaluation

The procedure followed by VirMon to analyze a submitted file, and the interaction between the application server and the analysis machine is plotted in Fig. 2:

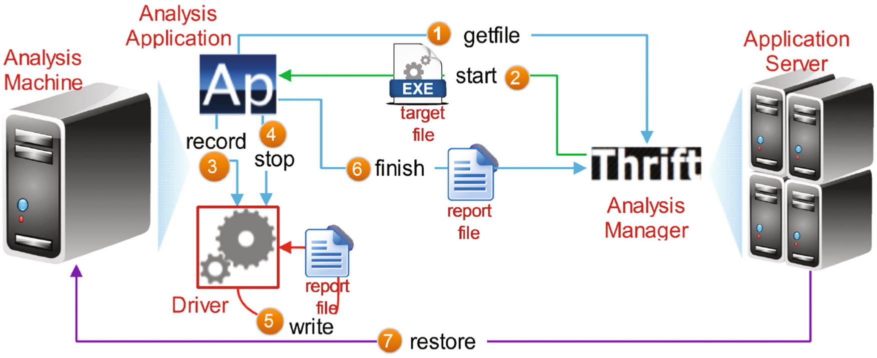


**Fig. 2.** Information flow and interactions between application server and analysis machine

- Analysis application sends a request to the application server to start a new analysis process.
- The application server chooses a file in its priority queue and sends it to a analysis machine.
- The analysis application injects codes into explorer.exe process. This process executes submitted file in the suspended mode and then explorer.exe writes PID of this recently created process to a shared memory. Subsequently, the analysis application reads PID info from the shared memory, sends a message involving this information to the driver and a request about recording the events initiated by this process.
- The analysis application waits until the analysed process exits or a timeout of 3 min occurs. Then, it sends a message to the driver to stop recording events.
- The driver stops recording and writes collected events to a log file.
- The analysis application sends the log file to the application server for parsing.
- The application server parses the log file and stores it in the database. Finally, it reverts the analysis machine to the clean state.

We analyze a trojan, named as Hesperbot [13] detected on August, 2014. This trojan is focused on stealing banking account information to be used towards unauthorized money transfers. The attackers social engineer victims to execute attached files by sending e-mail which looks like it is originated from one of the service providers in Turkey. The analysis of hesperbot is executed automatically on a Windows 8 Ultimate 64-bit OS. Since the number of events including system dll file and registry accesses gathered from VirMon for this malware is too high (10000+), only important events occurred on the system are displayed in Table 1. The intention of the malware sample can be easily derived from this table. When the sample is executed, the process created by explorer.exe creates a new process entitled "fatura_874217.exe" having the same name in its directory. In Turkish, "*fatura*" means "*the bill*". This technique, named as process hollowing, [14,15], has been recently used by malware to hide itself. Then, the process created by explorer.exe terminates itself. The fatura_874217.exe process created by hollowing technique, creates %APPDATA%\Sun and %APPDATA%\yseszpkf directories and drop randomly named binary files under them. To be hidden, the fatura_874217.exe process creates new explorer.exe, which in turn is used to carry out remaining activities, download configuration files from C&C server, drop new executable and writes it to the auto-start line in the registry, respectively. This analysis shows that the VirMon dynamic malware analysis system successfully collects the run-time behaviours of the file sample.

### 3.1 VirMon Compatibility on Windows 10

In order to show that VirMon (e.g., its mini-filter driver) is adaptable to the newest version of Windows OS, we conducted functional testing of developed mini-filter driver on Windows 10. This test is based on the analysis results of the recent malware sample, known as cyrptolocker, a variant of ransomware, that encrypts sensitive documents on the infected machine and forces to pay a ransom

**Table 1.** Important run-time activities of a trojan

| Time | Event | Process | Detail |
|---|---|---|---|
| 26/08/2014 2:48:44.423 | Create Process | C:\Windows\explorer.exe | fatura_874217.exe (analysed sample) MD5:186C097B9D85B3501EFCC4D8D374AFE1 |
| 26/08/2014 2:48:55.790 | Create Process | %Desktop%\fatura_874217.exe (analyzed sample) | fatura_874217.exe (analyzed sample) |
| 26/08/2014 2:48:55.920 | Terminate Process | fatura_874217.exe (analyzed sample) | - |
| 26/08/2014 2:48:56.264 | Create Folder | fatura_874217.exe | %APPDATA%\yseszpkf |
| 26/08/2014 2:48:56.269 | Create Folder | fatura_874217.exe | %APPDATA%\Sun |
| 26/08/2014 2:48:56.468 | Create File | fatura_874217.exe | %APPDATA%\yseszpkf\yqoletyz.dat |
| 26/08/2014 2:48:56.687 | Create File | fatura_874217.exe | %APPDATA%\Sun\yqoletyz.bkp |
| 26/08/2014 2:48:57.299 | Create Process | %Desktop%\fatura_874217.exe (analyzed sample) | C:\WINDOWS\system32\attrib.exe |
| 26/08/2014 2:48:57.301 | Terminate Process | %Desktop%\fatura_874217.exe (analyzed sample) | - |
| 26/08/2014 2:48:57.542 | Create Process | C:\WINDOWS\system32\attrib.exe | C:\Windows\explorer.exe |
| 26/08/2014 2:48:57.882 | Terminate Process | C:\WINDOWS\system32\attrib.exe | - |
| 26/08/2014 2:48:58.063 | DNS Query | C:\Windows\explorer.exe | followtweetertag.com |
| 26/08/2014 2:48:59.272 | Send Data | C:\Windows\explorer.exe | https://followtweetertag.com (possibly download config files) |
| 26/08/2014 2:49:01.120 | Create File | C:\Windows\explorer.exe | “yqomswoc.bkp, ajukiveq.bkp, yqoletyz.bkp under %APPDATA%\Sun ” |
| 26/08/2014 2:49:01.715 | Create File | C:\Windows\explorer.exe | “ajukiveq.dat, cfovpdiq.dat, oquthmjk.dat, yqoletyz.dat, yqomswoc.dat under %APPDATA%\yseszpkf” |
| 26/08/2014 2:49:02.012 | Create File | C:\Windows\explorer.exe | C:\windows\esem\ohotuzuf.exe (MD5: D082B6AD2F24040E6D651D271823D51C) |
| 26/08/2014 2:49:02.114 | Create Reg Key | C:\Windows\explorer.exe | HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\qzofpbuk=C:\windows\esem\ohotuzuf.exe |
| 26/08/2014 2:49:02.345 | Send Data | C:\Windows\explorer.exe | https://webislemx.com (for further commands) |

**Table 2.** Run-time activities of the cyrptolocker on Windows 10

| Time | Event | Process | Detail |
|---|---|---|---|
| 29/11/2014 16:15:53.478 | Create Process | C:\Windows\explorer.exe | %DESKTOP%\fatura_892738105.exe (MD5: 76387075C90533AAD14E82A5D94E8486) |
| 29/11/2014 16:15:53.678 | Create Process | %DESKTOP%\fatura_892738105.exe | %DESKTOP%\fatura_892738105.exe |
| 29/11/2014 16:15:53.755 | Terminate Process | %DESKTOP%\fatura_892738105.exe | - |
| 29/11/2014 16:15:53.964 | Create Folder | %DESKTOP%\fatura_892738105.exe | %APPDATA%\ytivyteqyfypequs |
| 29/11/2014 16:15:53.973 | Create File | %DESKTOP%\fatura_892738105.exe | %APPDATA%\ytivyteqyfypequs\01000000 |
| 29/11/2014 16:15:54.055 | Create Process | %DESKTOP%\fatura_892738105.exe | %WINDOWS%\explorer.exe |
| 29/11/2014 16:15:54.888 | Terminate Process | %DESKTOP%\fatura_892738105.exe | - |
| 29/11/2014 16:15:54.906 | Create File | %WINDOWS%\explorer.exe | %WINDOWS%\whdhufel.exe (MD5: 76387075C90533AAD14E82A5D94E8486) |
| 29/11/2014 16:15:54.925 | Create Reg Key | %WINDOWS%\explorer.exe | HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ ebotigob=%WINDOWS%\whdhufel.exe |
| 29/11/2014 16:15:55.162 | Create File | %WINDOWS%\explorer.exe | %APPDATA%\Microsoft\Address Book |
| 29/11/2014 16:15:55.162 | Create Process | %WINDOWS%\explorer.exe | %SYSTEM%\vssadmin.exe |
| 29/11/2014 16:15:55.198 | Create File | %WINDOWS%\explorer.exe | %APPDATA%\Microsoft\Address Book\user.wab |
| 29/11/2014 16:15:55.508 | Terminate Process | %SYSTEM%\vssadmin.exe | - |
| 29/11/2014 16:15:59.066 | Create File | %WINDOWS%\explorer.exe | “02000000, 03000000, 04000000, 05000000 under %APPDATA%\ytivyteqyfypequs\” |
| 29/11/2014 16:15:59.466 | DNS Query | %WINDOWS%\explorer.exe | IT-NEWSBLOG.RU |
| 29/11/2014 4:15:59.786 | Query Directory | %WINDOWS%\explorer.exe | C:\* |
| 29/11/2014 4:15:59.900 | Create File | %WINDOWS%\explorer.exe | Start to encrypt all files located under C:\* |
| 29/11/2014 16:16:00.068 | Send Data | %WINDOWS%\explorer.exe | https://IT-NEWSBLOG.RU |
| 29/11/2014 16:16:01.951 | Send Data | %WINDOWS%\explorer.exe | https://IT-NEWSBLOG.RU |
| 29/11/2014 16:16:02.859 | Send Data | %WINDOWS%\explorer.exe | https://IT-NEWSBLOG.RU |
| 29/11/2014 16:17:23.564 | Create Process | %WINDOWS%\explorer.exe | C:\Program Files\Internet Explorer\iexplore.exe |
| 29/11/2014 16:17:24.863 | Create Process | C:\Program Files\Internet Explorer\iexplore.exe | C:\Program Files\Internet Explorer\iexplore.exe |
| 29/11/2014 16:17:26.933 | Send Data | %WINDOWS%\explorer.exe | https://IT-NEWSBLOG.RU |
| 29/11/2014 16:17:27.186 | Send Data | %WINDOWS%\explorer.exe | https://IT-NEWSBLOG.RU |
| 29/11/2014 16:17:27.186 | Send Data | %WINDOWS%\explorer.exe | https://IT-NEWSBLOG.RU |

to make them usable again. Since other system components of VirMon work independently, it has been sufficient to install the developed mini-filter driver into the analysis machine to integrate new OS. Accordingly, we successfully installed VirMon's mini-filter driver on Windows 10 without any need to modify or build driver's code. Table 2 shows the summarised run-time activities of the cyrptolocker still observed by VirMon during its analysis. Cyrptolocker malware sample (see [16]) was active in the wild and a signature has not yet been created at the time of writing this paper. This malware's activities show that it uses the process hollowing technique as in the previously analyzed sample (i.e., hesperbot sample). Then, it probably searches specific file types under C:\drive to encrypt and makes them unusable unless one does not have the description key. It sends some information to its C&C server and asks ransom from its victim user.

## 4 Conclusions

In this paper, the virtualization-based dynamic malware analysis system and its components is presented. A mini-filter driver is built to monitor run-time activities of the file to be analyzed. Since kernel-callback scheme is reliable and compatible with all versions of Windows OS, analysis machine can support all Windows OS versions. In addition to its portability feature, the design supports virtualization and scalability, for instance the average rate of analysis can be adjusted by the virtualization approach. Two recent malware samples captured in the wild are analyzed to illustrate the portability feature and analysis success of the presented dynamic analysis system. The activities of the analyzed samples are extracted accurately and details of each activity are given with the timestamp, event and process description, which enhances readability of the analysis.

**Acknowledgements.** The authors gratefully acknowledge the support of Galatasaray University, scientific research support program under grant #16.401.004.

## References

1. Internet Security Threat Report (2016). Symantec: https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf. Accessed 15 June 2016
2. Sukwong, O., Kim, H., Hoe, J.: Commercial antivirus software effectiveness: an empirical study. Computer **44**, 63–70 (2011)
3. Willems, C., Holz, T., Freiling, F.: Toward automated dynamic malware analysis using CWSandbox. IEEE Secur. Priv. **5**, 32–39 (2007)
4. Cuckoo Foundation, Cuckoo Sandbox. http://www.cuckoosandbox.org/. Accessed 1 June 2016
5. Seiferta, C., Steensona, R., Welcha, I., Komisarczuka, P., Endicott-Popovskyb, B.: A behavioral analysis tool for applications and documents. Digit. Invest. Int. J. Digit. Forensics Incident Response **4**, 23–30 (2007)

6. Tirli, H., Pektaş, A., Falcone, Y., Erdogan, N.: Virmon: a virtualization-based automated dynamic malware analysis system. In: The Proceedings of the 6th International Information Security & Cryptology Conference, Istanbul, Turkey, pp. 1–6 (2013)
7. Microsoft Corporation, Writing Preoperation and Postoperation Callback Routines. https://msdn.microsoft.com/windows/hardware/drivers/ifs/writing-preoperation-and-postoperation-callback-routines. Accessed 1 Mar 2013
8. Lazarevic, A., Kumar, V., Srivastava, J.: Intrusion detection: a survey. Massive Comput. **5**, 19–78 (2005)
9. Open Information Security Foundation, Suricata IDS. http://suricata-ids.org/. Accessed 1 Jan 2012
10. Bro Project, The Bro Network Security Monitor. http://www.bro.org/. Accessed 15 Feb 2014
11. Chen, B., Lee, J., Wu, A.S.: Active event correlation in Bro IDS to detect multi-stage attacks. In: The Fourth IEEE International Workshop on Information Assurance (2006)
12. Oracle, Oracle VM Virtual Box. https://www.virtualbox.org
13. Hesperbot malware sample: Google Corp., Antivirus scan results for 186c097b9d85b3501efcc4d8d374afe1. https://www.virustotal.com/en/file/a34f954ffb49f5c0b8f42376e062971284c9bec864e1d90a7e8d2910ae7c2077/analysis/
14. White, A.: Identifying the unknown in user space memory. Institute for Future Environments Science and Engineering, Faculty Queensland University of Technology, pp. 138–140 (2013)
15. Ligh, M.H., Adair, S., Hartstein, B., Richard, M.: Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code. Wiley Publishing Inc, Indianapolis (2011)
16. Cyrptolocker malware sample: Google Corp., Antivirus scan results for 76387075c90533aad14e82a5d94e8486. https://www.virustotal.com/en/file/09fe21dd9561603217cc8b419f01c7996b1440aa3e64967f136e38e7f306d625/analysis/

# Projection-Based Person Identification

Dora Neubrandt[1](✉) and Krisztian Buza[2]

[1] Department of Computer Science and Information Theory,
Budapest University of Technology and Economics, Budapest, Hungary
dori@biointelligence.hu

[2] Knowledge Discovery and Machine Learning,
Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany
chrisbuza@yahoo.com
http://www.biointelligence.hu/typing.html

**Abstract.** The increasing interest in person identification based on keystroke dynamics can be attributed to several factors. First of all, it is a cheap and widely applicable technique, whereas online services such as internet banking or online tax declaration require reliable person identification methods. Furthermore, there are various attack techniques against the existing identification methods, thus combining the existing methods with new person identification methods could improve the reliability of the identification. Recent research shows that person identification based on machine learning using keystroke dynamics data works surprisingly well. This is because the dynamics of typing is characteristic to users and a user is hardly able to mimic the dynamics of typing of another user. In this paper, we propose to use a projection-based classification technique for the task of person identification based on keystroke dynamics.

**Keywords:** Keystroke dynamics · Machine learning · Person identification · PROCESS

## 1 Introduction

Person identification became an important research topic as it may be used in various online services ranging from internet banking to online tax declarations. Conventional person identification methods use, for example, passwords or biometric features such as fingerprints or iris patterns. Despite the fact that solely password-based identification systems are very vulnerable to various threats, most online services' identification procedures rely on passwords. With the dynamics of typing we mean a time series, in which each of the values correspond to the duration of a keystroke. The dynamics of typing has been shown to be characteristic to individuals [1,2], and people are hardly able to mimic others' typing dynamics. Furthermore, person identification based on keystroke dynamics is cheap and it may be implemented with a standard keyboard (i.e., it does not necessarily require special hardware). Therefore, it is widely applicable and suitable for the task of online person identification. Our goal is to develop an accurate

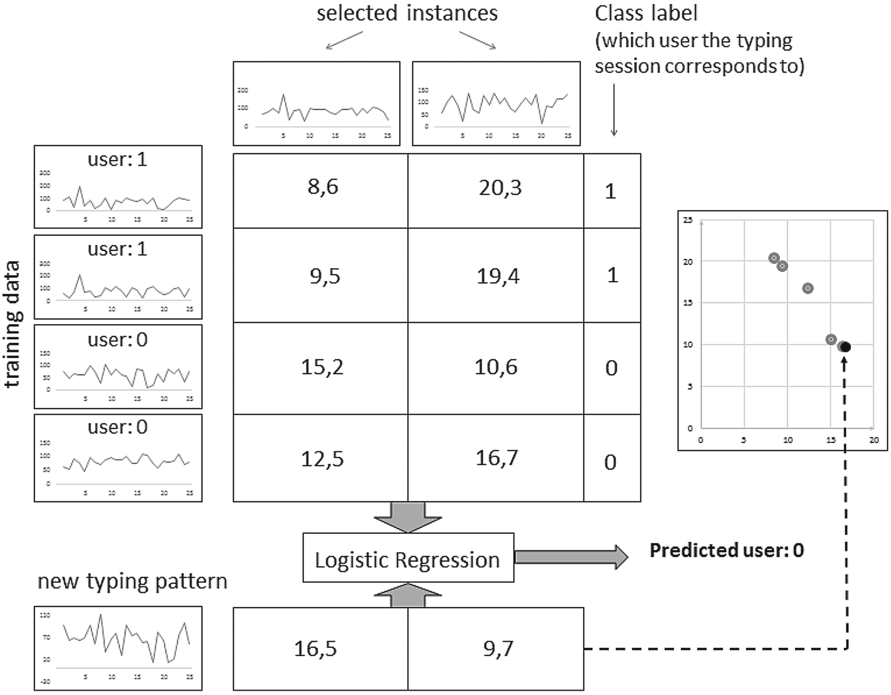
M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_23

keystroke dynamics-based person identification technique. We consider the task of *person identification based on keystroke dynamics* as a time-series classification problem for which various techniques have been developed recently[1], one of them is a projection-based approach, PROCESS [3], which has only been applied to the classification of electroencephalograph (EEG) signals previously. Despite the promising results, PROCESS has not been applied to person identification based on keystroke dynamics yet. This paper aims to close this gap. In particular, we propose to use a projection-based classification method for the task of person identification based on keystroke dynamics. We performed experiments on publicly available real-world typing dynamics data. Our results show that our approach outperformed other state-of-the-art classifiers. The remainder of the paper is organized as follows: Sect. 2 gives a brief overview of the most important related works. In Sect. 3 we describe the proposed projection-based classification of keystroke dynamics data. Section 4 presents our experimental results, while we draw conclusions in Sect. 5.

## 2 Related Work

Person identification based on keystroke dynamics can be considered as a classification task, for which various classifier may be applied, such as neural networks [4], data evolution methods [5] and classifiers based on Kolmogorov-Smirnov test [6]. While ensemble techniques [7,8] have been found to be powerful for various classification tasks, their recent applications include person identification based on keystroke dynamics data [9]. In this paper we consider the task of person identification based on keystroke dynamics as a time-series classification problem, for which nearest neighbor classifiers with dynamic time warping (DTW) distance [10] have been shown to be competitive [11,12] with various, more complex models such as neural networks [13], Hidden Markov Models [14] or "super-kernel fusion scheme". While these empirical results are supported by studies focusing on the theoretical aspects of classification [15,16], a few instances, called *bad hubs* have been shown to be responsible for a surprisingly large fraction of the classification error of the nearest neighbor classifier [17] which motivated the development of hubness-aware classifiers [18] and regressors [19] and their application to the person identification task [20]. On the other hand, in the case of electroencephalograph (EEG) signals, a projection-based classifier, PROCESS [3] outperformed several hubness-aware classifiers, such as $k$-Nearest-Neighbor with Hubness-aware Weighting (HW$k$NN), Naive Hubness Bayesian $k$-Nearest Neighbor (NHBNN) or Hubness Information $k$-Nearest Neighbor (HI$k$NN), both in terms of accuracy and computational time. Despite these promising results, none of the aforementioned works used projection-based classifiers, such as PROCESS, for the task of person identification based on keystroke dynamics previously. In this paper, we aim to close this gap by proposing to use a projection-based classifier for this task.

[1] See Sect. 2 for an overview of related works.



| | | |
|---|---|---|
| 8,6 | 20,3 | 1 |
| 9,5 | 19,4 | 1 |
| 15,2 | 10,6 | 0 |
| 12,5 | 16,7 | 0 |
| 16,5 | 9,7 | |

**Fig. 1.** Projection-based classification of keystroke dynamics data.

## 3 Projection-Based Classification of Keystroke Dynamics Data

As mentioned above, with keystroke dynamics data we mean time series, in which each of the values corresponds to the duration of a keystroke. Thus in our case the instances correspond to time series describing typing dynamics. In particular, each typing session corresponds to a time series. First, we project time series into a vector space. Subsequently, we train a logistic regression classifier using the projected instances which can be applied to classify new instances. This is illustrated in Fig. 1. Next we describe the details of this approach.

**Algorithm 1.** Projection of an instance

```
Require : instance $x$, set of reference instances $\mathcal{S}$
Ensure  : Projected instance $p$
for $i = 1 \dots d$ do
    $p[i] \leftarrow$ DTW distance of $x$ and the $j$-th instance of $\mathcal{S}$
end
```

**Algorithm 2.** Projection-based classification – Training

```
Require : Training data $\mathcal{D}$, integer $d$
Ensure  : Trained classifier
$\mathcal{S} \leftarrow$ select $d$ instances from $\mathcal{D}$ for $i = 1 \ldots |\mathcal{D}|$ do
    $\mathcal{P}[i] \leftarrow$ project the $i$-th instance of $\mathcal{D}$ with Alg. 1 using $\mathcal{S}$ as reference
      instances
end
$\mathcal{L} \leftarrow$ logistic regression classifier trained on $\mathcal{P}$
```

**Algorithm 3.** Classification of a new instance

```
Require : Selected instances $\mathcal{S}$, instance $x$ to be classified
Ensure  : Predicted class label of instance $x$
$p \leftarrow$ project the $x$ with Alg. 1 using $\mathcal{S}$ as reference instances $\mathcal{L}$.predict_label($p$)
```

First, we select a subset of the training instances. Then for each training instance, we calculate its distance from the selected instances. Thus we map the time series into a $d$-dimensional vector space where $d$ is the number of selected instances. For the calculation of distances, we use Dynamic Time Warping (DTW), see e.g. [18] for the detailed description of DTW. Next we train a logistic regression classifier using the projected training data. Algorithm 1 shows the pseudocode of projection, while Algorithm 2 summarizes the training procedure. If we want to classify a new time series, we project it into the same vector space by calculating its distance from the selected instances, and then we use the previously trained classifier (logistic regression) to predict the class label of the new instance, see Algorithm 3. We note that the standard logistic regression is a binary classifier, i.e., it assumes that only two class labels (usually coded as “0” and “1”) are present. However, it can be applied to multiclass classification tasks (such as person identification) if the multiclass task is reduced to a set of binary classification tasks. There are various techniques for that, such as one-vs-rest, one-vs-one or error correcting output coding, see e.g. [21] for details. In state-of-the-art implementations of logistic regression (such as the one in the *scikit-learn* machine learning library[2]) this issue is handled internally, i.e., the logistic regression classifier implements the interface of a multiclass classifier.

## 4 Experiments

In this section we describe the experimental evaluation of projection-based person identification. We collected time series describing keystroke dynamics, or *typing patterns* for short, from 12 different users over several months, resulting in a collection of 548 typing patterns in total. In each of the typing sessions, the users were asked to type the following short text based on the English Wikipedia page about Neil Armstrong: *That's one small step for a man, one giant leap for*

[2] http://scikit-learn.org.

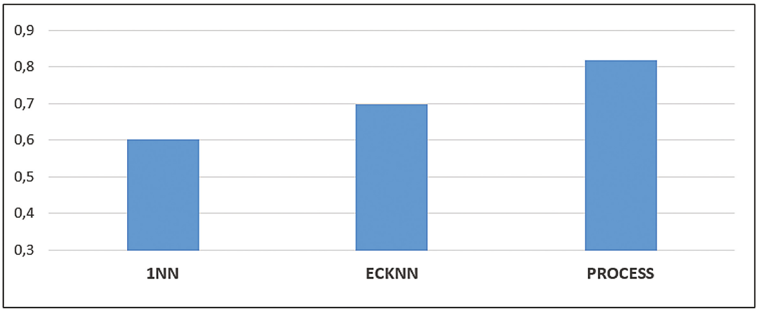
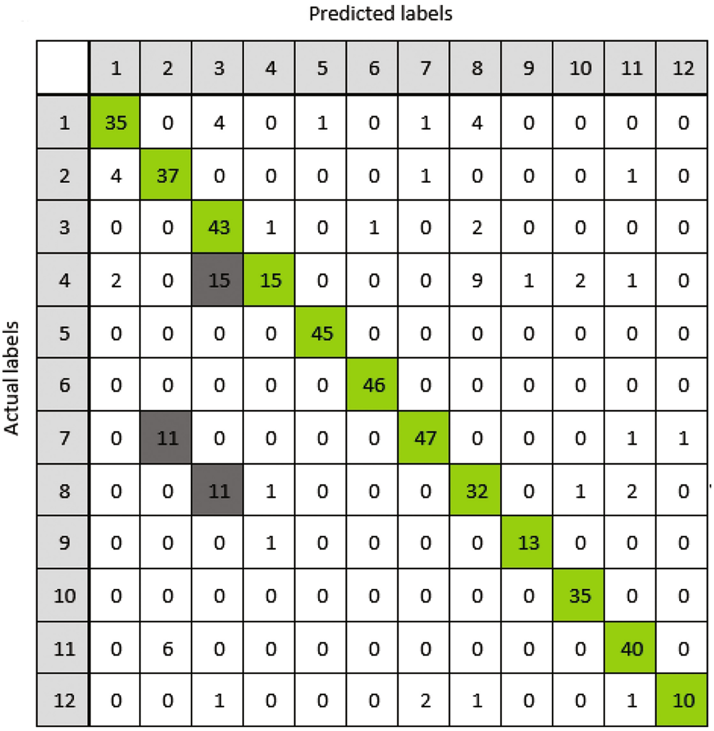


**Fig. 2.** Accuracy of person identification in case of the proposed projection-based approach, PROCESS, and the baselines.

*mankind. Armstrong prepared his famous epigram on his own. In a post-flight press conference, he said that he decided on the words just prior to leaving the lunar module.* In each typing session, we measured the duration of each keystroke, i.e., the time between pressing and releasing a key. Thus each typing pattern is a time series of keystroke duration values corresponding to a typing session. We used a self-made JavaScript application and a PHP script to capture the aforementioned time series and to save the data. Note that due to typing errors, the length of typing patterns may vary from session to session. For the evaluation of our approach, we considered the task of *person identification*, wherein a set of training time series and some "new" (i.e., test) time series are given, and the task is to decide, which user belongs to the "new" (i.e., test) time series.[3] From the collected data, we used the first 5 typing pattern from each user as training data, and the rest is used as test data. This corresponds to the assumption that the system is trained on a small set of typing patterns, such as the typing data recorded when the user registers to the system. We measured the quality of the models in terms of accuracy, i.e., the proportion of correctly classified typing patterns. The data, together with a web-based evaluation system, is publicly available at http://www.biointelligence.hu/typing-challenge/task2/. We implemented the projection-based approach in Python. We used all the training instances as "selected" instances, i.e., we calculated the distance of each training and test instance from all the training instances and mapped the data into a $12 \times 5 = 60$ dimensional vector space. After the projection, we trained a logistic regression classifier from the *sklearn* machine learning library on the projected training instances, and used this classifier to label the test instances. For the reasons mentioned in Sect. 2, we chose 1NN and $k$-Nearest-Neighbor with Error Correction (EC$k$NN) as baselines, see [20] for the detailed

[3] We note that the task of *person identification* is different from *person authentication*, in case of which the user claims an identify and the system has to decide whether the true identity matches the claimed identity. The task of person identification is inherently more challenging compared with the person authentication task, therefore, we decided to evaluate the proposed approach in context of person identification.


Predicted labels
Actual labels

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 35 | 0 | 4 | 0 | 1 | 0 | 1 | 4 | 0 | 0 | 0 | 0 |
| 2 | 4 | 37 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 43 | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 |
| 4 | 2 | 0 | 15 | 15 | 0 | 0 | 0 | 9 | 1 | 2 | 1 | 0 |
| 5 | 0 | 0 | 0 | 0 | 45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 46 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 11 | 0 | 0 | 0 | 0 | 47 | 0 | 0 | 0 | 1 | 1 |
| 8 | 0 | 0 | 11 | 1 | 0 | 0 | 0 | 32 | 0 | 1 | 2 | 0 |
| 9 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 13 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 35 | 0 | 0 |
| 11 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 40 | 0 |
| 12 | 0 | 0 | 1 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 1 | 10 |

**Fig. 3.** Confusion matrix of projection-based classification.

description of these models. As can be seen in Fig. 2, our results show that the proposed projection-based approach, PROCESS, outperformed both baselines (1NN and EC$k$NN). Figure 3 shows the confusion matrix of the projection-based classification. It can be seen that in the majority of the cases the predicted label equals to the actual label. However, we can notice that there are some users who have very similar typing patterns, so the predictions for their typing patterns may be confused – for example: user 3 and user 4.

### 4.1 Relevance to Real-World Applications

Based on the results of the above classification experiments, we aimed to approximate what additional security can be achieved using person identification based on keystroke dynamics in real-world applications. As an example, we consider credit card transactions on the internet. Usually, all the data required to perform a transaction is available on the bank card. We consider a hypothetical system which additionally checks the user's keystroke dynamics. There are two possible errors in such a system: (i) an illegitimate user is allowed to perform a transaction because his/her typing pattern is recognized erroneously as the typing pattern of the owner of the bank card, or (ii) a legitimate user is not allowed to perform a transaction because his/her typing pattern is not recognized properly. Under the assumption that the classification results are representative for the application, using *per-user* confusion matrices, we estimate the probability of a legitimate user not being allowed to perform a transaction to be approximately 20%. On the other hand, the system is expected to recognize 98% of the illegitimate transactions which may be considered as a reasonably high level of additional security for the case if the bank card gets lost or stolen.[4,5]

## 5 Conclusions and Outlook

In this paper we proposed to use a projection-based approach for the task of person identification based on keystroke dynamics. We evaluated our model on publicly available real-world typing data, and compared it with state-of-the-art classifiers. Our results show that our approach outperforms the baselines on the task of person identification and it may provide additional security in applications.

**Acknowledgment.** D. Neubrandt was supported by the "Új Nemzeti Kiválóság Program" (ÚNKP-16-1-1) of the "Emberi Erőforrások Minisztériuma".

## References

1. Antal, M., Szabó, L.Z., László, I.: Keystroke dynamics on android platform. Procedia Technol. **19**, 820–826 (2015)
2. Monrose, F., Rubin, A.D.: Keystroke dynamics as a biometric for authentication. Future Gener. Comput. Syst. **16**(4), 351–359 (2000)
3. Buza, K., Koller, J., Marussy, K.: PROCESS: projection-based classification of electroencephalograph signals. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2015. LNCS, vol. 9120, pp. 91–100. Springer, Cham (2015). doi:10.1007/978-3-319-19369-4_9
4. Wong, F., Supian, A.S.M., Ismail, A.F., Kin, L.W., Soon, O.C.: Enhanced user authentication through typing biometrics with artificial neural networks and k-nearest neighbor algorithm. In: 35th IEEE Asilomar Conference on Signals, Systems and Computers, vol. 2, pp. 911–915 (2001)

[4] We note that they were calculated under the assumption that the observations in the classification experiment are representative to real-world application scenarios. This includes (but it is not limited to) the assumption of a *naive attacker*. That is, we did *not* assume an "intelligent" attacker who would try to record and/or imitate the dynamics of the legitimate user. Instead, a *naive attacker* was assumed who simply steals a bank card (or the information printed on the card) and tries to use it for internet-based transactions without paying attention to imitate the owner's dynamics of typing.

[5] We also note that there is a trade-off between the aforementioned two types of error and, if required, recognition systems may be tuned in order to decrease one of them, while the other type of error may increase. Though in principle, such tuning is possible in case of projection-based classification as well (for example, based on the continuous output of logistic regression), this is left for future work.

5. Kozierkiewicz-Hetmanska, A., Marciniak, A., Pietranik, M.: Data evolution method in the procedure of user authentication using keystroke dynamics. In: Nguyen, N.-T., Manolopoulos, Y., Iliadis, L., Trawiński, B. (eds.) ICCCI 2016. LNCS, vol. 9875, pp. 379–387. Springer, Cham (2016). doi:10.1007/978-3-319-45243-2_35
6. Ceffer, A., Levendovszky, J.: Kolmogorov-Smirnov test for keystroke dynamics based user authentication. In: 17th IEEE International Symposium on Computational Intelligence and Informatics (2016). doi:10.1109/CINTI.2016.7846387
7. Wozniak, M., Jackowski, K.: Fusers based on classifier response and discriminant function – comparative study. In: Corchado, E., Abraham, A., Pedrycz, W. (eds.) HAIS 2008. LNCS, vol. 5271, pp. 361–368. Springer, Heidelberg (2008). doi:10.1007/978-3-540-87656-4_45
8. Kurzynski, M., Wozniak, M.: Combining classifiers under probabilistic models: experimental comparative analysis of methods. Expert Syst. **29**(4), 374–393 (2012)
9. Doroz, R., Porwik, P., Safaverdi, H.: The new multilayer ensemble classifier for verifying users based on keystroke dynamics. In: Núñez, M., Nguyen, N.T., Camacho, D., Trawiński, B. (eds.) ICCCI 2015. LNCS, vol. 9330, pp. 598–605. Springer, Cham (2015). doi:10.1007/978-3-319-24306-1_58
10. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech Sig. Process. **26**(1), 43–49 (1978)
11. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. In: Proceedings of the 23rd ACM International Conference on Machine Learning, pp. 1033–1040 (2006)
12. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. Proc. VLDB Endowment **1**(2), 1542–1552 (2008)
13. Nanopoulos, A., Alcock, R., Manolopoulos, Y.: Feature-based classification of time-series data. Int. J. Comput. Res. **10**(3), 49–61 (2001)
14. Kim, S., Smyth, P., Luther, S.: Modeling waveform shapes with random effects segmental Hidden Markov Models. In: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, pp. 309–316 (2004)
15. Chen, G.H., Nikolov, S., Shah, D.: A latent source model for nonparametric time series classification. Adv. Neural Inf. Process. Syst. **26**, 1088–1096 (2013)
16. Devroye, L., Györfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer, New York (1996)
17. Radovanović, M., Nanopoulos, A., Ivanović, M.: Hubs in space: popular nearest neighbors in high-dimensional data. J. Mach. Learn. Res. **11**, 2487–2531 (2010)
18. Nenad, T., Buza, K., Marussy, K., Kis, P.B.: Hubness-aware classification, instance selection and feature construction: survey and extensions to time-series. In: Feature Selection for Data and Pattern Recognition, pp. 231–262 (2015)
19. Buza, K., Nanopoulos, A., Nagy, G.: Nearest neighbor regression in the presence of bad hubs. Knowl. Based Syst. **86**, 250–260 (2015)
20. Buza, K.: Person identification based on keystroke dynamics: demo and open challenge. In: CAiSE Forum 2016, 28th International Conference on Advanced Information Systems Engineering (2016)
21. Witten, I.H., Frank, E., Hall, M., Pal, C.: Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, Burlington (2016)

# 3-Steps Keyboard: Reduced Interaction Interface for Touchless Typing with Head Movements

Adam Nowosielski(✉)

Faculty of Computer Science and Information Technology,
West Pomeranian University of Technology, Szczecin,
Żołnierska 52, 71-210 Szczecin, Poland
anowosielski@wi.zut.edu.pl

**Abstract.** This paper introduces a novel technique for touchless typing with head movements allowing to reach any alphabet character in only three steps. Head movements are frequently used for human-computer interaction by users with motor impairments unable to operate standard computer input devices. In such interfaces great difficulty is typing. Many directional head movements are required to reach subsequent characters using the on-screen keyboard and additional mechanism (like eye blink or mouth open) supplements the selection process. In this paper, a reduced interaction keyboard for touchless typing with head movements is proposed. The solution is based on recognition of head movements in four main directions.

**Keywords:** Touchless typing · Typing with head movements · Virtual keyboard · Gesture interaction · Human-computer interaction

## 1 Introduction

Researchers are actively engaged in the development of touchless interaction. Contemporary research on no-contact interfaces employs computer vision and pattern recognition approaches. Most solutions offer hands-free control of electronic devices using gestures. However, natural user interfaces are frequently of no use for disabled users with motor impairments. Those people need special interfaces to increase their independence in their daily lives. New assistive technologies can improve their participation in social activities and provide means for information access. For some people such interfaces enable computer-mediated communication with others. The control and input devices are of great importance for physically challenged people. Head operated interface is one of the option. It offers touchless interaction through movements of the user's head. Many approaches focus directly on a face or facial features and use the changes in their appearance for steering purposes. Most head operated interfaces focus on conventional mouse replacement and the pointer manipulation in the Graphical User Interface (GUI). The term *camera mouse* is frequently used for solutions of

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_24

these kind (e.g. [1,2]). Text entry is usually performed on the standard on-screen QWERTY keyboard through pointer manipulation. The process is tedious and last long. There are also several alternative techniques for touchless interfaces. Most popular of these include: speech recognition, eye tracking and brain computer interfaces. Some of them can not be applicable for some people due to their disabilities. The price level of other is far too high for many individuals. The computer vision approaches may provide an excellent alternative here. The rest of the paper is structured as follows. In Sect. 2 related works are outlined and discussed. Section 3 presents concepts applied in keyboards with reduced number of keys. In Sect. 4 reduced interaction keyboard is proposed and its handling with head movements for touchless typing is described in Sect. 5. The details of computer vision methods employed are provided in Sect. 6. Final conclusions and a summary are provided in Sect. 7.

## 2 Related Works

The operation principle of head operated interfaces is based on the detection and tracking of the user's face or facial features. Former solutions relied on markers attached to distinctive parts of the head (e.g. middle of the forehead). Their aim was to simplify the processes of detection and tracking. Modern solutions, however, are based on computer vision and pattern recognition approaches. Captured head (or face features) movements are translated into the motion of a pointer on a display presenting GUI. In [1] facial movements has been divided into two groups: rigid and non-rigid motions. The first denotes rotation and translation while the second includes actions like opening, closing, and stretching of the mouth, eye winks, cheeks twitch, etc. There are different approaches to handle pointer manipulation in GUI using head movements. In [3] the position of cursor is controlled by changes of the position of user's nostrils related to the face region. Similar approach is proposed in [4]. Here, the nose tracking is supported with eye regions which are processed for recognition of winks. Those are then interpreted as the mouse clicks. The involuntary blink and steering blink are distinguished on the basis of the closure duration. Eye blink is also used for confirmations in [5]. The mouse control is achieved here using the image plane position of the eyes. Mouse movements controlled by the user's eye movements is also proposed in [6]. Clicking events, on the other hand, are implemented through mouth shape changes (opening/closing). Detection of mouth motions applied for mouse events is also employed in [1]. The mouse cursor navigation in [1] is obtained by 3D head pose. 3D pose estimation for camera mouse is also utilized in [2]. Interestingly, the clicking events are obtained here through the distance between the camera and user head. Another imaging technique - a depth imaging - is adapted for control and navigate the cursor in [7]. From a depth image the nose position and the mouth status are detected and used for steering. The presented interfaces operated by head and face features focused on pointing and conventional mouse replacement in non-contact environment. The text typing in such interfaces is usually offered through the on-screen keyboard operated by pointing mechanism. [6] provided for example a “spelling board” which substitute a standard keyboard. It is operated with eye movements and mouth shapes. Touchless typing with head movements is also proposed in [8]. Here, the head movements are supported with three face gestures chosen for a key selection: mouth open, brows up and brows down. No additional gestures are required in head typing interface proposed in [9]. Only four directional head movements suffice to write. It is noteworthy that some finished products are released as open source. One of the most interesting project, recently shared publicly, is the Assistive Context-Aware Toolkit (ACAT) [10] originally created for Professor Stephen Hawking. It offers tools for wide applications and beside typing it enables managing documents or navigating the Internet. It can be operable with eye blinks, eyebrows movements, cheeks twitch or mouth open. A typical head operated interfaces available as open source projects are Enable Viacam (eViacam) [11] and QVirtboard [12]. Both offers mouse replacement through face movements. The QVirtboard also offers a dedicated keyboard and supports other modes of control (eye tracking or hand movements) [12,13]. Analysis of the existing approaches to text typing by head movements leads to the conclusion that this form of typing is time-consuming and usually requires many steps to reach the intended letter. Since this form of interface for some people is the only solution to information access or to computer-mediated communication, improvements or new ideas are desirable.

## 3 Keyboards with Reduced Number of Keys

The QWERTY keyboard still used today has been designed in the 19th century. More efficient and ergonomic layouts have been introduced, unfortunately without a great success. The QWERTY key arrangement keyboard still remains the standard - the standard which many argue is not suited for the current needs and interfaces. The widespread use of mobile devices forced the research for new solutions. The current trend is reduction in the number of keys and a change in the form of interaction with for example swipe gestures (employed also in the QWERTY keyboard). Figure 1 presents examples of reduced keyboards. The first example depicts well known and somewhat obsolescent phone numeric keypad. Each numeric key is associated with several alphabet characters and the typing proceeds with multitap or T9 technique. In the fist case, the user is required to press the key repeatedly as many times as the position of the letter is on the key (one to four keypresses). The second solution is dictionary based and allows a single keypress. From a pressing sequence possible words are matched and displayed for the user who selects the intended one.

A completely different approach is offered by the 8pen keyboard designed for touchscreens (http://www.8pen.com/). The keyboard is divided into four directional sectors (top, down, left or right) with specifically arranged letters on the borders. The text entry is stroke-based and starts from the central part of the keyboard. Movement to the directional sector indicates potential letters and the further circular movement through remaining sectors determines consecutive characters (the number of sectors crossed corresponds to the position

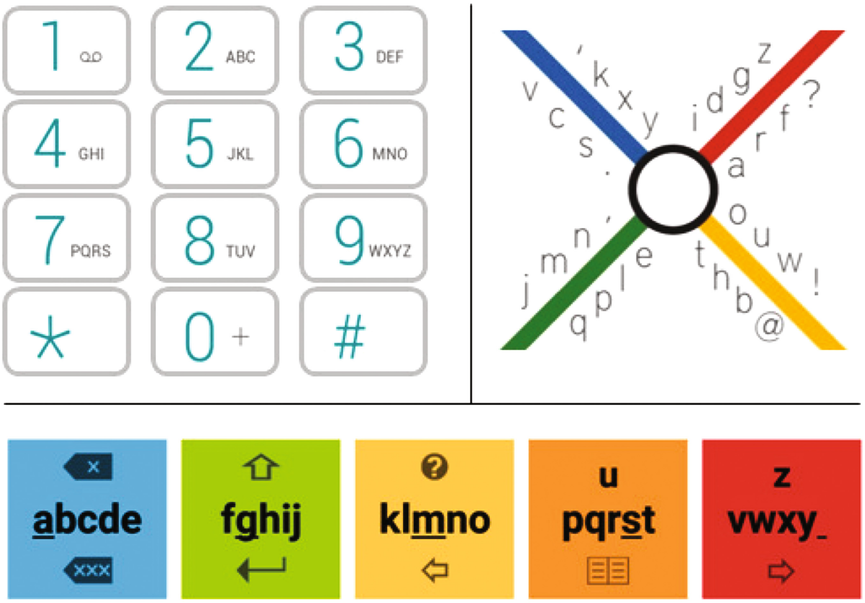


**Fig. 1.** Examples of the reduced keyboards: phone keypad (upper left), 8pen (upper right), 5-Tiles (bottom)

of the letter on the border). The process of letter selection ends with the return to the center. The 5-Tiles (http://fivetiles.com/) is another example of reduced keyboard. It offers 5 keys (called tiles) and the interaction is through keystrokes and swipe gestures simultaneously. A single tap on one of the five keys is responsible for the selection of the underlined character. Other letters are obtained with the swipe gesture which starts on the appropriate key (containing the intended character) and continues through adjacent tiles. Similarly as in the 8pen solution, the number of tiles crossed corresponds to the position of the letter on the key. The keyboards presented above are only examples. More solutions are available. Their common feature is reduction the number of buttons and shortening the path distances. New alternatives to QWERTY key arrangement can be a source of inspiration for head typing interface. However, there are many difficulties in the adaptation. Interactions like swipe gestures, circular movements or even key pressing present a challenge.

## 4 Proposition of the Layout

The most advantageous feature of the QWERTY keyboard is its ubiquity and known layout. Many approaches to change this standard resulted in a failure. Users are reluctant to learn new key arrangements. It seems that the only widely acceptable alternative to the QWERTY layout is the alphabetical order of keys.

The alphabetical order is intuitive and easily operable. It is present in (already mentioned in the previous section) the 12-keys mobile phone keypad or 5-Tiles keyboard. It was the standard for 5-keys pagers or now is willingly used in smart TVs. In our earlier approaches to touchless typing with head movements a single row alphabetical keyboard has been used [9]. The interface was operated with only directional head movements. Movements to the left and to the right were transformed to shifts of the active letter on the on-screen keyboard. Tilt of the head down (the nod gesture) was interpreted as pressing or affirmation of dictionary suggested word. The opposite upward gesture was used for backspace or leaving the dictionary suggestions row. The main advantage of the proposed interface was the elimination of the need for additional gestures like eye blink or mouth opening. The main drawback of the single row alphabetical keyboard is the frequent need to move through many characters (e.g. between ‘a’ to ‘l’ there are 10 other letters which have to be passed through). To reduce the number of passes the keyboard may adapt and deactivate letters which at a given stage do not allow the creation of proper dictionary words. Such approach is widely used in many applications where text entry is not comfortable enough (for example in GPS devices when typing the name of desired town). The technique of reducing the set of potential characters have been employed in single row alphabetical keyboard operated with head movements giving over 15% increase in typing speed [14]. Here, new concept to touchless text typing is proposed. First, in the Fig. 2 the keyboard layout is presented. It has the form of a single row with alphabeticaly arranged characters. The letters, however, are vertically translated in relation to each other and form distinctive groups. This is intentional and guides the appropriate interaction. The most important is that each letter can be entered with only three steps. The need to move through many characters has been completely eliminated.

Beside the alphabetic characters the keyboard offers access to other symbols or control keys. The extreme buttons represents backspace and space. Dot, comma and enter are also available for direct selection. Numbers and other symbols are accessed by switching the keyboard state (accomplished by the second key). The presented arrangement of keys is only a proposal and may be modified by the user. It is the 3-steps interaction which is crucial.

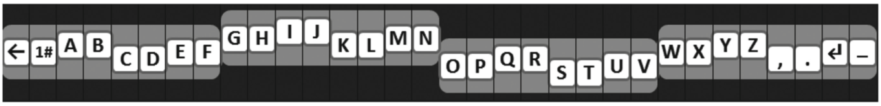


**Fig. 2.** Layout of the reduced interaction keyboard

## 5 Interaction with 3-Steps Keyboard

Each alphabetic character is assigned to its own key in the reduced interaction keyboard proposed in Sect. 4. They are arranged in four main groups. The first and fourth group are located on the same level. The two middle groups are shifted - one upward and the other downward. The mutual displacement indicates which direction has to be selected, in the first step, to reach each group. Thus, the first group is selected with the left direction movement, the second - with the up movement, the third - with the down movement, and the fourth - with the movement to the right. With the choice of individual group the others are deactivated. In the second step, within the already selected group, a subgroup (consisting of two letters) is selected using the same procedure. Subgroups of two letters are translated in the same manner as main groups relative to each other. This way, with a directional movement a pair of letters is selected (the remaining pairs are deactivated). The final step consists of the selection of appropriate letter from the pair with left or right direction movement. The selected letter is transcribed and the keyboard returns to the initial appearance (all the keys are activated again). The example is presented in Fig. 3 where the 'a' character is typed with three movements (left, up, left).

The interaction proposal enables access to any letter in only three steps. There is no need for long passing through many letters to reach the one intended.

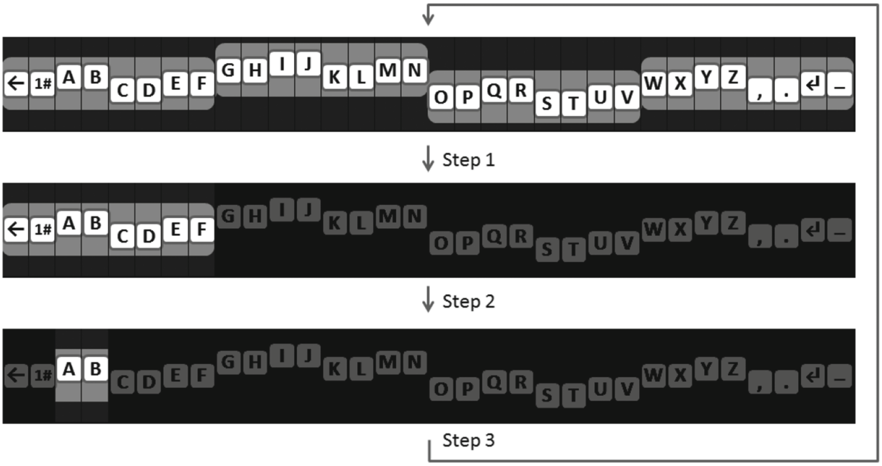

← 1# A B C D E F G H I J K L M N O P Q R S T U V W X Y Z , . ↵ _
Step 1
Step 2
Step 3

**Fig. 3.** Exemplary interaction: letter 'a' transcription with the left, up and left combination of movements

## 6 The Control Routine

To operate the proposed reduced interaction keyboard the four directional head movements have to be recognized. It is assumed that the user looks at the screen and the camera above records his actions. Captured head movements (i.e. motion parameters) are converted into the interaction with the keyboard. The typing starts with the head movement in one of the four directions (triggered when the motion parameters exceed specified thresholds). The up and down directions are achieved with the upward or downward nod gesture. Those rotary gestures are natural and very easy to make by the user. Figure 4 presents a person engaged in the process of performing up and down nods. Black lines indicate the position of the reference center. This center is dynamically adapted with new letter typed and periodically.

Left and right directions are accomplished with head tilt, shift or rotation. It is the user who chooses the preferred form of interaction. The fastest and easiest to perform are rotary movements. Users, however, are forced to look at the screen at an angle which might be uncomfortable to some people. Horizontal head movements are the hardest to perform and the tilt in this group is best balanced. A person practicing different ways of performing left and right gestures is presented in Fig. 5. It is important to notice that during horizontal head movements with the tilt approach, the change in vertical position of the head may also occur. As a remedy a locking procedure (described later) is executed.

The recognition of the user actions is based on the detection and tracking of the face. The procedure is as follows. First, well known Viola and Jones approach [15] is used for the initial face detection. The initial head centre, defined as the centre of the returned bounding box, is stored as the reference point. After the detection the face is tracked with the Kanade-Lucas-Tomasi (KLT) feature-tracking procedure. Corners, calculated for the face region using the minimum eigenvalue algorithm, are used for tracking. Example collections of points are depicted in Figs. 4 and 5. For each point the tracker attempts to find the corresponding point in the new frame. Then, the geometric transformation based on matched pairs is calculated. Outliers are excluded with the MSAC algorithm [16]. Finally, the calculated transform is applied to the bounding box of the previous face localization. Changes in the localization of head centre serve for steering purposes - detection of the head directional movements. Coordinates of the initial head centre are updated with small, natural user movements (adaptation). Movements exceeding the thresholds are used for steering. The motion detection in a given direction automatically locks other directions. This procedure prevents the unexpected behaviour and accidental transitions particularly in the case of tilt operation used for horizontal directions. Finally, after the movement in any direction the interface expects the return of the head to the centre. The referred approach offers stable and reliable head position for the real-time interaction.

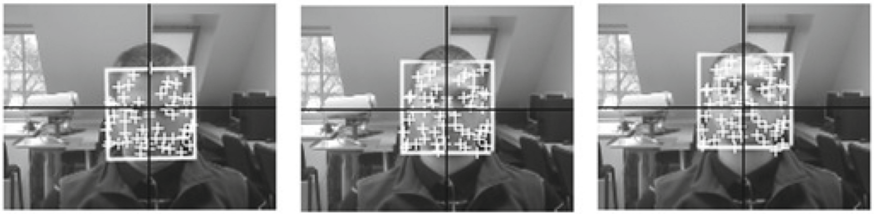


**Fig. 4.** Downward nod, neutral, and upward nod gestures performed by the user

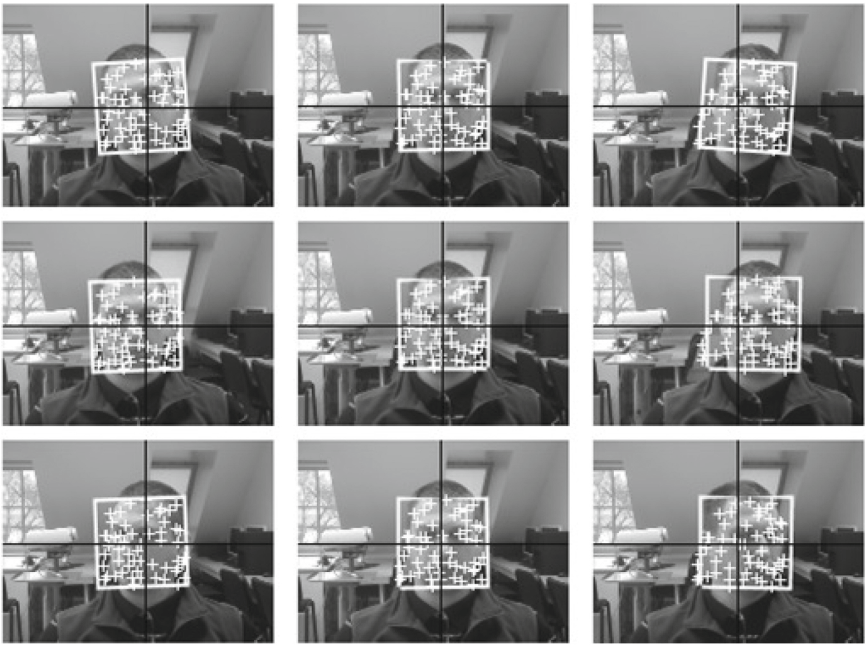


**Fig. 5.** Left and right directions performed with: tilt movement (top), shift movement (middle), and rotary movement (bottom)

To prevent unexpected errors, avoid the problem of accumulative error during tracking, and for calibration purposes, after new letter is typed the face detection and tracking procedure is restarted and new reference centre is calculated. The process is imperceptible by the user. When user remains in neutral position the centre is continuously adapting to involuntary head movements.

## 7 Conclusions

In the paper, the problem of touchless text input with head movements has been addressed. The solution proposed is innovative in context of letters selection. In the existing approaches of head operated interfaces user is expected to aim at the intended letter with an uncomfortable pointing procedure or to traverse through many letters to reach the one intended. Here, each letter can be entered with only three steps where the individual step consists of movement of the head in one of the four directions and the return. The proposed 3-Steps Keyboard consist of alphabetically arranged characters. All the letters form a single row but are vertically translated in relation to each other. This is intentional and guides the appropriate interaction. The main purpose of creating a keyboard was to reduce the number of steps required for typing with head movements. Such interfaces are of particular importance for physically challenged people who are unable to operate the standard computer input devices. These computer users may also have difficulties with head movements, therefore reducing the number of required moves is particularly important. The 3-Steps Keyboard has been well received by volunteers participating in initial experiments. They pointed out that the method of letters selection is uncommon but easy. It requires, however, some familiarity. New users, after a brief acquaintance were able to write at a rate of 12–14 cpm. More thorough experiments are planed in the future with the emphasis on learning ability. Other, beside the head, control mechanisms are also planed to be adapted and assessed with the 3-Steps Keyboard.

## References

1. Tu, J., Tao, H., Huang, T.: Face as mouse through visual face tracking. Comput. Vis. Image Underst. **108**(2007), 35–40 (2007)
2. Nabati, M., Behrad, A.: 3D Head pose estimation and camera mouse implementation using a monocular video camera. Sign. Image Video Process. **9**(1), 39–44 (2015)
3. Morris, T., Chauhan, V.: Facial feature tracking for cursor control. J. Netw. Comput. Appl. **29**(2006), 62–80 (2006)
4. Varona, J., Manresa-Yee, C., Perales, F.J.: Hands-free vision-based interface for computer accessibility. J. Netw. Comput. Appl. **31**(4), 357–374 (2008)
5. Santis, A., Iacoviello, D.: Robust real time eye tracking for computer interface for disabled people. Comput. Methods Program. Biomed. **96**(1), 1–11 (2009)
6. Shin, Y., Ju, J.S., Kim, E.Y.: Welfare interface implementation using multiple facial features tracking for the disabled people. Pattern Recogn. Lett. **29**(2008), 1784–1796 (2008)
7. Bian, Z.-P., Hou, J., Chau, L.-P., Magnenat-Thalmann, N.: Facial position and expression-based human–computer interface for persons with tetraplegia. IEEE J. Biomed. Health Inf. **20**(3), 915–924 (2016)
8. Gizatdinova, Y., Spakov, O., Surakka, V.: Face typing: vision-based perceptual interface for hands-free text entry with a scrollable virtual keyboard. In: IEEE Workshop on Applications of Computer Vision, Breckenridge, CO, USA, pp. 81–87 (2012)
9. Nowosielski, A.: Minimal interaction touchless text input with head movements and stereo vision. In: Chmielewski, L.J., Datta, A., Kozera, R., Wojciechowski, K. (eds.) ICCVG 2016. LNCS, vol. 9972, pp. 233–243. Springer, Cham (2016). doi:10.1007/978-3-319-46418-3_21
10. Assistive Context-Aware Toolkit (ACAT). Project page (2017). https://01.org/acat
11. Enable Viacam (eViacam) (2017). http://eviacam.sourceforge.net
12. QVirtboard. Project page (2017). http://qvirtboard.sourceforge.net
13. Nowosielski, A., Chodyła, Ł.: Touchless input interface for disabled. In: Burduk, R., Jackowski, K., Kurzynski, M., Wozniak, M., Zolnierek, A. (eds.) Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013. AISC, vol. 226, pp. 701–709. Springer, Heidelberg (2013)
14. Nowosielski, A.: Acceleration of touchless typing with head movements by limiting the set of potential characters. Przegl. Elektrotechniczny **2016**(12), 241–244 (2016). doi:10.15199/48.2016.12.62
15. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision **57**(2), 137–154 (2004)
16. Torr, P.H.S., Zisserman, A.: MLESAC: a new robust estimator with application to estimating image geometry. Comput. Vis. Image Underst. **78**(1), 138–156 (2000)

# An Algorithm for Selective Preprocessing of Multi-class Imbalanced Data

Szymon Wojciechowski, Szymon Wilk$^{(\boxtimes)}$, and Jerzy Stefanowski

Institute of Computing Science, Piotrowo 2, 60-965 Poznan, Poland
szymon.wilk@cs.put.poznan.pl

**Abstract.** In this paper we propose a new algorithm called SPIDER3 for selective preprocessing of multi-class imbalanced data sets. While it borrows selected ideas (i.e., combination of relabeling and local resampling) from its predecessor – SPIDER2, it introduces several important extensions. Unlike SPIDER2, it is able to handle directly multi-class problems. Moreover, it considers the relevance of specific decision classes to control the order of their processing. Finally, it uses information about relations between specific classes (modeled with misclassification costs) to better control the extent of changes introduced locally to preprocessed data. We performed a computational experiment on artificial 3-class data sets to evaluate and compare SPIDER3 to SPIDER2 with temporarily aggregated classes and the results confirmed advantages of the new algorithm.

## 1 Introduction

Class imbalance is associated with classification problems and manifests itself as uneven distribution of examples across decision classes, where one or more classes (*minority classes*) are heavily underrepresented in comparison to the other ones (*majority classes*). It is often accompanied by additional data difficulty factors, such as rare sub-concepts in the minority classes, overlapping regions between the minority and majority classes, or examples from the minority classes located inside the majority classes (see a discussion in [6]). A combination of class imbalance and additional difficulty factors poses a significant challenge for well-known learning methods and can seriously deteriorate classification performance, especially for the minority classes [6]. For the sake of simplicity, in the subsequent text we will refer all these combined factors as to class imbalance.

There is a growing research interest in class imbalance driven by its practical and theoretical importance. While many methods have been already proposed (see [3] for a review), there are still relevant challenges that need to be addressed [4]. One of them is an ability to deal with multiple decision classes. Current research focuses mostly on class imbalance in binary classification [4]. This may be attributed to the nature of typically considered problems, where a single minority class is so important from the domain perspective that all effort is aimed at improving its correct recognition and the remaining classes can be collapsed (aggregated) prior to processing. However, there are problems

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_25

with several important minority classes and their aggregation is questionable or inappropriate as it discards potential relations between these classes. To address this issue several methods based on the *one-versus-one* (OVO) or *one-versus-all* (OVA) schemes have been applied [2]. These schemes iterate over all possible pairs of classes or aggregations of classes (e.g., one class versus all remaining ones) and in each iteration apply methods developed for binary problems. The OVA scheme, although preserves original classes, still discards relations between them. OVO does not aggregate classes, however, it is more complex than OVA, as it needs to process all possible pairs of classes.

In this work we go further and propose an algorithm for direct selective preprocessing of multi-class imbalanced data. Unlike methods based on OVO or OVA, it does not introduce additional complexity and it is able to use information about local relations between specific classes. For the sake of simplicity and intuitive presentation in this paper we focus on three decision classes – in addition to the minority and majority ones we consider an *intermediate* class that in terms of its relevance (and usually the number of examples) is located in-between. However, the algorithm is also applicable to more complex problems with larger numbers of classes. An introduction of this intermediate class has been inspired and motivated by our experience with clinical decision problems, where often patients are classified as very sick, moderately sick and healthy. Then, it is questionable to collapse very sick and moderately sick patients, as the latter should not be given therapies that are often aggressive and expensive. Moreover, moderately sick patients should not be combined with healthy ones to ensure the former are given treatment they require.

Relations between classes similar to the ones described above are considered within cost-sensitive learning [1]. However, that approach is aimed at constructing classifiers that are aware of misclassification costs and try to minimize the total cost, while we focus on the preprocessing phase and use the relations to control modifications introduced to data, so that no additional requirements are imposed on learning algorithms applied later.

The algorithm proposed in this paper is based on SPIDER2 [7] – our technique for selective preprocessing of imbalanced data. SPIDER2 is aimed at problems with two classes and it combines local undersampling and relabeling of examples from the majority class with oversampling (amplification) of examples from the minority class. These modifications are controlled by local characteristics of processed examples. Unlike other techniques, in particular SMOTE and its variants [5], it does not create any artificial examples and it introduces less extensive changes to the distribution of classes. Due to these properties, SPIDER2 is well suited to practical problems, including clinical ones.

The new SPIDER3 algorithm employs the same techniques of selective preprocessing (relabeling and local under- and oversampling) as its predecessor. It uses information about absolute relevance of decision classes captured by their ordering and about local relations between pairs of classes modeled by misclassification costs. Since these costs may be not available or difficult to establish for certain problems [2], they are not strictly required – in such case SPIDER3 uses default costs that correspond to a 0–1 loss function. The algorithm processes classes in the order of their increasing relevance. At each step it controls the extent of modifications to limit the impact on other (more relevant) classes and to account for relations between classes. In the paper we also describe a computational experiment where SPIDER3 has been evaluated on a set of artificial data sets and compared to a modified variant of SPIDER2 with temporary aggregation of decision classes. Obtained results show advantages of the new proposal.

Summarizing, contribution of this work is three-fold: (1) it proposes to consider both absolute relevance of decision classes and local relations between pairs of classes modeled with misclassification costs (this distinguishes our approach from other preprocessing techniques where costs are derived from imbalance ratios [2]), (2) it introduces a new algorithm for selective preprocessing of multi-class imbalanced data, and (3) it provides its experimental evaluation where it favorably compares to its earlier version.

## 2 Related Work

Research on multi-class imbalanced classification problems is still in its early stages in comparison to what has been already proposed for binary problems [4]. It focuses on three major themes: (1) multi-class decomposition, (2) multi-class preprocessing and (3) multi-class classifiers (see [4] for a more detailed discussion).

The first theme is concerned with techniques that decompose original multi-class problems into a set of binary problems which may be handled using existing and mature methods. Proposed techniques range from relatively simple OVO and OVA (discussed in Sect. 1) to more complex hierarchical decompositions. Research related to the second theme aims at developing new preprocessing methods employing advanced sampling that combines several resampling methods. Finally, the third theme covers the research on methods to construct classifiers directly from imbalanced data. Already proposed solutions use modified distance measures (e.g., Hellinger distance) or ensembles of diversified (in terms of competencies) classifiers.

An important challenge that appears in all three themes discussed above is an identification and analysis of difficulty factors in multi-class data sets (e.g., overlapping and noisy examples) and development of taxonomies that capture the difficulty of processed examples. This line of research is well demonstrated in [6] where the authors proposed a taxonomy introducing safe, borderline, rare and outlier examples (we rely on this taxonomy when generating artificial data sets – see Sect. 4).

Our research fits the second theme, i.e., multi-class processing. According to [8], there are only a few specialized methods, mostly SMOTE-derivatives, e.g., Static-SMOTE. A more typical approach is to apply decomposition first and then to use binary preprocessing – see an extensive experimental study in [8]. Its results also demonstrate that introducing misclassification costs at this stage is beneficial for the accuracy attained by constructed classifiers. This provides a sound justification and motivation for our research and the SPIDER3 algorithm.

## 3 SPIDER3 Algorithm

SPIDER3, similarly to its predecessor, modifies a processed data set by relabeling selected examples, cleaning their neighborhood (local undersampling) or amplifying them (local oversampling). To control these modifications, the algorithm checks $k$-nearest neighborhood of a considered example (i.e., a *seed*) identified with heterogeneous value difference metric (HVDM) [7]. In addition to the distribution of classes in this local neighborhood, it also considers relations between each of these classes and the class of the seed.

As already explained, relations between pairs of classes are modeled as misclassification costs organized into a cost matrix. An example of such matrix is given in Table 1 (we use $c_{min}$, $c_{int}$ and $c_{maj}$ symbols to denote the minority, intermediate and majority class respectively). We assume $cost[c_i, c_j]$ indicates the cost associated with misclassifying an example from class $c_j$ to $c_i$, e.g., according to Table 1 the cost if misclassifying $c_{min}$ as $c_{maj}$ is 7, while the cost of an opposite mistake is 3.

An analysis of the nearest neighborhood of the seed is aimed at identifying *minimum-cost classes*, i.e., classes leading to the minimum cost after being (mis)classified as classes appearing in this neighborhood (we assume neighbors "attract" the seed thus leading to a possible misclassification). Formally, minimum-cost classes are identified as:

$$min\_cost\_classes\,(x, k, S, cost) = \underset{c_j \in C}{\arg\min} \sum_{c_i \in C} \frac{|knn\,(x, k, S, c_i)|}{k} \cdot cost\,[c_i, c_j],$$

where $C$ is a set of all classes ($C = \{c_{maj}, c_{int}, c_{min}\}$), $x$ is the seed, $k$ is the number of nearest neighbors, $S$ is a processed data set and $knn(x, kS, c_i)$ returns these among $k$-nearest neighbors of $x$ in $S$ that belong to class $c_i$. Given a set of minimum-cost classes for a given example $x$ we can categorize this examples as *safe* if its class is among minimum-cost classes, and *unsafe* otherwise.

Identification of minimal-cost classes occurs several times in SPIDER3. Its pseudo-code is given in Fig. 1. The algorithm accepts three parameters: $k$ – the number of considered nearest neighbors, $DS$ – a processed set and $cost$ – a cost matrix. In addition to symbols introduced earlier, $class\,(x)$ indicates the class of example $x$, $knn\,(x, k, S)$ returns a set of $k$-nearest neighbors of $x$ in set $S$, $nearest\,(x, S)$ returns the nearest example to $x$ in $S$, and finally $X\,[c_i]$ indicates these examples from $X$ that belong to class $c_i$ (in the pseudo-code $X$ is replaced either with $DS$ or $AS$).

**Table 1.** An example of a cost matrix

| | $c_{min}$ | $c_{int}$ | $c_{maj}$ |
|---|---|---|---|
| $c_{min}$ | 0 | 2 | 3 |
| $c_{int}$ | 3 | 0 | 2 |
| $c_{maj}$ | 7 | 5 | 0 |

```
1  procedure spider3($k, DS, cost$)
2      $AS := \{\}$ ;              /* a set of amplified or relabeled examples */
3      $RS := \{\}$ ;   /* a set of examples from $c_{maj}$ that can be discarded or
        relabeled to amplify other classes */
4      for $\forall x \in DS[c_{maj}]$ do
5          if $c_{maj} \notin$min_cost_classes($x, k, DS, cost$) then $RS := RS \cup \{x\}$;
6      end
7      $DS := DS \setminus RS$;
8      for $c_i \in \{c_{int}, c_{min}\}$ do
9          for $\forall x \in DS[c_i]$ do relabel_nn($x$);
10         for $\forall x \in DS[c_i] \cup AS[c_i]$ do clean_nn($x$);
11         for $\forall x \in DS[c_i]$ do amplify($x$);
12     end
13     $DS := DS \cup AS$
1  procedure relabel_nn($x$)
2      $TS := RS \cap$ knn($x, k, DS \cup AS \cup RS$) ;   /* candidates for relabeling */
3      while $TS \neq \{\} \wedge \{c_{maj}\} \in$ min_cost_classes($x, k, DS \cup AS \cup RS, cost$) do
4          $y :=$nearest($x, TS$);
5          $TS := TS \setminus \{y\}$;
6          $RS := RS \setminus \{y\}$;
7          class($y$) := class($x$);
8          $AS := AS \cup \{y\}$;
9      end
1  procedure clean_nn($x$)
2      $TS :=$ knn($x, k, DS \cup AS \cup RS, c_{maj}$) ;       /* candidates for removal */
3      while $TS \neq \{\} \wedge \{c_{maj}\} \in$ min_cost_classes($x, k, DS \cup AS \cup RS, cost$) do
4          $y :=$ nearest($x, TS$);
5          $TS := TS \setminus \{y\}$;
6          $DS := DS \setminus \{y\}$;
7          $RS := RS \setminus \{y\}$;
8      end
1  procedure amplify($x$)
2      while class($x$)$\notin$ min_cost_classes($x, k, DS \cup AS \cup RS, cost$) do
3          $y :=$ create a copy of $x$;
4          $AS := AS \cup \{y\}$;
5      end
```

**Fig. 1.** Pseudo-code for the SPIDER3 algorithm

In the first phase (lines 4–7 in the `spider3` procedure) the algorithm deals with the majority class. Specifically, it identifies unsafe examples that will be either relabeled to amplify the other classes or discarded. In the second phase (lines 8–13) the algorithm processes the remaining classes – $c_{int}$ and $c_{min}$. It iterates over examples from these classes and for each example $x$ it attempts to strengthen $x$ and its neighborhood. First, the algorithm relabels selected neighbors of $x$ (`relabel_nn`), then it cleans the neighborhood by removing some of the neighbors (`clean_nn`), and finally it amplifies $x$ (`amplify`). The algorithm starts with relabeling, and then proceeds with cleaning to produce such neighborhoods of examples from $c_{int}$ and $c_{min}$ that are dense and isolated from $c_{maj}$ examples. Starting with cleaning may have resulted in relabeling seemingly distant examples and obtaining more sparse and “ragged” neighborhoods.

Relabeling is limited to unsafe examples from $c_{maj}$ in the nearest neighborhood of the current example $x$ (line 2 in `relabel_nn`). In each iteration a candidate nearest to $x$ is selected and its class is changed to the class of $x$ (lines 4–8). This is repeated as longs as $c_{maj}$ is among minimum-cost classes for the current example $x$ and there are candidates to relabel. Cleaning removes these unsafe examples from $c_{maj}$ that are among nearest neighbors of $x$ (line 2 in `clean_nn`). Again, this process is repeated while $c_{maj}$ is among minimum-cost classes for $x$ and there are candidates for removal. Finally, $x$ is amplified by creating its copies (lines 3–4 in `amplify`) until it becomes safe, i.e., its class appears among its minimum-cost classes.

## 4 Experimental Design

The computational experiment was aimed at comparing the performance of selected classifiers combined with SPIDER3 to the performance of the same classifiers combined with SPIDER2, and at examining how the expert knowledge about relations between classes (captured as misclassification costs) affected the performance of classifiers combined with SPIDER3. We were primarily interested in accuracies for specific classes, therefore we used *true-positive rate* (TPR) for these classes as primary evaluation measures. We also considered their aggregation using a geometric mean (GM) [9].

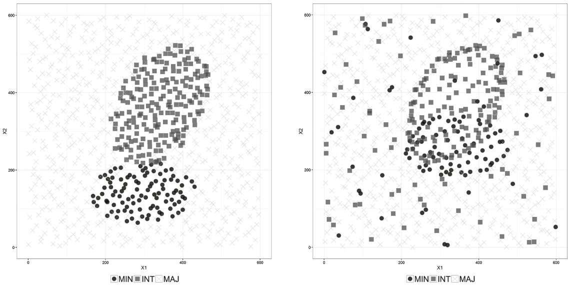
SPIDER2 is limited to handle data with binary classes. Thus, in order to apply it to 3-class data sets we had to introduce a temporary aggregation of classes (i.e., selected classes were aggregated prior to preprocessing, and disaggregated before building a classifier). Such an approach is conceptually similar to OVA, however, it does not introduce additional complexity (only a single aggregation is considered) and a constructed classifier relies on original classes. We considered two aggregation variants: with $c_{min}$ and $c_{int}$ combined into a new temporary minority class and with $c_{int}$ and $c_{maj}$ combined into a new temporary majority class. In subsequent tables and text we denote the first variant as SP2-min+int, and the second one as SP2-int+maj.

We also considered two variants of SPIDER3: with default costs (corresponding to a situation when misclassification costs are not available [2]) and with costs given in Table 1. These costs represent “similarity” relations between $c_{min}$ and $c_{int}$, and “dissimilarity” ones between these two classes and $c_{maj}$. The first variant of SPIDER3 is further denoted as SP3-default, while the second one as SP3-costs. In the study we included the following classifiers: k-NN (with $k = 1$ and $k = 3$, denoted as 1-NN and 3-NN), unpruned decision rules induced with PART (PART-U) and unpruned decision trees generated with C4.5 (C45-U). All these classifiers were implemented in WEKA[1]. Selection of classifiers was driven by experience from our past studies ([6,9]). We also decided not to include classifiers that require extensive parameter tuning, e.g., support vector

[1] http://www.cs.waikato.ac.nz/ml/weka/.

machines. While this may be seen as a shortcoming, our primary goal was to compare performance of both versions of SPIDER as opposed to find an optimal combination of a preprocessing method and a classifier. To provide a baseline, all classifiers were also applied without any preprocessing (denoted as none).

In the experiment we used 2-dimensional artificial data sets created with our generator to control data difficulty factors. While focusing on 2 dimensions may be seen as a limitation, it allowed us for better visual verification of generated data sets. Specifically, we considered 9 data configurations of varying difficulty. In all these configurations, $c_{min}$ and $c_{int}$ classes formed two elliptical regions and $c_{maj}$ surrounded them. To control difficulty, we modified overlapping between $c_{min}$ and $c_{int}$ classes and the distribution of example types in these classes based on the taxonomy proposed in [6]. We considered three possibilities of overlapping: none, medium and strong (denoted as 0, 1 and 2) and three distributions of example types: 70:30:0:0, 40:50:10:0 and 30:40:15:15 (specific numbers correspond to the ratio of safe, borderline, rare and outlier examples respectively). In Fig. 2 we present the easiest (70:30:0:0, overlap = 0) and the most difficult configuration (30:40:15:15, overlap = 2) among the considered ones.



(a) 70:30:0:0, overlap=0 (b) 30:40:15:15, overlap=2

**Fig. 2.** Selected configurations of artificial data sets considered in this study

## 5 Results of Experiment

Results of the experiment are given in Table 2. Due to space restrictions it is limited to two classifiers – 1-NN and PART-U (3-NN performed similarly to 1-NN, and C45-U to PART-U), and we focus on TPRs for specific classes (GM is discussed in the text). The best value for a given classifier and data configuration are marked with bold font, while the second best – with italics. The most important observations are the following:

- SPIDER3 worked better on difficult data configurations (distributions 40:50:-10:0 and 30:40:15:15) when it was combined with 1-NN. Such combination led the best TPR for $c_{min}$ and $c_{int}$ which implies it is especially well suited for

**Table 2.** TPRs for specific decision classes and considered data configurations

| Distribution | Overlap | Method | 1-NN | | | PART-U | | |
|---|---|---|---|---|---|---|---|---|
| | | | $c_{min}$ | $c_{int}$ | $c_{maj}$ | $c_{min}$ | $c_{int}$ | $c_{maj}$ |
| 70:30:0:0 | 0 | None | 0.772 | 0.838 | **0.913** | 0.804 | 0.961 | **0.870** |
| | | SP2-min+int | *0.936* | 0.838 | 0.873 | *0.958* | **0.982** | 0.835 |
| | | SP2-int+maj | 0.918 | 0.838 | *0.901* | **0.962** | 0.956 | 0.848 |
| | | SP3-default | 0.922 | *0.964* | 0.882 | 0.948 | 0.971 | *0.856* |
| | | SP3-costs | **0.972** | **0.992** | 0.856 | *0.958* | *0.973* | 0.842 |
| | 1 | None | 0.638 | 0.792 | **0.929** | 0.690 | **0.946** | **0.900** |
| | | SP2-min+int | 0.772 | 0.792 | 0.897 | 0.786 | *0.904* | 0.881 |
| | | SP2-int+maj | **0.860** | 0.766 | *0.918* | *0.946* | 0.845 | 0.893 |
| | | SP3-default | 0.758 | *0.906* | 0.901 | 0.926 | 0.844 | *0.897* |
| | | SP3-costs | *0.788* | **0.922** | 0.880 | **0.960** | 0.840 | 0.893 |
| | 2 | None | 0.434 | 0.680 | **0.933** | 0.522 | **0.769** | **0.933** |
| | | SP2-min+int | *0.548* | 0.680 | 0.911 | 0.796 | *0.718* | 0.909 |
| | | SP2-int+maj | **0.792** | 0.608 | *0.927* | *0.924* | 0.601 | *0.919* |
| | | SP3-default | 0.532 | *0.765* | 0.913 | 0.912 | 0.630 | 0.913 |
| | | SP3-costs | 0.552 | **0.778** | 0.898 | **0.958** | 0.629 | 0.902 |
| 40:50:10:0 | 0 | None | 0.674 | 0.769 | **0.863** | 0.764 | 0.879 | **0.841** |
| | | SP2-min+int | 0.868 | 0.769 | 0.815 | *0.872* | 0.884 | 0.793 |
| | | SP2-int+maj | 0.850 | 0.767 | *0.844* | 0.860 | 0.835 | *0.829* |
| | | SP3-default | *0.890* | *0.961* | 0.760 | 0.842 | *0.894* | 0.788 |
| | | SP3-costs | **0.932** | **0.969** | 0.748 | **0.892** | **0.909** | 0.779 |
| | 1 | None | 0.568 | 0.730 | **0.878** | 0.510 | **0.846** | **0.880** |
| | | SP2-min+int | 0.710 | 0.730 | 0.838 | 0.700 | *0.790* | 0.834 |
| | | SP2-int+maj | *0.758* | 0.704 | *0.863* | *0.836* | 0.695 | *0.867* |
| | | SP3-default | 0.738 | *0.898* | 0.790 | 0.820 | 0.756 | 0.827 |
| | | SP3-costs | **0.764** | **0.907** | 0.777 | **0.880** | 0.751 | 0.809 |
| | 2 | None | 0.458 | 0.702 | **0.895** | 0.292 | *0.840* | **0.914** |
| | | SP2-min+int | 0.596 | 0.702 | 0.860 | 0.314 | **0.875** | 0.848 |
| | | SP2-int+maj | **0.688** | 0.625 | *0.880* | 0.846 | 0.520 | *0.895* |
| | | SP3-default | 0.656 | *0.838* | 0.810 | *0.856* | 0.546 | 0.860 |
| | | SP3-costs | *0.664* | **0.843** | 0.802 | **0.882** | 0.563 | 0.838 |
| 30:40:15:15 | 0 | None | 0.480 | 0.584 | **0.803** | 0.584 | 0.656 | **0.843** |
| | | SP2-min+int | 0.664 | 0.584 | 0.751 | *0.678* | 0.611 | 0.690 |
| | | SP2-int+maj | 0.684 | 0.567 | *0.792* | **0.688** | 0.563 | *0.831* |
| | | SP3-default | *0.758* | *0.884* | 0.581 | 0.584 | *0.693* | 0.640 |
| | | SP3-costs | **0.806** | **0.907** | 0.523 | 0.664 | **0.868** | 0.273 |
| | 1 | None | 0.448 | 0.598 | **0.788** | 0.420 | 0.623 | **0.898** |
| | | SP2-min+int | 0.594 | 0.598 | 0.742 | 0.470 | *0.696* | 0.667 |
| | | SP2-int+maj | 0.630 | 0.564 | *0.777* | 0.640 | 0.485 | *0.848* |
| | | SP3-default | *0.692* | *0.853* | 0.556 | *0.646* | 0.602 | 0.604 |
| | | SP3-costs | **0.738** | **0.871** | 0.517 | **0.690** | **0.745** | 0.316 |
| | 2 | None | 0.418 | 0.555 | **0.794** | 0.174 | 0.606 | **0.932** |
| | | SP2-min+int | 0.528 | 0.555 | 0.755 | 0.270 | **0.752** | 0.613 |
| | | SP2-int+maj | 0.588 | 0.517 | *0.785* | 0.666 | 0.369 | *0.877* |
| | | SP3-default | *0.638* | *0.806* | 0.561 | **0.694** | 0.464 | 0.669 |
| | | SP3-costs | **0.668** | **0.825** | 0.522 | *0.680* | *0.634* | 0.280 |

difficult data sets (such finding is consistent with our earlier studies related to $k$-NN classifiers [9]). While the best performance was observed for SP3-costs, SP3-default was usually the second best. On the one hand, it proves information about misclassification costs is beneficial for the performance, but on the other hand, it demonstrates SPIDER3 can work reasonably well when this information is not available.

- There are no clear patterns in the performance of PART-U. In most cases SP3-costs led to the best TPR for $c_{min}$, but at the same time the best TPR for $c_{int}$ was associated with SP2-min+int or no preprocessing. This implies that temporal aggregation of $c_{min}$ and $c_{int}$ may have resulted in more cleaner boundary between $c_{int}$ and $c_{maj}$ which was beneficial for a symbolic classifier.
- All preprocessing methods deteriorated the performance for $c_{maj}$ and this change was smaller for SPIDER2 (especially for the 30:40:15:15 distribution and PART-U). However, the performance of SIDER3 for $c_{min}$ and $c_{int}$ was better and more even, therefore GM favored (with the only exception mentioned above) our new algorithm for its ability to ensure consistently good performance across all classes. These observations were confirmed by Friedman tests. While a small number of data sets did not allow for sound statistical analysis, average ranks of SPIDER3 were for most classifiers favorable to those of SPIDER2.

## 6 Discussion

Multi-class imbalanced classification problems pose a significant challenge that calls for new processing and learning methods. We have responded to this call by proposing SPIDER3 – an algorithm for preprocessing of multi-class data sets (in this paper we focus on 3 classes – minority, intermediate and majority – but the algorithm handles more classes). SPIDER3 processes classes according to their global relevance (starting with the least relevant one) and uses optional information about local relations between pairs of classes (captured as misclassification costs) to better control introduced modifications.

SPIDER3 was evaluated in a computational experiment on artificial data and compared to its predecessor – SPIDER2 – which is limited to binary problems and requires temporary aggregation of classes. SPIDER3 resulted in a better performance for the minority class, and when combined $k$-NN classifiers also for the intermediate class. Moreover, its performance across all decision classes was more even and resulted in a better geometric mean of their true positive rates. Given these findings, we posit SPIDER3 is a valid and interesting technique for multi-class imbalanced data sets.

We also performed an additional experiment involving real-life clinical data that has confirmed our initial observations[2]. Now we plan to work on improving processing of the majority class to limit the deterioration of performance

[2] See the on-line appendix available at http://www.cs.put.poznan.pl/swilk/cores2017/spider3-appendix.pdf.

observed for difficult data sets and on the techniques for tuning misclassification costs for specific problems.

**Acknowledgments.** The authors would like to acknowledge support by the Polish National Science Center under Grant No. DEC-2013/11/B/ST6/00963.

## References

1. Elkan, C.: The foundations of cost-sensitive learning. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence, IJCAI 2001, vol. 2, pp. 973–978. Morgan Kaufmann Publishers Inc., San Francisco (2001). http://dl.acm.org/citation.cfm?id=1642194.1642224
2. Fernández, A., López, V., Galar, M., Del Jesus, M.J., Herrera, F.: Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. Know.-Based Syst. **42**, 97–110 (2013). http://dx.doi.org/10.1016/j.knosys.2013.01.018
3. He, H., Ma, Y.: Imbalanced Learning: Foundations, Algorithms and Applications. Wiley, New York (2013)
4. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Prog. Artifi. Intell. **5**(4), 221–232 (2016)
5. Maciejewski, T., Stefanowski, J.: Local neighbourhood extension of SMOTE for mining imbalanced data. In: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, pp. 104–111 (2011)
6. Napierala, K., Stefanowski, J.: Types of minority class examples and their influence on learning classifiers from imbalanced data. J. Intell. Inform. Syst. **46**, 563–597 (2016)
7. Napierała, K., Stefanowski, J., Wilk, S.: Learning from imbalanced data in presence of noisy and borderline examples. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS (LNAI), vol. 6086, pp. 158–167. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13529-3_18
8. Sáez, J.A., Krawczyk, B., Woźniak, M.: Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. Pattern Recognit. **57**, 164–178 (2015)
9. Wilk, S., Stefanowski, J., Wojciechowski, S., Farion, K.J., Michalowski, W.: Application of preprocessing methods to imbalanced clinical data: an experimental study. In: Piętka, E., Badura, P., Kawa, J., Wieclawek, W. (eds.) Information Technologies in Medicine. AISC, vol. 471, pp. 503–515. Springer, Cham (2016). doi:10.1007/978-3-319-39796-2_41

# The Method of Person Verification by Use of Finger Knuckle Images

Rafal Doroz(✉), Krzysztof Wrobel, Piotr Porwik, and Hossein Safaverdi

Institute of Computer Science, University of Silesia,
ul. Bedzinska 39, 41-200 Sosnowiec, Poland
{rafal.doroz,krzysztof.wrobel,piotr.porwik,hossein.safaverdi}@us.edu.pl
http://zsk.tech.us.edu.pl
http://biometrics.us.edu.pl

**Abstract.** The paper proposes a personal identity verification method based on images of finger knuckles. The knuckle images were recorded using a digital camera and then proceed to extract the furrows appearing on them. The verification was performed by comparing the locations and courses of the furrows on the pattern being verified and on the reference image of finger knuckles. In order to determine the similarity between the images, a new similarity measure was proposed. During the analysis of finger knuckle images, there appears a problem which consists in the fact that the location and size of the same furrow may be different in subsequent images obtained from the same person. This problem results from the elasticity of the human skin. To minimize the problem in question, this paper proposes a solution that consists in matching the furrows with each other before they are compared. For this purpose, a method based on Thin Plate Spline and Shape Context has been used. The usability of this method was verified experimentally.

**Keywords:** Biometrics · Finger-knuckle · Image pre-processing · Person verification

## 1 Introduction

Nowadays, one of the applications of biometric systems is to control access to different kinds of resources [1,2]. This task can be accomplished with biometric techniques such as fingerprints, iris and handwritten signature [3–5]. These methods are well known, while their disadvantages and advantages have been presented in many publications [6]. However, there is a continuous search for new biometric methods and the current ones are modified in order to improve their effectiveness. The paper presents a biometric verification method based on the analysis of a human finger knuckles [7]. It is a relatively new biometric method in comparison with the analysis of e.g. fingerprints or handwritten signatures, but in the future it may be used in unimodal and multimodal biometric systems. An analysis of a finger knuckle image performed with the use of advanced image

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_26

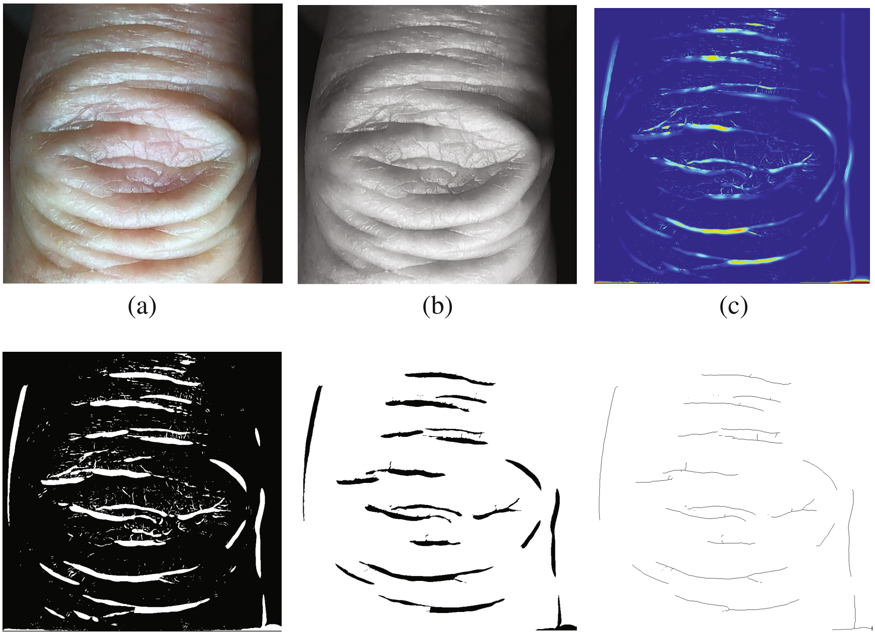
processing techniques allows extracting furrows located on the surface of the finger knuckle [8]. Figure 1c shows an example of a finger knuckle image with visible furrows. As demonstrated in [9], furrows are features unique for each person and thus they may be considered as a new set of biometric features. An unquestionable advantage of the proposed method of analysis of finger knuckle images is that it is enough to use a digital camera or video camera to acquire such images. It should also be noted that the acquisition of a finger knuckle image is contactless. So far, biometric systems based on an analysis of a human finger knuckle have been described in many publications. In [10], Hidden Markov Models (HMM) and SVM method were used to classify the features extracted from a finger knuckle. In [11], finger knuckle images were represented by a code system proposed by the authors, while for their extraction and classification there were used, inter alia, the following methods: Radon transformation, PCA analysis, independent component analysis (ICA), and linear discriminant analysis (LDA). In the studies dedicated to the analysis of finger knuckles, there were used the techniques such as Gabor filter [12], surface curvature analysis [13], as well as the texture analysis [9] and the SIFT method [14].

## 2 Proposed Method

The method proposed in this paper consists of two main stages. During the enrollment stage, a number of finger knuckle images is taken from each user. The method presented here is based on a comparison of furrows located on finger knuckles and therefore Hessian filter [15] is applied to the image in order to improve visibility of furrows. Then, a search for furrows in the image is performed and each furrow is converted into a chain of points. All the chains of points determined for a given finger knuckle image form the Finger Knuckle Pattern (FKP) of a given user. The last stage of the enrollment process is to add to the database all the finger knuckle images together with the FKPs created on their basis. During the verification stage of an unknown user, an image of his/her finger knuckle is taken and, similarly as at the stage of enrollment, it is processed to extract chains of points. In addition, before the process of chain extraction, the image being verified and the reference image from the database are superimposed and matched for best fit. The matching consists in adequate deformation of the reference image in order to eliminate slight differences in the positions of the furrows. Chains of points are extracted from the image being verified, after matching it with the reference image, and compared with the reference chains of points. A similarity coefficient based on the Czekanowski coefficient [4] was used for comparing the chains. The next stages of the method are described in detail further in this paper.

## 3 Detection of Finger Knuckle Patterns

The first stage of the analysis of finger knuckle images consists in the detection of furrows located on them. The analysed image (an example is shown in Fig. 1a)



(a) (b) (c)

(d) (e) (f)

**Fig. 1.** Stages of furrows extraction in a finger knuckle image: (a) original image, (b) original image converted to greyscale, (c) original image subjected to FRF, (d) binarized image, (e) image after removal of artifacts, (f) image after skeletonization.

is converted to grayscale (Fig. 1b). The next stage is to use Frangi filter (FRF) (Fig. 1e). This filter allows making visible dark objects on a light background and vice versa [16]. In our case, dark objects are the furrows located on the knuckles. Then, the values obtained after FRF filtration are thresholded in order to extract the furrows (Fig. 1f). After the operation of thresholding, there may remain binary artifacts in the image. They may be misinterpreted as elements of the furrows and consequently hinder the further analysis. Therefore, they are removed from the image using the erosion and closing operations [17] (Fig. 1g). The image prepared in this way is subjected to the operation of skeletonization (Fig. 1h).

### 3.1 Matching Knuckle Images

In the course of the analysis of knuckle images, we can notice a problem which consists in the fact that the location and size of the same furrow may be different in subsequent images obtained from the same person. This problem results from a high flexibility of human skin and has been described in [18]. In order to minimize the problem mentioned above, a solution is proposed in this paper that consists in matching the furrows before they are compared. In the proposed method we treat the knuckle image $I^1$ as a set of $n$ black points $P = \{p_1, ..., p_n\}$. For each point $p_i$ on the image $I^1$, we compute a histogram $h_i$ of the relative coordinates of the remaining $n-1$ black points:

$$h_i\left(k\right) = card\left\{q \neq p_i : \left(q - p_i\right) \in bin\left(k\right)\right\}, \quad k = 1, ..., n-1, \tag{1}$$

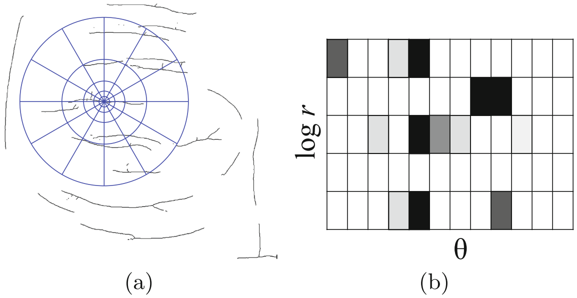
where $bin(k)$ is $k$'th element of histogram.

This histogram is the Shape Context of point $p_i$ (Fig. 2b). We make this histogram for random chosen $K$ black points in our image. The influence of the $K$ parameter on verification accuracy has been described in research section. To make the descriptor more sensitive to positions of nearby sample points than the farther ones we used bins that are uniform in log-polar space (Fig. 2a). For each point $p_i$ on the first knuckle image $I^1$ we have to find the best matching point $q_j$ on the second knuckle image $I^2$. To compute the cost of the matching of these two points we used the following formula:

$$C_{ij} \equiv C\left(p_i, q_j\right) = \frac{1}{2}\sum_{b=1}^{B}\frac{\left[h_i\left(b\right) - h_j\left(b\right)\right]^2}{h_i\left(b\right) + h_j\left(b\right)}, \tag{2}$$

where $h_i(b)$ and $h_j(b)$ denotes the $b'th$ bin histogram at $p_i$ and $q_j$ respectively and $B$ is the number of bins.

Shape contexts are extremely rich descriptors, they are insensitive to small perturbations of part of the knuckle images. Given a set of correspondences between points on two shapes, can lead to estimate a plane transformation $Tr: R^2 \rightarrow R^2$ which may be used to map arbitrary points from one shape to the other. We need to choose $Tr$ from suitable family of transformations. One way is to use Thin Plate Spline (TPS) model which is frequently used to represent flexible coordinate transformation. TPS is a spline-based technique for data interpolation and smoothing. Thanks to this method, during the matching of the image $I^1$ and image $I^2$, one of them is deformed, so as to superimpose to another image. The degree of deformation can be controlled by changing the parameter $J$, which determines the number of iteration to match two images. Increase in $J$ value



(a) (b)

**Fig. 2.** (a) Diagram of log-polar histogram bins used in computing the Shape Contexts, (b) Example Shape Context.

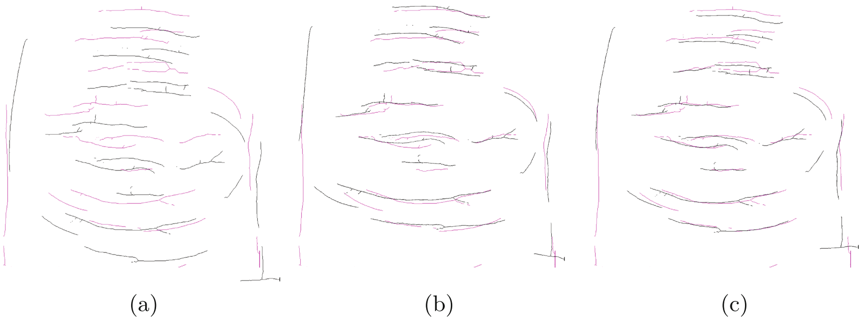


**Fig. 3.** Two overlapped knuckle images (marked by black and red color): (a) without matching, (b) matching with the use of SC and TPS, one iteration, ($J = 1$), (c) two iterations, ($J = 2$).

leads to better images matching. Examples of matching two knuckle images, for different values of the parameter $J$ are shown in Fig. 3.

A detailed description of the Shape Context method can be found in [19].

### 3.2 Extraction of Chains of Points

The next stage consists in the conversion of the furrows visible in the finger knuckle images into chains of points. Such a representation of furrows is used to determine the similarity between them. In order to determine the chain of points in the skeletonised image, the points of ends and bifurcations of the furrows are located. These points are located by counting the number of black pixels in the vicinity $3 \times 3$ of the pixel being analysed, where the middle pixel is the analysed pixel, while 8 pixels on the sides of the matrix are "neighbours". For each black pixel, the number of black pixels in its vicinity is calculated according to the following formula:

$$J(x,y) = \sum_{a=-1}^{1} \sum_{b=-1}^{1} I(x+a, y+b), \quad x \in [2, ..., W-1], y \in [2, ..., H-1], \tag{3}$$

where $(x, y)$ are coordinates of analyzed point, $W$ and $H$ are the width and height of the image $I$, respectively. Based on the value $J(x, y)$ calculated for each black pixel, $p(x, y)$ the label $p^T$ (end of the furrow) or $p^B$ (bifurcation on the furrow) is assigned to this pixel using the following formula:

$$p(x,y) = \begin{cases} p^T(x,y) & \text{if} \quad J(x,y) = 2 \\ p^B(x,y) & \text{if} \quad J(x,y) > 3 \end{cases}. \tag{4}$$

The image may contain many points, in which furrow ends or bifurcations are located. In a further part of this study, in order to simplify the description, these points will be designated as $p^{T_i}$ and $p^{B_j}$, where $i$ and $j$ are numbers of ends and bifurcations. After labeling the all points of ends and bifurcations, the procedure of extracting from the image the chain of points representing individual fragments of the furrows begins. This task is performed with the use of Algorithm 1, which is executed separately for each image of finger knuckles. As a result, each image $I$ is described by a set $C = \{c_1, ...\}$ containing all the chains found in this image. Each $i$'th chain $c_i$ consists of points forming the $i$'th furrow.

**Algorithm 1.** Procedure of extracting the chains of points from the knuckle image.

```
Data: Thinned fingerprint image; List of the image points with labels
      $W = \{p^{T_1}, p^{T_2}, ..., p^{B_1}, p^{B_2}, ...\}$
Result: List of chains of the points $C = \{c_1, ...\}$, where each chain $c_i$ has
        starting and ending points with any type label ($p^T$ or $p^B$)
$i = 1$;
foreach labeled point $p \in W$ of the set $W$ do
    add coordinates of the point $p \in W$ to the chain $c_i$;
    do
        move the analyzed point from the point $p$ to the neighbor black pixel $p^*$
          which not belongs to any chain from the list $C$;
        add point $p^*$ to the chain $c_i$;
        set analyzed point $p = p^*$;
    while the analyzed point $p \notin W$;
    $i = i + 1$;
end
```

## 4 Determination of the Similarity Between Images

In the verification phase, an unknown person claims his/her identity and provides a knuckle image to be verified. This image is compared sequentially with all the reference images in the database taken from the person being verified. The similarity between the selected reference image $I^R$ and the image being verified $I^V$ is determined by finding for each chain in the reference image a chain in the image being verified that is most similar to it. After the analysis of all chains in the reference image, the similarity values obtained are averaged. Since individual images differ from each other in a number of chains determined in them, therefore they were compared with the use of a measure based on the Czekanowski coefficient which allows comparing data sets with a different size. The final formula determining the similarity between the images $I^R$ and $I^V$ is given as follows:

$$sim(I^R, I^V) = \frac{1}{m} \sum_{i=1}^{m} \min\left(d(c_i^R, c_1^V), d(c_i^R, c_2^V), ..., d(c_i^R, c_n^V)\right), \quad (5)$$

where:
$C^R = \{c_1^R, c_2^R, ..., c_m^R\}$ – the set of chains describing the reference image $I^R$,
$C^V = \{c_1^V, c_2^V, ..., c_n^V\}$ – the set of chains describing the verified image $I^V$.
If $sim(I^R, I^V) = 0$ this means that the images being compared are identical. The similarity decreases along with an increase in the value $sim(I^R, I^V)$.
The value of the distance $d(c_i^R, c_j^V)$ between the two chains being compared $c_i^R$ and $c_j^V$ is calculated using the following formula:

$$d(c_i^R, c_j^V) = \frac{1}{r}\sum_{k=1}^{r} \min\left(euc(p_k^R, p_1^V), euc(p_k^R, p_2^V), ..., euc(p_k^R, p_s^V)\right), \tag{6}$$

where $r$ and $s$ are the numbers of points in the chain $c_i^R$ and $c_j^V$, respectively, while $euc(p_k^R, p_s^V)$ is the Euclidean distance between the point $p_k^R$ in the reference chain $c_i^R$ and the point $p_s^V$ in the chain being verified $c_j^V$. Since the image being verified is compared with $w$ reference images $I^{R_i}$ from the database, where $i = 1, ..., w$, as a result there are obtained $w$ values of similarities $sim(I^{R_i}, I^V)$, $i = 1, ..., w$. If $sim(I^{R_i}, I^V)$ is less than the assumed threshold $T$, the image being verified is considered to be similar to the $i$'th reference image, and the $i$'th result of the verification there assigned the value $D_i = 1$. Otherwise, the image is considered to be dissimilar, and the value $D_i = -1$:

$$D_i = \begin{cases} 1 \; if \; sim(I^{Ri}, I^V) < T \\ -1 \; if \; sim(I^{Ri}, I^V) \geq T. \end{cases} \tag{7}$$

The threshold $T$ value was calculated in the research part. The final decision determining whether the image being verified $I^V$ is original or is a forgery is made on the basis of the majority voting of the results of the comparison of the image being verified with all the images in the database.

$$I^V = \begin{cases} orginal \quad if \quad \sum_{i=1}^{w} D_i \geq 1 \\ false \quad otherwise \end{cases} \tag{8}$$

If, as a result of majority voting, the image being verified $I^V$ was found to be similar to a majority of the reference images, the image is belong to the individual who the verified person claims to be and otherwise is a forgery. In order to obtain an unambiguous decision, the number of reference images $w$ in the database should be odd.

## 5 Experiments

The effectiveness of the proposed method was determined experimentally. For the needs of the tests, a database containing 150 images obtained from 30 people (5 images from each person) was used. The database and the test rig for acquisition of finger knuckle images are described in detail in [20]. When performing the tests, among all the finger knuckle images belonging to a given person, one image was selected successively and verified by a comparison with three images of this person selected randomly from the remaining images. In order to evaluate the effectiveness of the detection of forgeries, reference images of a given person were compared with a randomly selected image of another person. The effectiveness of this method was calculated using Receiver Operating Characteristic (ROC). Examples of graphs determined for different parameters of the method are shown in Fig. 4. The point of intersection of the $FAR/FRR$ curves determines the $EER$ error and the value of the decision threshold $T$, for which the smallest EER error was obtained (see Eq. 7).

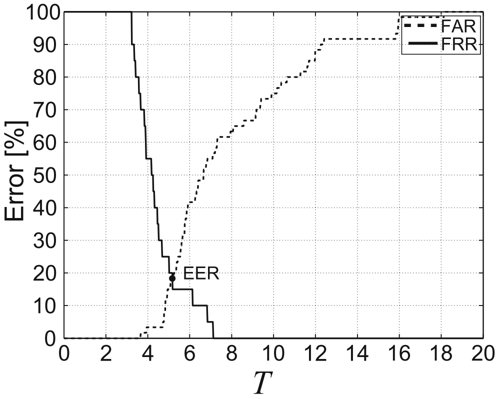


**Fig. 4.** Receiver Operating Characteristic (ROC)

During the tests, the following analyses were performed:

- impact of the parameter $J$ (number of iterations during the process of matching the images being compared) on the effectiveness of the method,
- impact of the number of points $K$ found on each image during the process of matching on the effectiveness of the method.

The results of the tests carried out for different values of the parameters mentioned above are presented in Table 1. The results of the tests showed that the smallest error of verification was obtained in the case of 2 iterations. An increase in the number of iterations to more than 2 caused a too large degree of matching the original image to the forged one. As a result, the level of similarity between images from different individuals is high, which increases the value of EER. In turn, a too little number of iterations ($K < 2$) causes that it is not possible to eliminate minor differences between the furrows coming from two images of the same person. These differences are inherent and result from the factors described in Sect. 3.1. When analysing the impact of the number of points $K$ found on each image during the process of matching, it can be seen that the minimum EER

**Table 1.** The EER [%] values obtained for different combinations of values of the parameters.

| Number of points $K$ | Number of iterations $J$ | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| 10 | 17.85 | 16.21 | 18.58 | 20.34 |
| 50 | 13.93 | 9.52 | 10.45 | 15.68 |
| 100 | 10.43 | 5.54 | 6.71 | 11.05 |
| 200 | 9.76 | 5.55 | 6.44 | 10.96 |
| 300 | 9.56 | 5.54 | 6.46 | 10.83 |

error stabilizes at a level of 100 points. This suggests that it is enough to determine only 100 points on the furrows to perform an effective verification. In order to assess the usability of the preliminary stage of superimposition of images (described in Sect. 3.1), the results obtained were compared with the results of a method, in which this stage was omitted. In this a case, the value of the EER was significantly higher – at a level of 21.48%.

## 6 Conclusions

The knuckle recognition method described in the paper appeared to be a fully functional solution. From the experiments carried out follows that finger-knuckle based technique is a promising method of biometric solutions. In our solutions, knuckles features extraction is based on Hessian filter, SC+TPS and Czekanowski coefficient. A satisfactory effectiveness of recognition was obtained using well-known similarity measures which are easy to implement. The results obtained, despite their preliminary nature, are promising and this method can be further developed, so it is worth to continue working on it. Future works will include testing the other databases. It is also planned to introduce more sophisticated classification methods. The study will also cover the work on proposing other methods of pre-processing the image in order to more accurately visualize the furrows in the knuckle images.

## References

1. Campisi, P. (ed.): Security and Privacy in Biometrics. Springer, London (2013)
2. Kudłacik, P., Porwik, P., Wesołowski, T.: Fuzzy approach for intrusion detection based on user's commands. Soft Comput. **20**(7), 2705–2719 (2016)
3. Kasprowski, P.: The impact of temporal proximity between samples on eye movement biometric identification. Lecture Notes in Computer Science, vol. 8104, pp. 77–87 (2013)
4. Porwik, P., Doroz, R., Wrobel, K.: A new signature similarity measure. In: Proceedings of the 2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009, pp. 1022–1027 (2009)

5. Wrobel, K., Doroz, R., Porwik, P.: Fingerprint reference point detection based on high curvature points. Lecture Notes in Computer Science, vol. 9714, pp. 538–547 (2016)
6. Jain, A., Flynn, P., Ross, A.A. (eds.): Handbook of Biometrics. Springer, US (2008)
7. Choraś, M., Kozik, R.: Knuckle recognition for human identification. Advances in Intelligent and Soft Computing (AINSC), vol. 95, pp. 61–70 (2011)
8. Kumar, A., Wang, B.: Recovering and matching minutiae patterns from finger knuckle images. Pattern Recogn. Lett. **68**, 361–367 (2015)
9. Kumar, A., Ravikanth, C.: Personal authentication using finger knuckle surface. IEEE Trans. Inf. Forensics Secur. **4**(1), 98–110 (2009)
10. Ferrer, M.A., Travieso, C.M., Alonso, J.B.: Using hand knuckle texture for biometric identifications. IEEE Aerosp. Electron. Syst. Mag. **21**(6), 23–27 (2006)
11. Kumar, A., Zhou, Y.: Human identification using knucklecodes. In: IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems, pp. 98–109 (2009)
12. Xiong, M., Yang, W., Sun, C.: Finger-knuckle-print recognition using LGBP. Lecture Notes in Computer Science, vol. 6676, pp. 270–277 (2011)
13. Woodard, D.L., Flynn, P.J.: Finger surface as a biometric identifier. Comput. Vis. Image Underst. **100**(3), 357–384 (2005)
14. Morales, A., Travieso, C.M., Ferrer, M.A., et al.: Improved finger-knuckle-print authentication based on orientation enhancement. Electron. Lett. **47**(6), 380–382 (2011)
15. Iwahori, Y., Hattori, A., Adachi, Y., et al.: Automatic detection of polyp using Hessian Filter and HOG features. Procedia Comput. Sci. **60**(1), 730–739 (2015)
16. Ng, C.-C., Yap, M.H., Costen, N., Li, B.: Automatic wrinkle detection using hybrid hessian filter. Lecture Notes in Computer Science, vol. 9005, pp. 609–622 (2015)
17. Soille, P.: Morphological Image Analysis. Springer, Heidelberg (2004)
18. Fager, M., Morris, K.: Quantifying the limits of fingerprint variability. Forensic Sci. Int. **254**, 87–99 (2015)
19. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. **24**, 509–522 (2002)
20. Doroz, R., et al.: A new personal verification technique using finger-knuckle imaging. Lecture Notes in Computer Science, vol. 9876, pp. 515–524 (2016)

# On a New Method for Improving Weak Classifiers Using Bayes Metaclassifier

Marcin Majak(✉) and Marek Kurzyński

Wrocław University of Science and Technology, Wrocław, Poland
{marcin.majak,marek.kurzynski}@pwr.edu.pl

**Abstract.** In this paper new algorithm called Bayes metaclassifier (BMC) will be introduced as a method for improving weak classifiers performance. In general, BMC constitutes the probabilistic generalization of any base classifier and has the form of the Bayes scheme. To validate BMC classification two experiments were designed. In the first one three synthetic datasets were generated from normal distribution to calculate and check empirically upper bound for improving base classifier when BMC approach is applied. Furthermore, to validate usefulness of this algorithm extensive simulations from 22 available benchmarks were performed comparing BMC model against 8 base classifiers with different design paradigms.

**Keywords:** Classification · Improving weak classifiers · Randomized reference classifier

## 1 Introduction

Pattern recognition task was successfully applied in many areas from medical, financial and critical safety applications. Each problem requires its individual approach and there is no easy solution which classifiers or algorithm should be applied because each method has its prons and cons. Nowadays, due to the huge development of the artificial intelligence, many classification methods are at hand, there are statistical, distance based, neural methods to name only a few [4,6,14]. Since classification problems are very complex and multidimensional, simple base classifier usage is not sufficient so advanced method for building multiclassifier systems (MC) are developed, but still proper base classifier choice is critical because we need accurate and diverse ones [19]. In the literature one can find many method for constructing robust MC models, but the most commonly used are bagging, boosting and random subspace [5,12,13]. In shortly, bagging applies sampling with replacement to obtain independent training datasets for each individual classifier. Boosting modifies the input data distribution processed by each classifier in a sequence from the results of classifiers trained before, paying more attention on difficult samples. All the aforementioned algorithms focus on using weak classifiers for building powerful MC systems. In this paper

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_27

we introduce an algorithm called Bayes metaclassifier (BMC) as a method for improving weak classifier in terms of its classification performance. In general, BMC constitutes the probabilistic generalization of any base classifier independent of its design paradigm and has the form of the Bayes scheme. Since BMC provides probabilistic interpretation for base classifier correct classification and misclassification, this method can be used in sequential classification or as a fusing mechanism in MC systems [10,11]. The main goal of this study is to investigate how BMC can improve base classifier performance. For this purpose two experiments were performed, the first one with synthetic data and the second one on 22 real life benchmark datasets. The paper is divided into four sections and organized as follows. In Sect. 2 pattern recognition task is introduced and later the concept of probabilistic Bayes metaclassifier is described. The results of computer experiments are described in Sect. 3 and Sect. 4 concludes the paper together with proposition of future plans and improvements.

## 2 Problem Statement

### 2.1 Pattern Recognition Task

This paper deals with pattern recognition task in which we assume that the pattern is in class $j \in \mathcal{M}$, where $\mathcal{M}$ is an $m$-element set of possible states numbered with the successive natural numbers ($j \in \mathcal{M} = \{1, 2, \ldots, M\}$). Label $j$ is unknown and does not undergo our direct observation. What we can only observe are the features by which an object manifests itself. We will denote a $d$-dimensional measured feature vector by $x \in \mathcal{X}$ (thus $\mathcal{X}$ is the feature vector space). In order to classify unknown patterns, as usual in practice, we assume that we have to our disposal so called training set, which in the investigated decision task consists of $N$ training patterns:

$$\mathcal{S} = \{(x_1, j_1), \ldots, (x_N, j_N)\}, \tag{1}$$

where $x_k, j_k$ denote $d$-dimensional pattern and its true classification, respectively. In general, the decision algorithm with learning should use every time as well observed data i.e. the feature vector $x$ as the knowledge included in the training set $\mathcal{S}$. In consequence, the general, classification algorithm with learning is of the following form:

$$i = \psi(\mathcal{S}, x), i \in \mathcal{M}. \tag{2}$$

$\psi$ is a canonical model of a base classifier which means that for a given $x \in \mathcal{X}$, it produces class label and a vector of class supports [8]. Single $d_j(x)$ indicates the support of $\psi$ to the hypothesis that the object $x$ belongs to the class $j$.

$$d(x) = [d_1(x), d_2(x), \ldots, d_M(x)], \sum_{i=1}^{M} d_i(x) = 1. \tag{3}$$

## 2.2 BMC Algorithm Construction

In this section BMC algorithm construction over single classifier will be presented. Firstly, the probabilistic model of classification must be introduced in which the feature vector $x \in \mathcal{X}$ and class label $j \in \mathcal{M}$ are observed values of the pair of random variables $\mathbf{X}$ and $\mathbf{J}$, respectively. The probability distribution of $(\mathbf{X}, \mathbf{J})$ is determined by the *a priori* class probabilities $p_j = P(\mathbf{J} = j)$ and class-conditional density functions $f(x|j) = f_j(x)$. The Bayes metaclassifier (BMC) $\psi^{BMC}$, which originally was introduced in [11], constitutes the probabilistic generalization of base classifier (2) which has the form of the Bayes scheme built over the classifier $\psi$. This means, that $\psi^{BMC}$ takes the decision according to the maximum *a posteriori* probability rule [10]:

$$\psi^{BMC}(\psi(x) = k|i) \longleftrightarrow p(i|\psi = k) = \max_{i \in \mathcal{M}} p(i|\psi = k). \tag{4}$$

*A posteriori* probabilities $p(i|k) \equiv P(\mathbf{J} = i|\psi(x) = k), i \in \mathcal{M}$ are given by Bayes rule:

$$P(\mathbf{J} = i|\psi = k) = \frac{p_i \; p(k|i)}{\sum_j p_j \; p(k|j)}, \tag{5}$$

where probability $p(k|i) \equiv P(\psi(x) = k|i)$ denotes class-dependent probability of error (if $k \neq i$) or correct (if $k = i$) classification of an object $x$ by the base classifier $\psi$. Placing the base classifier $\psi$ in a probabilistic frame defined by the BMC ($\psi^{BMC}$), we get a common probabilistic interpretation of responses of base classifiers, regardless of their design paradigms. The key element in the BMC scheme described by Eqs. (4) and (5), is the calculation of probabilities $P(\psi(x) = k|i)$ at point $x$, i.e. class-dependent probabilities of correct and misclassification for base classifier $\psi$. Normally, for any base deterministic classifier these probabilities are equal to 0 or 1. In this paper, the proposed method of evaluation of these probabilities is based on the original concept of a hypothetical classifier called a Randomized Reference Classifier (RRC) [16,17]. The RRC $\psi_l^{RRC}(x)$ is a stochastic classifier that classifies object $x$ according to the maximum rule for vector of class support $[\gamma_1(x), \gamma_2(x), \ldots, \gamma_M(x)]$ which are observed values of random variables (rvs) $[\varDelta_1(x), \varDelta_2(x), \ldots, \varDelta_M(x)]$. Probability distribution of rvs is chosen in such a way that RRC acts, on average, as a modeled base classifier only when the following constraints are fulfilled:

$$\begin{gathered} \varDelta_j(x) \in \langle 0, 1 \rangle, \sum_{j=1}^{M} \varDelta_j(x) = 1 \\ E[\varDelta_j(x)] = d_j(x), j \in \mathcal{M}, \end{gathered} \tag{6}$$

where $E$ is the expected value operator. Since RRC performs classification in a stochastic manner, so it is possible to calculate class-dependent probabilities of correct classification $P_c(j|x)$ and misclassification $P_e(j|x)$ and furthermore consider them equivalent to the modeled base classifier:

$$P(\psi(x) = k|i) \approx P(\psi^{RRC}(x) = k|i). \tag{7}$$

It should be noted that (7) is a heuristic approach for estimating base classifier probabilities of correct classification (misclassification). For BMC training phase it is assumed that a validation set containing pairs of feature vectors and their corresponding class labels is available [15]. This set can be obtained by random ratio subsampling from training set or alternatively as a stack generalization [18]:

$$\mathcal{V} = \{(x_1, j_1), (x_2, j_2), \ldots, (x_N, j_N)\}; \quad x_k \in \mathcal{X}, \; j_k \in \mathcal{M}. \tag{8}$$

$\mathcal{V}$ is used to calculate $P(\psi(x) = k|i) \approx P(\psi_l^{RRC}(x) = k|i)$ which denotes that an objects $x$ belongs to class $i$ given that $\psi(x) = k$. These values are only known at discrete points from $\mathcal{V}$ so to enable dynamic calculation of any new object $x$ from feature space we need a neighborhood function describing how probabilities at validation points affects new $x$. In this study, Gaussian potential function was used:

$$\begin{aligned} P_c^{RRC}(j|x) &= \frac{\sum_{x_k \in \mathcal{V}, j_k = j} P_c(j|x) \cdot \exp(-\alpha \cdot ||x, x_k||^2)}{\sum_{x_k \in \mathcal{V}, j_k = j} \exp(-\alpha \cdot ||x, x_k||^2)}, \\ P_e^{RRC}(j|x) &= \frac{\sum_{x_k \in \mathcal{V}, j_k \neq j} P_e(j|x) \cdot \exp(-\alpha \cdot ||x, x_k||^2)}{\sum_{x_k \in \mathcal{V}, j_k \neq j} \exp(-\alpha \cdot ||x, x_k||^2)}, \end{aligned} \tag{9}$$

where $\alpha$ value in Eq. (9) is a scaling factor and should be adjusted independently to classification problem. In this article, instead of manual selection, $\alpha$ is selected during preparation step of BMC using training and validation set. In this case, for each object from $\mathcal{V}$ we calculate $P_e(j|x), P_c(j|x)$ and further we randomly select 40% of objects from training set with stratification and use this new set to check for which $\alpha$ BMC obtains the highest classification rate. For all experiments, $\alpha$ was chosen from set: $\alpha = \{1, 2, 4, 6, 8\}$.

## 3 Experiments

The goal of this study was twofold. Firstly, we wanted to check how BMC works with synthetically generated data coming from 1D normal distribution described by $(\mu, \sigma)$. This is very simplified approach, but allows us to calculate an optimal Bayes decision boundary and Bayes probability of misclassification. In a consequence, we can determine empirical probability of error of BMC in relation to this probability. In the second experiment BMC model was applied to 22 benchmark data sets taken from the UCI Machine Learning Repository [3], Ludmila Kuncheva Repository [9] and Keel [1] for 8 classifiers with different decision paradigm. A brief description of the data sets used is given in Table 1.

### 3.1 Experimental Setup

In the first experiment three synthetic datasets with two classes were generated from normal distribution $N(\mu, \sigma)$ using different $(\mu, \sigma)$ (see Fig. 1). For generated set it was confirmed (for $p \ll 0.01$) that each sample is not statistically significantly different from normal distribution using Kolmogorow Smirnow, Lillieforsa and W Shapiro-Wilka statistical tests. For simulations, 50k and 10k objects were generated for each class for testing and validation sets, and *a'priori* class probabilities were equal. Each simulation was repeated 10 times and results were averaged. Since we deal with scalar values, we use linear classifier ($\psi_l$) which decision boundary location is determined by point $x = p$ so training procedure is not needed. Furthermore, using $\mu_{1,2}, \sigma_{1,2}$ for each model an optimal Bayes decision boundary can be calculated and set as $p_{optimal}$.

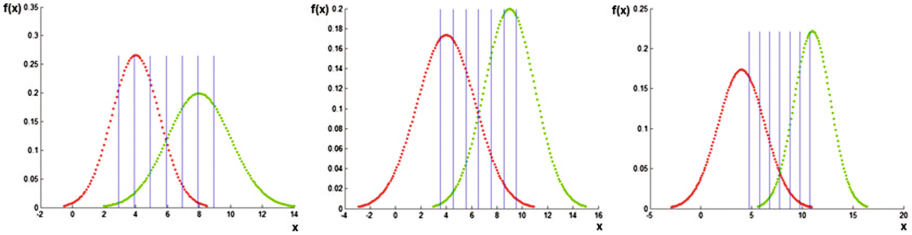


**Fig. 1.** Pdf functions for three synthetic datasets used in the first experiment. From the left: (1) $\mu_1 = 4, \sigma_1 = 1.5, \mu_2 = 8, \sigma_2 = 2.0$, (2) $\mu_1 = 4, \sigma_1 = 2.3, \mu_2 = 9, \sigma_2 = 2.0$, (3) $\mu_1 = 4, \sigma_1 = 2.3, \mu_2 = 11, \sigma_2 = 1.8$

For the second experiment the following base classifiers were used:

- **qdc** - quadratic classifier based on normal distributions with the same covariance matrix for each class,
- **nmc** - nearest mean classifier,
- **6-nn** - 6 nearest neighbors,
- **parzen** - Parzen classifier,
- **dtc** - Decision tree with information gain splitting criterion,
- **bpxn-1** - Feed-forward back-propagation neural network with 1 hidden layer and the number of learning epochs set to 120,
- **bpxn-2** - Feed-forward back-propagation neural network with 2 hidden layer and the number of learning epochs set to 120,
- **svm** - support vector machine with 3rd order polynomial kernel.

For each data set from the second experiment, feature vectors were normalized to zero mean and unit standard deviation. Simulations were conducted according to $5 \times 2$ cross-validation method for extracting training and testing sets from each data set [2]. For the comparison purposes the following approach was applied. As a reference, each base classifier was trained using whole training set, while for BMC preparation, from available training set 30% of objects were selected randomly to form validation set while the rest objects were used for base classifier fitting. All experiments were conducted in MATLAB using PRTools 5.0 packet.

**Table 1.** Datasets used in experiments

| # | Database | Source | # Objects | # Features | # Classes | *a'priori* probability |
|---|---|---|---|---|---|---|
| 1 | Weaning | LKC | 302 | 17 | 2 | .5, .5 |
| 2 | Parkinson | UCI | 195 | 22 | 2 | .75, .25 |
| 3 | Transfusion | UCI | 748 | 4 | 2 | .76, .24 |
| 4 | Wisconsin | UCI | 683 | 9 | 2 | .65, .35 |
| 5 | Dermatology | UCI | 358 | 34 | 6 | .3, .2, .2, .1, .1, .1 |
| 6 | Glass | UCI | 214 | 9 | 6 | .3, .4, .1, .1, .0, .1 |
| 7 | Haberman | UCI | 306 | 3 | 2 | .7, .3 |
| 8 | Mamographics | Keel | 830 | 5 | 2 | .5, .5 |
| 9 | Iris | UCI | 150 | 4 | 3 | $3 \times \frac{1}{3}$ |
| 10 | Laryngeal3 | LKC | 353 | 16 | 3 | .2, .6, .2 |
| 11 | Voice3 | LKC | 238 | 16 | 3 | .2, .1, .7 |
| 12 | Pima | UCI | 768 | 5 | 2 | .3, .7 |
| 13 | Thyroid | UCI | 215 | 5 | 3 | .7, .2, .1 |
| 14 | Vowel | UCI | 990 | 10 | 11 | $11 \times \frac{1}{11}$ |
| 15 | Segmentation | UCI | 2310 | 19 | 7 | $7 \times \frac{1}{7}$ |
| 16 | Yeast | UCI | 1484 | 8 | 10 | .164, .289, .312, .030, .034, .110, .024, .020, .014, .003 |
| 17 | Sonar | UCI | 208 | 60 | 2 | .5, .5 |
| 18 | OptDigits | UCI | 3823 | 64 | 10 | $10 \times \frac{1}{10}$ |
| 19 | Spam | UCI | 4601 | 57 | 2 | .4, .6 |
| 20 | Statlog | UCI | 270 | 13 | 2 | .45, .55 |
| 21 | Page blocks | UCI | 5473 | 10 | 5 | .90, .06, .01, .01, .02 |
| 22 | Phoneme | Keel | 5404 | 5 | 2 | .3, .7 |

### 3.2 Experiment Results

Results for the first simulation scenario are presented in Table 2 where 7 classification error rates are shown for $\psi_l$ with decision boundary set to: $x = p_{optimal} - 3, x = p_{optimal} - 2, x = p_{optimal} - 1, x = p_{optimal}, x = p_{optimal} + 1, x = p_{optimal} + 2, x = p_{optimal} + 3$. After analyzing results in Table 2 it can be seen that the upper bound for BMC improvement is Bayes error calculated for optimal decision boundary. Higher correction can be obtained for classifier $\psi_l$ when decision boundary is farther from the optimal. One should keep in mind, that this experiment is very simple, but it can prove that BMC works according to Bayes scheme. Table 3 shows results for base classifiers versus BMC overlay for benchmark datasest. Classification accuracies were averaged over 5 repetitions of two–fold cross–validation. We have performed pairwise and multiple comparison statistical tests. For the first one 5 × 2 CV F-test was applied to indicate which

**Table 2.** Classification error rates for the first experiment with synthetic data. Numbers: **1**, **2**, **3** denote generated datasets. Notation used in the first column is as follows: p - dividing (boundary) point, C - an analytic calculation of the Bayes probability of error for decision boundary at point $x = p$, $\psi_l$ - an average classification error rate for linear classifier with decision boundary at point $x = p$, BMC - an average classification error rate for $\psi_l$ with applied BMC scheme.

| p | $p_{optimal} - 3$ | $p_{optimal} - 2$ | $p_{optimal} - 1$ | $p_{optimal}$ | $p_{optimal} + 1$ | $p_{optimal} + 2$ | $p_{optimal} + 3$ |
|---|---|---|---|---|---|---|---|
| | **1** | | | | | | |
| C | 0.384 | 0.270 | 0.165 | 0.125 | 0.161 | 0.245 | 0.339 |
| $\psi_l$ | 0.389 | 0.271 | 0.165 | 0.121 | 0.153 | 0.238 | 0.338 |
| BMC | 0.122 | 0.124 | 0.158 | 0.121 | 0.121 | 0.122 | 0.122 |
| | **2** | | | | | | |
| C | 0.291 | 0.210 | 0.146 | 0.122 | 0.148 | 0.217 | 0.308 |
| $\psi_l$ | 0.284 | 0.208 | 0.141 | 0.119 | 0.145 | 0.211 | 0.303 |
| BMC | 0.119 | 0.120 | 0.120 | 0.119 | 0.125 | 0.120 | 0.120 |
| | **3** | | | | | | |
| C | 0.184 | 0.111 | 0.061 | 0.044 | 0.064 | 0.128 | 0.227 |
| $\psi_l$ | 0.180 | 0.108 | 0.062 | 0.044 | 0.062 | 0.128 | 0.225 |
| BMC | 0.044 | 0.044 | 0.044 | 0.044 | 0.048 | 0.044 | 0.044 |

algorithm is better. In Table 3 $+/=/-$ means that the BMC system is statistically significantly better/no statistically significant difference/statistically significantly worse, respectively. For assessing the ranks of proposed methods over all examined benchmarks, we have used Friedman ranking test and after that the Shaffer post-hoc test to check which of the tested methods are distinctive among *nxn* comparisons [7]. The level of $p < 0.05$ was considered as statistically significant. From the Friedman ranking test, which checks the overall performance on all available datatsets BMC proposal was 5 out of 8 cases statistically significantly better than a base classifier. Major improvement can be observed for problems: 1, 5, 8, 9, 13, 14, 15, 17, 18. This indicates that BMC works better with balanced datasets, while for imbalanced problems application of a'priori probabilities in BMC decision scheme causes to prefer majority class. For imbalanced problems such as: 12, 21, 22 BMC does not improve classifier accuracy. This can be explained by the fact that random sampling from training set is used to create validation set. In this case, $\mathcal{V}$ contains mainly examples from the majority class. Another important conclusion coming from this experiment is presented in Fig. 2 which shows a box plot for an exemplary laryngeal database for selected base classifier versus BMC algorithm. It indicates that apart from improving classification performance also BMC variance between subsequent cross-validation sets is decreased.

**Table 3.** Classification accuracies (in percentage) for the second experiment using different base classifiers. For each comparison, the first column presents results for base classifier trained with the whole training set and the second one contains accuracies for BMC algorithm. +/ = /− means that the BMC system is statistically significantly better/no statistically significant difference/statistically significantly worse ($p = 0.05$). F stands for Friedman ranking test checking overall performance on all databases.

| | qdc | | | nmc | | | 6-nn | | | parzen | | | dtc | | | bpxnc-1 | | | bpxnc-2 | | | svm | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 89.6 | 90.2 | = | 77.9 | 90.5 | + | 68.7 | 73.3 | + | 76.4 | 76.7 | + | 75.4 | 75.9 | = | 55.2 | 74.3 | = | 53.3 | 74.5 | = | 82.6 | 83.0 | = |
| 2 | 72.1 | 86.8 | + | 74.6 | 86.0 | + | 84.1 | 85.1 | = | 88.2 | 88.5 | + | 81.2 | 84.3 | = | 74.5 | 83.3 | = | 72.5 | 86.2 | = | 85.8 | 86.6 | = |
| 3 | 75.3 | 76.6 | = | 75.8 | 76.6 | = | 77.3 | 76.6 | = | 76.6 | 76.6 | = | 75.2 | 76.6 | = | 69.3 | 76.5 | = | 76.1 | 76.6 | = | 76.0 | 76.4 | + |
| 4 | 96.0 | 96.5 | = | 95.7 | 96.2 | = | 96.3 | 96.2 | = | 96.5 | 96.9 | + | 93.8 | 95.1 | = | 80.9 | 95.7 | = | 88.7 | 95.2 | = | 95.3 | 96.1 | = |
| 5 | 18.9 | 94.1 | + | 95.3 | 95.1 | = | 94.6 | 95.5 | + | 94.3 | 94.7 | + | 93.8 | 95.3 | = | 17.2 | 93.9 | + | 18.9 | 94.7 | + | 93.5 | 93.9 | + |
| 6 | 7.5 | 57.0 | + | 52.0 | 55.6 | = | 63.4 | 62.6 | = | 59.9 | 60.6 | = | 63.5 | 64.3 | = | 13.6 | 50.7 | + | 22.9 | 53.7 | + | 60.2 | 60.6 | = |
| 7 | 74.8 | 73.5 | = | 74.8 | 73.5 | = | 72.9 | 73.5 | = | 73.4 | 73.5 | = | 72.0 | 73.3 | = | 73.0 | 74.0 | + | 71.0 | 73.5 | = | 74.1 | 73.5 | = |
| 8 | 80.9 | 81.7 | + | 81.8 | 82.2 | + | 79.7 | 80.4 | = | 80.5 | 80.9 | + | 82.8 | 83.2 | + | 68.1 | 80.6 | = | 68.8 | 80.8 | = | 82.0 | 81.3 | = |
| 9 | 94.1 | 94.5 | + | 95.6 | 96.0 | + | 95.5 | 95.9 | + | 96.3 | 96.7 | + | 93.5 | 93.9 | + | 39.3 | 90.8 | + | 31.7 | 93.1 | + | 95.6 | 96.0 | + |
| 10 | 62.3 | 65.4 | = | 71.2 | 68.6 | = | 71.9 | 72.6 | = | 72.4 | 71.7 | = | 66.3 | 69.5 | = | 43.9 | 67.6 | = | 51.7 | 66.6 | = | 72.3 | 72.1 | = |
| 11 | 62.9 | 71.9 | = | 74.7 | 70.9 | = | 77.7 | 72.9 | = | 71.2 | 69.5 | − | 70.4 | 73.2 | = | 55.5 | 70.3 | = | 67.6 | 70.7 | = | 75.8 | 72.2 | = |
| 12 | 73.8 | 74.8 | = | 74.4 | 72.9 | = | 72.2 | 73.6 | = | 73.1 | 73.1 | = | 71.0 | 71.4 | = | 61.3 | 67.2 | = | 64.5 | 66.8 | = | 75.3 | 69.3 | − |
| 13 | 96.6 | 97.0 | + | 91.2 | 91.6 | + | 89.9 | 90.3 | + | 93.0 | 93.4 | + | 90.6 | 91.2 | = | 65.6 | 77.7 | = | 43.6 | 77.5 | + | 92.0 | 88.7 | = |
| 14 | 63.6 | 72.0 | + | 53.4 | 69.1 | + | 65.4 | 68.6 | + | 88.7 | 88.7 | = | 61.4 | 68.5 | + | 7.4 | 61.7 | + | 9.7 | 62.1 | + | 56.9 | 77.6 | + |
| 15 | 62.4 | 82.2 | = | 81.3 | 84.7 | = | 80.4 | 84.0 | = | 82.1 | 82.8 | = | 81.1 | 83.6 | = | 43.7 | 77.6 | + | 45.0 | 77.8 | + | 88.5 | 90.2 | = |
| 16 | 21.6 | 43.1 | + | 50.6 | 53.5 | = | 54.9 | 52.3 | − | 55.1 | 54.8 | = | 56.2 | 55.8 | = | 26.0 | 37.7 | = | 33.6 | 37.5 | = | 36.6 | 42.4 | = |
| 17 | 68.4 | 84.6 | + | 71.2 | 85.5 | = | 69.5 | 84.2 | + | 82.9 | 83.8 | = | 68.4 | 71.5 | + | 74.3 | 78.5 | + | 77.3 | 81.0 | + | 85.1 | 83.9 | = |
| 18 | 62.5 | 96.8 | + | 91.7 | 96.5 | + | 97.3 | 97.9 | + | 98.1 | 98.5 | + | 85.2 | 94.6 | + | 25.0 | 97.4 | + | 28.1 | 97.5 | + | 98.5 | 98.9 | + |
| 19 | 80.4 | 86.9 | + | 83.9 | 85.1 | = | 87.1 | 87.5 | = | 85.0 | 85.0 | = | 89.8 | 89.8 | = | 71.1 | 70.4 | = | 75.3 | 75.2 | = | 66.2 | 70.4 | = |
| 20 | 80.6 | 82.7 | = | 86.4 | 85.0 | = | 85.7 | 84.8 | = | 85.1 | 84.8 | = | 82.9 | 83.3 | = | 85.2 | 85.3 | = | 85.3 | 85.6 | = | 81.9 | 84.4 | = |
| 21 | 83.4 | 91.4 | = | 74.5 | 90.5 | + | 94.9 | 90.9 | − | 94.1 | 91.5 | − | 96.6 | 93.2 | = | 89.8 | 89.9 | = | 89.8 | 89.9 | = | 94.6 | 93.5 | = |
| 22 | 75.9 | 70.7 | − | 73.5 | 70.7 | − | 84.4 | 84.4 | = | 85.5 | 85.5 | = | 83.5 | 83.5 | = | 74.6 | 71.1 | = | 77.0 | 71.5 | − | 78.6 | 70.7 | − |
| F | + | | 10 | = | | 6 | = | | 5 | + | | 6 | + | | 5 | + | | 8 | + | | 7 | = | | 3 |

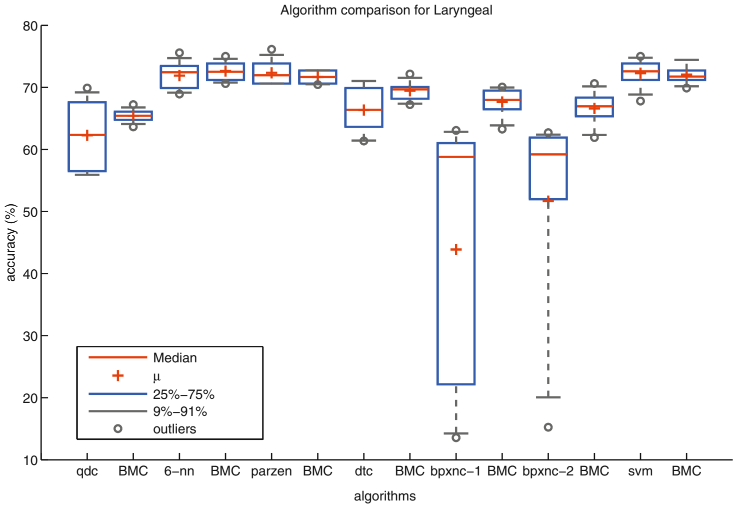


**Fig. 2.** Box plot for laryngeal dataset for selected base classifier versus BMC algorithm.

## 4 Conclusions

In this paper a detailed BMC algorithm construction was presented and validated. The first experiment confirmed that the upper bound of BMC improvement over base classifier is the Bayes error. This is not very suprising since BMC works according to Bayes scheme maximizing *a'posteriori* probabilities for decision of a base classifier. From three synthetic dataset analysis it is straightforward that BMC performance directly depends on the classification quality of a base classifier $\psi$. Further experiment on benchmark datasets shows that BMC can significantly correct base classifier classification, especially for cases where base classifier is not optimal. Overall performance according to Friedman ranking test indicates that BMC proposal was statistically significantly better 5 out of 8 different classifiers. Additionally, when analyzing results from each cross-validation phase, it is clearly visible that BMC decreases classification variance comparing to its base counterpart results. On the other hand, the main drawback of BMC involves proper validation set generation. BMC works better with balanced datasets, while for imbalanced problems usage of *a'priori* probabilities in BMC design, causes that its decision boundary is moved towards majority class. To sum up, since BMC provides probabilistic interpretation for a base classifier response for correct and incorrect classification this method can be directly used in a sequential classification problems, or in MC systems, especially during classifier fusion. As a future work, we would like to apply BMC for ensemble construction at both crisp and continuous level of classifier response.

**Acknowledgments.** This work was supported by the statutory funds of the Department of Systems and Computer Networks, Wroclaw University of Science and Technology.

## References

1. Alcala-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., Garcia, S., Sanchez, L., Herrera, F.: Keel data-mining software tool: data set repository. J. Multiple-Valued Logic Soft Comput. **17**, 255–287 (2011)
2. Alpaydin, E.: Combined 5 × 2 cv F test for comparing supervised classification learning algorithms. J. Neural Comput. **11**, 1885–1892 (1999)
3. Asuncion, A., Newman, D.: UCI machine learning repository (2007). http://www.ics.uci.edu/mlearn/MLRepository.html
4. Duda, R., Hart, P., Stork, D.: Pattern Classification, 3rd edn. Wiley-Interscience, New York (2001)
5. Freund, Y.: Boosting a weak learning algorithm by majority. Inf. Comput. **121**(2), 256–285 (1995)
6. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press Professional Inc., San Diego (1990)
7. Garcia, S., Fernandez, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Inf. Sci. **180**, 2044–2064 (2010)

8. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms, 3rd edn. Wiley-Interscience, New York (2004)
9. Kuncheva, L.: Real medical data sets repository (2004). http://pages.bangor.ac.uk/~mas00a/activities/real_data.htm
10. Kurzynski, M., Majak, M.: Meta-Bayes classifier with Markov model applied to the control of bioprosthetic hand. In: Czarnowski, I., Caballero, A.M., Howlett, R.J., Jain, L.C. (eds.) Intelligent Decision Technologies 2016. SIST, vol. 57, pp. 107–117. Springer, Cham (2016). doi:10.1007/978-3-319-39627-9_10
11. Kurzynski, M., Majak, M., Zolnierek, A.: Multiclassifier systems applied to the computer-aided sequential medical diagnosis. J. Biocybern. Biomed. Eng. **36**, 619–625 (2016)
12. Skurichina, M., Duin, R.P.W.: Bagging, boosting and the random subspace method for linear classifiers. Pattern Anal. Appl. **5**(2), 121–135 (2002)
13. Skurichina, M., Duin, R.P.W.: Bagging and the random subspace method for redundant feature spaces. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 1–10. Springer, Heidelberg (2001). doi:10.1007/3-540-48219-9_1
14. Strunk Jr., W., White, E.B.: Statistical Decision Theory and Bayesian Analysis, 3rd edn. Springer, New York (1987)
15. Woloszynski, T.: Classifier competence based on probabilistic modeling (ccprmod.m) (2013)
16. Woloszynski, T., Kurzynski, M.: A probabilistic model of classifier competence for dynamic ensemble selection. Pattern Recogn. **44**, 2656–2668 (2011)
17. Woloszynski, T., Kurzynski, M.: A measure of competence based on random classification for dynamic ensemble selection. Inf. Fusion **13**, 207–213 (2012)
18. Wolpert, D.: Stacked generalization. Neural Netw. **5**, 241–259 (1992)
19. Wozniak, M., Grana, M., Corchado, E.: A survey of multiple classifier systems as hybrid systems. Inf. Fusion **16**, 3–17 (2014)

# Recent Advances in Image Pre-processing Methods for Palmprint Biometrics

Agata Wojciechowska(✉), Michał Choraś, and Rafał Kozik

Faculty of Telecommunications, Computer Science and Electrical Engineering,
University of Science and Technology, Bydgoszcz, Poland
agata.wojciechowska@utp.edu.pl

**Abstract.** Biometric identification may be used in real-life applications like security, forensic and common smartphones. However to ensure the robustness of biometric methods, the proper pre-processing method has to be applied. In this paper we focus only on this part of the whole recognition process. Finding an appropriate method is crucial, especially when the identification system is dedicated to a mobile scenario. An image acquired by a smartphone may have lower quality, be more defected by noises and blurred. In this paper we discuss only palmprint as a feature that may distinguish people. It is used in identification systems but still is rarely implemented in mobile.

**Keywords:** Biometrics · Palmprint recognition · Image pre-processing

## 1 Introduction

A human identification, as a part of an image processing, has been developed for more than 40 years [1], and still is one of the most demanding tasks in computer vision community. Nowadays, face, iris of the eye, fingerprints, finger knuckles and palm prints are analysed in order to perform the recognition. There are also some behavioural methods where gait, typing or mouse actions are investigated. The bio-metric identification is now implemented in various applications such as security, forensics or in common smartphones. It is worth emphasizing that thanks to effective pre-processing methods, features may be extracted and calculated even if the resolution and quality of the input image is low. In this paper we focus on image pre-processing methods applied to palmprint biometrics. The paper is organized as follows: in Sect. 2 the comparison between palm print and other biometric features is presented. Section 3 contains the description of pre-processing with the most commonly implemented methods. Conclusions are provided in afterwards.

## 2 General Characteristics of Palmprint Biometric

A good identification system has to fulfil the following requirements [2,3]:

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_28

1. universality system is able to identify each person;
2. uniqueness system gives only one identification for one person;
3. permanence the identification cannot change;
4. performance the accuracy and robustness is as high as possible while the execution time is relatively short;
5. acceptability the identification method is user-friendly and accepted by people in their daily life;
6. circumvention system is not easy to defraud.

Palmprints can meet all the above-mentioned expectations [4]. First of all, palm prints are unique. They are formed between 3 and 5 month of pregnancy [5]. It is not possible to find any similarity even in case of twins. Moreover, image of a palm has a rich texture. It contains different ridges, creases, lines, fingers ends, points between fingers and others [3,6]. It gives a big chance to detect many key points and possibly simplifies the identification. The structure of a palmprint remains static during the whole person life, except for the change in size [7]. Depending on the specific application a high or a low image resolution may be used for the identification [8]. For a forensic purpose the resolution has to be at least 400 ppi, while for a civilian purpose it may be only 100 ppi. Some previous papers present new dedicated devices that may obtain the palm print for the recognition. They use a thermal map [9] or a common web camera [6]. However, in a non-forensic application any specific device is needed [10]. It is even possible to use a mobile phone for acquiring the image [3]. On the other

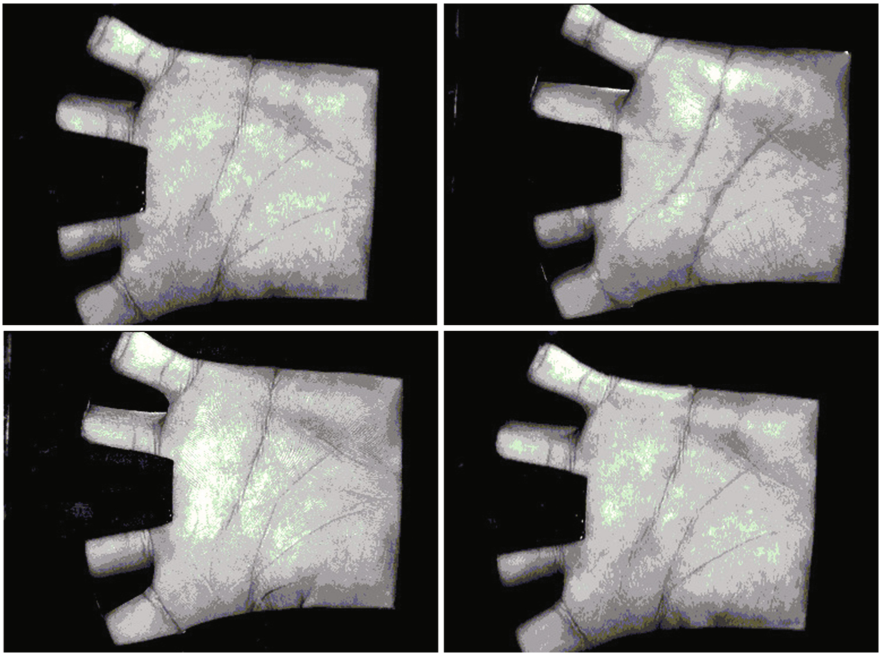


**Fig. 1.** Palmprint image examples from the PolyU Database

side, to obtain an image of the human iris, an advanced scanner is essential [11]. Thus, thanks to using the palmprint for the identification, it is possible to reduce costs of the whole recognition system. Last but not least aspect to mention is non-intrusiveness. The system is more user-friendly when the investigated person does not has to touch any device or stand in a certain position in front of the camera. The palmprint recognition is also more robust to different illumination conditions, which make e.g. the face recognition extremely challenging [12]. Most scientific papers use well-known databases to ensure that the results are reliable and comparable. The example of such a common used database is the PolyU Palmprint Database which is available online [13]. Images coming from this database are presented in the Fig. 1. The PolyU provides palmprints taken in different illumination conditions, from 250 volunteers in age range from 20 to 60. It contains images of left and right hand as well.

## 3 Palmprint Pre-processing Methods

The main steps of the palmprint recognition algorithm is presented in Fig. 2. Then, in this section, selected pre-processing methods are overviewed.

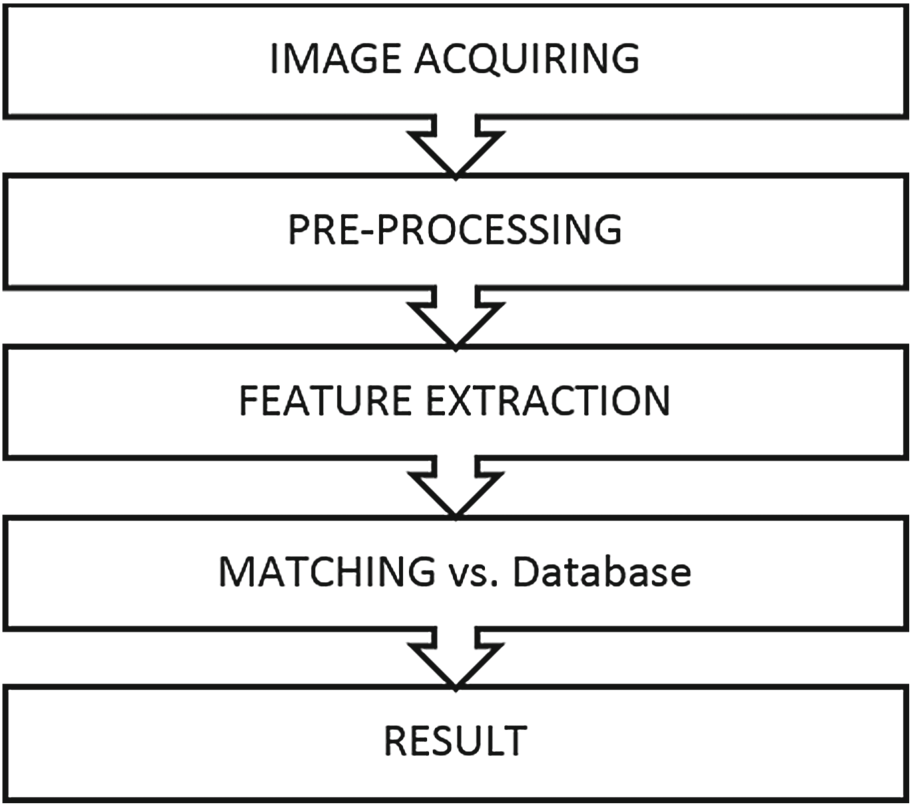


**Fig. 2.** The general scheme of the palmprint recognition systems

### 3.1 Color Detection

The other approach to distinguish a hand is a color detection [3]. Skin has the specific color that may be presented as RGB values (Red, Green, Blue). Skin is detected when values fulfil the conditions expressed in the Eq. 1.

$$\begin{aligned} R > 95\ AND\ G > 40\ AND\ B > 20\ AND \\ \max(R, G, B) - \min(R, G, B) > 15\ AND \\ |R - G| > 15\ AND \\ R > G\ AND\ R > B \end{aligned} \tag{1}$$

### 3.2 Edges Detection

There are many methods that enable edges detection: Roberts operator, Sobel operator [14], Grad operator, Prewitt operator, Log operator, Laplacian operator, Canny operator [7] and others. Examples of palmprints after the edges detection are presented in Fig. 3. Detected edges are used to distinguish the background, find a ROI and to extract key points.

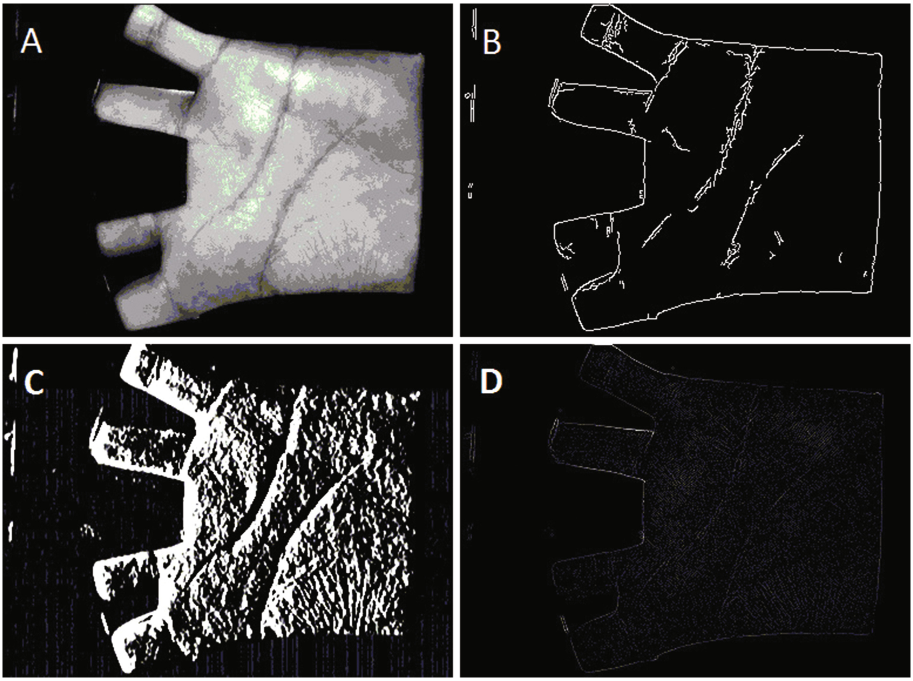


**Fig. 3.** An original image form the PolyU database (A), an image with the Canny operator performed (B), an image with the Sobel operator performed (C), an image with the Laplacian operator performed (D)

### 3.3 Smoothing

The edge detection has to be preceded by smoothing. Among the smoothing methods a Gaussian smoothing and a median filter may be listed. Blurring is essential in a noise reduction. Moreover, the more smooth the image is, the less edges are detected. The Gaussian blur is especially needed to get rid of the Gaussian noise [15]. It is expressed in the Eq. 2, where x, y are the distance along both axis and $\sigma$ is the standard deviation:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{\frac{-x^2+y^2}{\sigma^2}} \tag{2}$$

The median filter considers the value of each pixel and check whether the value suits to its closest neighborhood. If it does not, the filter replaces the value with a median. The median is calculated separately for each block of the image. The median filter is especially useful in the salt and pepper noise reduction [16].

### 3.4 Thresholding

Thresholding is the simplest method of an image segmentation. It is used to separate the region of interests from the background. The main idea of thresholding is to compare each pixel value with the constant value. Then, depending on this comparison, it gives the binary value (1 or 0). However, the basic method is not robust to noise and changing background. Thus, there are also many adopted methods that perform thresholding like Otsu, Savuola. Savuola binarization used by Jaafar et al. [7] is the local method. It analyses the contrast in a specific tile of the image. If the contrast is high, the threshold value is set close to mean. Otherwise, if the contrast is low, the value is set below mean proportional to the local standard deviation. Otsu method used by Harb et al. [15] is based on the histogram calculation. It divides the whole image to two classes: a foreground and a background. The threshold value is set so that it may separate before mentioned classes in the best possible way. It is performed by minimalizing the inner-class variation.

### 3.5 Size Filter

The size filtering is the approach that may substitute the edges detection. It requires the thresholding before the filtering. Then the biggest area is chosen and set as ROI. However, on the real image the segmentation may be inaccurate due to rings, watches or other unexpected elements. Hence, as a part of a post-processing, the ring artifact removal algorithm was proposed by Yoruk et al. [17]. This algorithm enables re-building unknown parts of fingers or wrist.

### 3.6 Gabor Filters

Gabor filter is widely used in the image processing and also for palmprint preprocessing. It is represented in the Eq. 3.

$$G(x, y) = \frac{1}{2\pi\sigma^2} exp\{\frac{-x^2+y^2}{2\sigma^2}\} exp\{2\pi i(ux\cos\theta + uy\sin\theta)\} \tag{3}$$

where $i = \sqrt{-1}$ is the orientation of the function, $\sigma$ is the standard deviation and u is the frequency of the sinusoidal wave. The Gabor filter and its modifications (2D Gabor filter, Log-Gabor filter) have been implemented by Jain et al. to enhance the palmprint image [8], by Bala et al. to reduce an unwanted noise [18] and by Zhang et al. to extract features [1,19,20]. In spite of being commonly used, it has some disadvantages like: a heavy computation burden, a need for a lot of memory to store Gabor features and a lack of robustness to varying illumination condition [21].

### 3.7 Anisotropic Filter

Due to existing drawbacks of the Gabor filter, Li et al. used the the Anisotropic Filter (AF) instead [21]. It is expressed in the Eq. 4.

$$G(u,v) = (4u^2 - 2)exp(-(u^2 + v^v)) \tag{4}$$

where parameters u and v are described by the Eq. 5.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1/\alpha & 0 \\ 0 & 1/\beta \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x - x_o \\ y - y_0 \end{bmatrix} \tag{5}$$

in which $(x_0, y_0)$, $\theta$, $\alpha$ and $\beta$ are a center of the Anisotropic Filter, a rotation and scaling parameters along the X-axis and Y-axis, respectively. The AF is robust to drastic illumination changes and preserves important discriminative information.

### 3.8 Haar Wavelet

Haar wavelet (Db1) is the simplest and the fastest wavelet. It contains only two co-efficients: the first to perform the low pass filter and the second to perform the high pass filter. Hence, using the Haar wavelet may lead to reduce a computation cost. Basically, it is expressed in the Eq. 6.

$$\psi(t) = \begin{cases} 1 & 0 \leq t < 1/2 \\ -1 & 1/2 \leq t < 1 \\ 0 & otherwise \end{cases} \tag{6}$$

Tamrakar et al. used the Haar wavelet to find an AROI instead a ROI [22]. The AROI is a region four-times smaller than the ROI. In that way, the computation complexion is reduced. The solution is also invariant to noise thanks to band pass filtering.

## 4 Conclusions

In this paper we have presented the short overview of image pre-processing methods for palmprint biometrics. Efficient pre-processing methods are particularly necessary in mobile biometrics scenarios (e.g. access by mobile devices).

In such a case, the input images can be of low quality, and the task of the pre-processing methods is to enhance them before the feature extraction step. From our analysis, the trend now is towards using transformation and filtering based methods, and Gabor filters in particular. Therefore, in our future work, we plan to implement well-tuned advanced filtering methods as the pre-processing step for contactless palmprint recognition for mobile scenario.

## References

1. Zhang, D., Kong, W.-K., You, J., Wong, M.: Online palmprint identification. IEEE Trans. Pattern Anal. Mach. Intell. **25**, 1041–1050 (2003)
2. Hong, L., Jain, A.: Integrating faces and fingerprints for personal identification. IEEE Trans. Pattern Anal. Mach. Intell. **20**, 1295–1307 (1998)
3. Choraś, M., Kozik, R.: Contactless palmprint and knuckle biometrics for mobile devices. Pattern Anal. Appl. **15**, 73–85 (2012)
4. Tabejamaat, M.: Selective Algorithm Outline (SAO): an alternative approach for fusing different palm-print recognition algorithms. Neural Process. Lett. **43**, 709–726 (2015)
5. Dubey, P., Kanumuri, T.: Optimal local direction binary pattern based palmprint recognition. In: 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1979–1984. IEEE (2015)
6. Jadhav, S.B., Raut, M.S.D., Humbe, V.T., Kartheeswaran, T.: A low-cost contactless palm print device to recognize person based on texture measurement (2016)
7. Jaafar, H., Ibrahim, S., Ramli, D.A.: A robust and fast computation touchless palm print recognition system using LHEAT and the IFkNCN classifier. Comput. Intell. Neurosci. **2015**, 117 (2015)
8. Jain, A.K.: Jianjiang Feng: latent palmprint matching. IEEE Trans. Pattern Anal. Mach. Intell. **31**, 1032–1047 (2009)
9. Czajka, A., Bulwan, P.: Biometric verification based on hand thermal images. In: 2013 International Conference on Biometrics (ICB), p. 16. IEEE (2013)
10. Kong, W.K., Zhang, D.: Palmprint texture analysis based on low-resolution images for personal authentication. In: Proceedings of 16th International Conference on Pattern Recognition 2002, pp. 807–810. IEEE (2002)
11. Aishwarya, D., Gowri, M., Saranya, R.K.: Palm print recognition using liveness detection technique. In: Second International Conference on Science Technology Engineering and Management (ICONSTEM), pp. 109–114. IEEE (2016)
12. Mudunuri, S.P., Biswas, S.: Low resolution face recognition across variations in pose and illumination. IEEE Trans. Pattern Anal. Mach. Intell. **38**, 1034–1040 (2016)
13. The Hong Kong Polytechnic University (PolyU) Palmprint Database. http://www4.comp.polyu.edu.hk/biometrics/
14. Ray, K.B., Misra, R.: Palm print recognition using hough transforms, December 2015
15. Harb, A., Abbas, M., Cherry, A., Jaber, H., Ayache, M.: Palm print recognition. In: 2015 International Conference on Advances in Biomedical Engineering (ICABME), pp. 13–16. IEEE (2015)
16. Patil, J.P., Nayak, C., Jain, M.: Palmprint recognition using DWT, DCT and PCA techniques. In: 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), p. 15. IEEE (2015)

17. Yoruk, E., Konukoglu, E., Sankur, B., Darbon, J.: Shape-based hand recognition. IEEE Trans. Image Process. **15**, 1803–1815 (2006)
18. Bala, S.: Nidhi: comparative analysis of palm print recognition system with repeated line tracking method. Procedia Comput. Sci. **92**, 578–582 (2016)
19. Jaswal, G., Nath, R., Kaul, A.: Texture based palm Print recognition using 2-D Gabor filter and sub space approaches. In: 2015 International Conference on Signal Processing, Computing and Control (ISPCC), pp. 344–349. IEEE (2015)
20. Ota, H., Aoyama, S., Watanabe, R., Ito, K., Miyake, Y., Aoki, T.: Implementation and evaluation of a remote authentication system using touchless palmprint recognition. Multimed. Syst. **19**, 117–129 (2013)
21. Li, H., Zhang, J., Wang, L.: Robust palmprint identification based on directional representations and compressed sensing. Multimed. Tools Appl. **70**, 2331–2345 (2014)
22. Tamrakar, D., Khanna, P.: Kernel discriminant analysis of block-wise Gaussian derivative phase pattern histogram for palmprint recognition. J. Vis. Commun. Image Represent. **40**, 432–448 (2016)

# Some Properties of Consensus-Based Classification

Vitaliy Tayanov(✉), Adam Krzyżak, and Ching Suen

Department of Computer Science and Software Engineering,
Concordia University, Montreal H3G 1M8, Canada
{tayanov,krzyzak,suen}@encs.concordia.ca

**Abstract.** The objective of this paper is to consider some properties of decisions produced by classifiers that are in consensus. Consensus allows strong classifiers to obtain very reliable classification on the objects on which consensus has been reached. For those ones where consensus is not reached the reclassification procedure should be applied based on other classification algorithms. Properties of different consensuses are described using algebraic approach and performance evaluation routine.

## 1 Introduction

For any classification algorithm it is very important to satisfy the following three conditions: high performance, good generalisation ability and low computational cost. First condition might be satisfied using advanced classifiers like support vector machines (SVM), decision trees, neural networks, etc. or their combinations [9,11,18]. Generalisation ability is connected with Statistical Machine Learning Theory firstly developed by Vapnik and Chervonenkis [13,14]. Generalisation ability tells us about prediction ability on future examples. In general theory tells how good learning can be and if it is possible to learn at all. Vapnik's bounds are very loose and that is why many improvements have been introduced [16]. To reduce overfitting effect one might mention bagging of boosted week classifiers [5], e.g., AdaBoost [4] or splitting classifiers by similarity and error rate they produce [15]. However the question is how to combine classifiers to obtain the most reliable labeling in terms of error and low overfitting effects. It is very important to understand that there exists a probability of classification error for every sample. It can be evaluated both by different schemes of cross-validation [9] or using particular algorithms for concrete type of generative or discriminative models. For k-NN classifiers these estimates might be obtained using distribution of distances from the target pattern to all other patterns [2,12]. For discriminative models it can be evaluated using notation of potentials which are proportional to the level of margin for that probe. The concept is similar to the approach used for the Markov Random Fields (MRF) [8]. This can be done similarly for generative models as well. The quality of generalisation depends on the number of patterns that are near separating hyperplane ('hard' or 'ambiguous' patterns) [1,6]. They have large

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_29

classification error. The idea proposed in this paper is to postpone classification of these patterns to the second stage where they are classified by other algorithms. Application of random classifiers to heavy overlapping classes may be a good idea. To detect such ‘ambiguous’ patterns one may use consensus-based association of classifiers. The proposed algorithm for combining classifiers addresses the two problems mentioned above. Combination or fusion can be done at different levels of classification process [17]. However here one is interested in combining the results of classification (decision level) which are represented as binary classification problems. First work devoted to usage of combination in a way of consensus for combining clustering results in satellite imaging was proposed in [10]. The present work determines algebraic operations of co-association and allows to estimate some classification errors using consensus. One of the key points of the proposed approach is to divide objects in functional groups with further reclassification of objects from the group where consensus is not reached (‘ambiguous’ objects).

## 2 Algebraic Approach

### 2.1 Algebra on Classification Results

In this subsection we are going to consider algebraic operations on classification results produced by different strong classifiers. Some of these results will be combined to create a consensus of classifiers. The issue concerning the decision which classification results should be combined will be considered in the following sections. Because algebraic operations are applied to classification results they are defined on lattice with three elements: 1 (positive decision), 0 (negative decision) and $\Delta$ (no decision is made). Such abstract algebraic operations were first considered in [19] where the author introduced an algebraic operation of addition of binary classification results ($\oplus$ algebraic operation). Consensus of classifiers corresponds to an abstract multiplication operation of classification results ($\otimes$ algebraic operation). One demands a set of four algebraic laws both for addition and multiplication operations to be respected:

- associative law;
- preserving of element law: $a \oplus a = a$ and $a \otimes a = a$;
- preserving of the set law: $a \oplus \Delta \in \{a, \Delta\}$ and $a \otimes \Delta \in \{a, \Delta\}$;
- commutative law.

Let us give general rules on how to apply consensus operation on the lattice with three elements $a$, $b$ and $c$ such that $a \neq b$ and $c = a \otimes b$. So the result of operation of taking consensus belongs to a set $\{a, b, c\}$:

$$\begin{aligned}
\otimes(a,a) = a \otimes a = a;\\
\otimes(a,b) = a \otimes b = b \otimes a = c;\\
\otimes(c,c) = c \otimes c = c;\\
\otimes(a,c) = a \otimes c = c \otimes a = \{a,c\};\\
\otimes(a,a,b) = (a \otimes a) \otimes b = c \neq\\
(a \otimes b) \otimes a = a \otimes c = \{a,c\};\\
\otimes(a,a,b,b) = (a \otimes a) \otimes (b \otimes b) =\\
(a \otimes b) \otimes (a \otimes b) = (a \otimes b) = c;\\
\otimes(a,a,a,b) = (a \otimes a) \otimes (a \otimes b) = a \otimes c = \{a,c\}.
\end{aligned} \tag{1}$$

As seen any consensus with $n$ experts can be divided into consensus of consensuses with two experts or consensus of consensus with two experts and one separate expert. This is important property which allows to use binary co-association matrixes for any number of experts in consensus. In (1) one applies all four algebraic laws introduced before. However in some cases the order of operations is important and that is why operations might also not form a group or even a semi-group. Majority voting is a particular case of generalised consensus:

$$a \otimes c = a \otimes (a \otimes b) = a. \tag{2}$$

Applying consensus of experts to the objects where consensus was not reached gives the possibility to make a decision if consensus will be reached using another subset of classifiers which are distinct from the previous subset of experts or using a majority voting rule. Thus corresponding algebraic operations look as follows:

$$c \otimes (a \otimes a) = (a \otimes b) \otimes (a \otimes a) = a. \tag{3}$$

In case of using multiple experts in consensus in one time to reach a consensus all of them should be in consensus. In this case algebraic operation on them and the outcome it produces has the following form:

$$a \otimes c = a \otimes a \otimes b = c. \tag{4}$$

In case of classification let us put $a = 1$ (decision is positive), $b = 0$ (decision is negative) and $c = \Delta$ (situation when consensus is not achieved). In case of odd number of classifiers two options are possible: consensus is achieved or majority voting is applied. In case of even number of classifiers when the number of classifiers is more than two, three options are possible: consensus, majority voting or no consensus. In the latter case decision might be made by random classification. In other words if one adds one more classifier to the set of two classifiers where consensus was not reached we can classify this by majority voting or combine two classifiers to reach a consensus which consists of two classifiers. This is how majority voting works in terms of consensus. Another option is to reclassify the probe by another set of two classifiers that are in consensus. These are practical issues of using the algebra given above. It is possible to create consensuses with different bases. By a base of consensus one denotes the number of experts used in it. So minimal consensus has a base equal to 2. Consensus of arbitrary number of experts can be presented as consensus of consensuses with base equal to 2. This allows using elementary binary co-association matrixes to present a consensus with arbitrary base. All this allows to build very different consensuses some of which might produce very reliable decisions. Applying several consecutive consensuses that are also under the operation of consensus might serve for two different objectives: first one is to reclassify objects where consensus is not reached and the second one is to make consensus stronger. Besides of the fact that mathematically consecutive consensuses look similarly to one big consensus where all experts are under consensus they might have completely different objectives.

### 2.2 Elementary Binary Co-association Matrices for Classification Results

Assume that there is a set of strong classifiers $F$, so that any strong classifier belongs to this set, e.g. $f_i \in F, i = 1, ...n$, where $n$ is a number of strong classifiers in the set, so that $|F| = n$. Here one considers classification into $N$ classes. Let $I$ be the number of objects in the set and $P$ be the number of classification results. Every classification $p$ $(p = 1, ..., P)$ associates observation $k$ from the sample with one and only one class. Elementary co-association matrix $\mathbf{A}^k$ contains the information according to which algorithms $u$ and $v$ have consensus or not with respect to some class label $i$ for probe $k$:

$$a^k_{u,v} = \begin{cases} 1, \text{ for } u \equiv v \\ 0, \text{ otherwise} \end{cases} \tag{5}$$

where $\equiv$ denotes consensus between classifiers $f_u$ and $f_v$. Because $u \equiv v \Rightarrow v \equiv u$, $\mathbf{A}^k$ is an $n \times n$ symmetric binary matrix. The number of different consensuses that could be created is equal to $P = \binom{n}{2} = \frac{n(n-1)}{2}$. From the entire set of possible consensuses equal to $P$ one needs to take two of them that are maximally dissimilar in terms of Hamming distance between binary results of classification for objects $k \in I$ of the training sample of the set $\chi$. Finding maximal Hamming distance between classifiers spans entire classifier space $F$. This allows to consider all possible combinations between classifiers and use all possible Hamming distances between classifiers which is important during optimisation of consensus. Results of classification are represented as binary vector $\mathrm{v}^p$ that consists of zeros and ones. The elements of vector p are taken from the set of matrices $\mathrm{A}^k$:

$$v_k = a^k_{u,v}. \tag{6}$$

The size of vector $\mathrm{v}^p$ is equal to the number of objects $I$. Then tensor A has size $n \times n \times I$. Now the only task left to do is to determine the pair of classifiers that yield the maximal Hamming distance. Determination of appropriate indices could be done by the following procedure

$$\Big\{i, j\Big\} = \arg\min_{u,v} \sum_k A^k_{u,v}. \tag{7}$$

To estimate the pair of the most dissimilar classifiers one needs to use some training set. That is why solution (7) is approximate.

## 3 Consensus Performance Evaluation

Let us assume that there are two independent classifiers with probability of errors $\epsilon_1^k$ and $\epsilon_2^k$ for every particular probe pattern $k$. If one combines them in a way of consensus then the probability of error of labeling is equal to

$$\epsilon^k = \epsilon_1^k \times \epsilon_2^k. \tag{8}$$

The question is when one can apply formula (8). Theory says that experts should be independent in terms of errors they produce. One assumes that for larger independence experts should be as dissimilar as possible with respect to binary classification results they produce. This can be measured using Hamming distance. The simplest consensus model requires (in terms of (8)) two strong classifiers from the set $F$: $F \subset A(X \to Y)$. Using more classifiers can lead to less errors defined by (8). However in this case classifiers are less independent and assumption of independence is not accurate to use (8). On the other hand using additional classifiers leads to general augmentation of complexity of co-association and number of probes where consensus was not reached. It is important to note that if it is possible to estimate the probability of classification error for every particular probe then the complex routine of cross-validation might be avoided. As it was mentioned before some methods presented in [2,7,12] can be applied. In general for different types of classification algorithms these methods need some particular development. In general case the lower bound on classification error for consensus $A$ of two algorithms $A = A_1 \otimes A_2$ is equal to 0 (in case of full anti-correlation). However in case if errors produced by every algorithm are independent and identically distributed random variables then the error of consensus can be bounded by their product. Taking expectation using all instances we obtain

$$\inf\Big(\epsilon(A)\Big) = \mathbb{E}(\epsilon(A_1) \times \epsilon(A_2)) = \mathbb{E}(\epsilon(A_1)) \times \mathbb{E}(\epsilon(A_2)). \tag{9}$$

Then inequality for the error of consensus can be written in the following form:

$$\mathbb{E}(\epsilon(A_1)) \times \mathbb{E}(\epsilon(A_2)) \leq \epsilon(A) \leq \min(\epsilon(A_1), \epsilon(A_2)). \tag{10}$$

or in case of ant-correlation left bound is equal to 0:

$$0 \leq \epsilon(A) \leq \min(\epsilon(A_1), \epsilon(A_2)). \tag{11}$$

Total classification error probability for consensus of classifiers is equal to

$$\epsilon_t(A) = \epsilon_c(A) + \alpha\epsilon_{nc}(A), \tag{12}$$

where $\epsilon_c(A)$ is a probability of erroneous consensus, $\epsilon_{nc}(A)$ is a probability when consensus is not reached and $\alpha$ is a portion of errors after reclassification.

In the case of random classifier $\alpha = 0.5$. According to [15] using the most dissimilar classifiers for classifier combining leads to larger overtraining. Training of classifiers on difficult probes leads to high diversity in classification results because this produces diverse classifiers. To solve this problem one proposes to use the co-association scheme of combining classifiers. In this case the overtraining effect should be reduced due to the group of objects where consensus is not reached and which are reclassified with random classifier having constant error rate. For classification into arbitrary number of classes these consensuses divide our set onto three functional groups, i.e., a group of objects on which consensus of two algorithms is reached and it is correct ($\Im_1$), a group of objects on which consensus is not reached ($\Im_2$) and the group of objects on which consensus is reached and is incorrect ($\Im_3$). Obviously $\Im_j \subseteq I, j = 1, 2, 3$. Probability estimates for each group will be $\hat{P}(\Im_1)$, $\hat{P}(\Im_2)$ and $\hat{P}(\Im_3)$, respectively. The number of objects in the third group cannot be reduced. Thus total error is bounded as follows:

$$\hat{P}(\Im_3) \leq \hat{\epsilon}_t \leq \hat{P}(\Im_3) + \alpha\hat{P}(\Im_2). \tag{13}$$

## 4 Experiments

We used 5 data sets from UCI repository [3] as indicated in Tables 1 and 2. Three of them ('pima', 'bupa' and 'heart') belong to medicine and two others ('german' and 'sonar') are not medical data sets. Pima Indians Diabetes Database is a database of patients having diabetes or not. So the diagnostic is a binary-valued variable. There are 8 attributes which represent some blood test results. Total number of instances is equal to 768. BUPA liver disorders data set is also binary-valued with binary output indicating whether patients have a diabetes or not. The number of attribute is equal to 6 and they correspond to blood test analysis results with additional parameter describing the amount of alcohol consumed by a person per day. The number of examples for this data set is equal to 345. The heart disease data set consists of several subsets. The most popular is Cleveland set which consists of 303 patients each assigned 76 attributes. In practice 14 of them are used. The last is the prediction binary value whether patient has a heart disease or not. First of non-medical tasks is the German credit data. This data set has two versions with symbolic and categorical attributes. We use categorical version of the data set. The number of instances (customers) is equal to 1000 and the number of attributes is equal to 24 with additional binary label describing whether customer is "good" or "bad". The last data set consists of sonar signals reflected from rocks or mines (metallic cylinders). So the problem is to distinguish between them having the parameters of the reflected sonar signal. Total number of examples is equal to 208 and each pattern is characterised by 60 measurement attributes. The problem is binary as we have to distinguish between rocks and mines. All these five data sets are used in different tests of machine learning algorithms. Regardless that the first three data set belong to medical domain, they are described by completely different features and these

**Table 1.** Accuracy of classification for different combination algorithms: decision trees

| Method /task | 'pima' | 'bupa' | 'heart' | 'german' | 'sonar' |
|---|---|---|---|---|---|
| Bagging | 75.4 | 68.7 | 78.7 | 75.6 | 79.3 |
| AdaBoost | 69.8 | 65.8 | 78.1 | 73.3 | 79.8 |
| Random subspace | 73.6 | 65.8 | 80.5 | 73.9 | 80.3 |
| Rotation forest | 75.4 | 69.5 | 79.4 | 77.0 | 81.8 |
| Random oracle | 77.2 | 69.3 | 78.1 | 74.4 | 78.4 |
| Proposed (NB) | 78.6 | 69.6 | 83.6 | 71.2 | 73.7 |
| Proposed (LR) | 78.6 | 60.3 | 76.9 | 68.0 | 82.2 |
| Proposed (AB) | **86.0** | **84.3** | **88.2** | **81.3** | **92.4** |

**Table 2.** Accuracy of classification for different combination algorithms: linear classifiers

| Method /task | 'pima' | 'bupa' | 'heart' | 'german' | 'sonar' |
|---|---|---|---|---|---|
| Bagging | 76.8 | 67.8 | 65.3 | 77.7 | 74.5 |
| AdaBoost | 77.3 | 68.6 | 69.4 | 76.7 | 73.6 |
| Random subspace | 75.9 | 59.7 | 80.1 | 74.1 | 78.0 |
| Rotation forest | 63.4 | 67.8 | 63.6 | 76.9 | 73.6 |
| Random oracle | 76.0 | 73.0 | 67.7 | 76.4 | 78.9 |
| Proposed (NB) | 78.6 | 69.6 | 83.6 | 71.2 | 73.7 |
| Proposed (LR) | 78.6 | 60.3 | 76.9 | 68.0 | 82.2 |
| Proposed (AB) | **86.0** | **84.3** | **88.2** | **81.3** | **92.4** |

sets of features have different discrimination power. This means that they have different data complexity in terms of [1] which is very suitable to demonstrate the power of the proposed approach.

To build a consensus one uses a set of k-NN classifiers with different $L_p$ norms as dissimilarity functions. We changed $p$ in $L_p$ norm from 0.1 to 5 with a step-size of 0.1. Value of $k$ was changed from 1 to 30 using only odd values of k nearest neighbours. So the entire pool consists of 800 k-NN classifiers. Classification results are compared to those obtained by the most popular ensembles of classifiers or their modifications e.g. bagging, boosting, random forests, random subspace and random oracle. In the first scenario (Table 1) decision tree has been used as a week classifier and in the second scenario (Table 2) we used linear classifier. The accuracy of these five ensembles of classifiers is compared with three versions of the developed algorithm for each of two scenarios. The difference between them is only in reclassification of "difficult" or "ambiguous" patterns. So first version uses naive Bayes (NB) classifier to reclassify these objects, the second one uses logistic regression (LR) and the third one uses AdaBoost (AB) with decision stumps as week classifiers. All three versions use 90% of "ambiguous" objects for training and 10%–for testing. In some situations when the portion of difficult data is small in comparison with all available data the ratio between training and testing partitions should be reduced. Comparing naive Bayes and logistic regression using Tables 1 and 2 one may conclude that naive Bayes outperforms logistic regression on "difficult" data in most cases. This means that generative model might better fit "difficult" data than discriminative one which is oriented to training data and completely learns from the data. Nevertheless AdaBoost learns very well on entirely "difficult" patterns and actually does not overfit. This explains that reclassifying "ambiguous" data with boosting algorithms one might obtain very small error rates even on testing examples. Such a behaviour of AdaBoost can be explained by the fact that one reduced the hypothesis space by selecting the two most dissimilar classifiers from the pool of generated classifiers. These classifiers determine the set of "ambiguous" data. If one selects other two classifiers from the pool they can detect different subsets of such data. So appropriate selection of classifiers is a very important issue. Thus learning classifiers on a partition of "difficult" patterns which are grouped near the separating hyperplane and are characterised by lower variance in comparison to entire set allows us to reduce the set of all potential classification hypotheses. This leads to better learning, less error rate on the test data and hence less overfitting. In Table 3 the decomposition of data into functional groups is given. So any object can be misclassified, ambiguous or correctly classified by a consensus of algorithms. The difficulty of a data set can be estimated by a number of misclassified patterns. This is irreducible error. To have lots of "ambiguous" data might be not very bad since such kind of data can be learned effectively using for example AdaBoost. Table 4 shows how "good" are some classifiers in reclassifying "ambiguous" data. As seen the best results of such reclassification gives Adaboost. It outperforms naive Bayes and logistic regression quite a lot. On another hand naive Bayes outperforms logistic regression in most of cases. So based on two tables total error can be computed using (12) and is complement to accuracy in Tables 1 and 2. As seen from last two tables error of reclassification of "ambiguous data" using naive Bayes and logistic regression is high enough which means that selecting the most dissimilar classifiers is reasonable. However if we need less dissimilar classifiers we can also find them in the entire pool of generated classifiers. So selecting algorithms to build a consensus should be done in a way that allows us to optimise a total error which is trade-off between misclassification and error caused by "ambiguous" data.

**Table 3.** Distribution of different types of data (%)

| Method /task | 'pima' | 'bupa' | 'heart' | 'german' | 'sonar' |
|---|---|---|---|---|---|
| Misclassified | 14.0 | 12.7 | 10.2 | 15.0 | 7.6 |
| Ambiguous | 21.6 | 42.7 | 32.2 | 34.0 | 34.0 |
| Correctly classified | 64.4 | 44.6 | 57.6 | 51.0 | 58.4 |

**Table 4.** Reclassification accuracy of ambiguous data %

| Method /task | ‘pima’ | ‘bupa’ | ‘heart’ | ‘german’ | ‘sonar’ |
|---|---|---|---|---|---|
| naive Bayes | 65.8 | 58.5 | 80.8 | 59.5 | 45.0 |
| Logistic regression | 65.7 | 36.7 | 60.0 | 50.0 | 70.0 |
| AdaBoost | **100** | **93.0** | **95.0** | **89.0** | **100** |

## 5 Conclusions

The proposed scheme of combining pattern classifiers allows to obtain generally much better classifications in terms of performance and overfitting. Correct consensus probability does not vary much for different consensuses built for different Hamming distances for classifiers that are in consensus. Among different subsets of strong classifiers it is better to use ones with the highest entropy. To add one more classifier to the set of classifiers one might use greedy search which maximizes entropy at every step. Number of classifiers in the final subset should be minimised.

## References

1. Basu, M., Ho, T.K.: Data Complexity in Pattern Recognition. Springer, London (2006)
2. Biau, G., Devroye, L.: Lectures on the Nearest Neighbor Method. Springer, Cham (2015)
3. Frank, A., Asuncion, A.: Uci repository of machine learning databases. Technical report, School of Information and Computer Sciences, University of California, Irvine, CA (2010)
4. Freund, Y., Schapire, R.E.: A desicion-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995). doi:10.1007/3-540-59119-2_166
5. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning, 2nd edn. Springer, New York (2009)
6. Ho, T.K., Basu, M.: Complexity measures of supervised classification problems. IEEE Trans. Pattern Anal. Mach. Intell. **24**(3), 289–300 (2002)
7. Kapustii, B., Rusyn, B., Tayanov, V.: Classifier optimization in small sample size condition. Autom. Control Comput. Sci. **40**(5), 17–22 (2006)
8. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press, Cambridge (2009)
9. Kuncheva, L.: Combining Pattern Classifiers. Wiley, Hoboken (2014)
10. Kyrgyzov, I.O., Maitre, H., Campedel, M.: A method of clustering combination applied to satellite image analysis. In: 14th International IEEE Conference on Image Analysis and Processing, pp. 81–86 (2007)
11. Rokach, L.: Pattern Classification Using Ensemble Methods. World Scientific, Singapore (2009)
12. Rusyn, B., Tayanov, V., Lutsyka, O.: Upper-bound estimates for classifiers based on a dissimilarity function. Cybern. Syst. Anal. **48**(4), 592–600 (2012)

13. Vapnik, V., Izmailov, R.: Statistical inference problems and their rigorous solutions. In: Gammerman, A., Vovk, V., Papadopoulos, H. (eds.) SLDS 2015. LNCS, vol. 9047, pp. 33–71. Springer, Cham (2015). doi:10.1007/978-3-319-17091-6_2
14. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
15. Vorontsov, K.V.: Splitting and similarity phenomena in the sets of classifiers and their effect on the probability of overfitting. Pattern Recogn. Image Anal. **19**(3), 412–420 (2009)
16. Vorontsov, K.: Combinatorial probability and the tightness of generalization bounds. Pattern Recogn. Image Anal. **18**(2), 243–259 (2008)
17. Wozniak, M.: Hybrid Classifiers: Methods of Data, Knowledge, and Classifier Combination. Springer, Heidelberg (2013)
18. Xu, L., Krzyżak, A., Suen, C.Y.: Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. Syst. Man Cybern. **22**(3), 418–435 (1992)
19. Zhuravlev, Y.I.: On an algebraic approach to solving recognition or classification problems. Prob. Cybern. **33**, 5–68 (1978). (in Russian)

# Knowledge Based Active Partition Approach for Heart Ventricle Recognition

Arkadiusz Tomczyk(✉) and Piotr S. Szczepaniak

Institute of Information Technology, Lodz University of Technology,
Wolczanska 215, 90-924 Lodz, Poland
{arkadiusz.tomczyk,piotr.szczepaniak}@p.lodz.pl

**Abstract.** In the paper a method of automatic localization of heart ventricles in CT images is presented. Analysis of their shape can be an important element of pulmonary embolism diagnosis. For that purpose active partitions, a generalization of active contour approach, was used with superpixel representation of image content. Active partitions, similarly to active contours, possess a natural ability to incorporate external experience into object localization process. It means that not only information contained in the image itself but also experience of the radiologist and the medical knowledge can be used to improve segmentation results.

**Keywords:** Active contours · Active partitions · Superpixels · Medical imaging · Heart ventricles · Intelligent segmentation

## 1 Introduction

There are many different segmentation techniques that directly or indirectly can be applied for object localization in the images. However, classic methods like thresholding or region growing, usually have problems with utilization of external knowledge. Such knowledge is crucial in those tasks where the image itself contains insufficient information required for proper semantic interpretation of its content (medical images and experience of the radiologists are a typical example of such situation). Active contour techniques [2,3,5,7,10] constitute a group of methods that allow to overcome that problem. They assume that the space of contours, unambiguously identifying objects in the image, can be defined. Next the optimal contour is sought within this space by proper selection of objective function (energy) and optimization algorithm (evolution). The name of the method is a result of fact that optimization is usually an iterative procedure and consequently contour changes in every iteration. The external knowledge in active contours can be introduced in several ways:

- Proper selection of contour model can eliminate semantically incorrect solutions.
- Proper constraints in optimization process can make it impossible to obtain unacceptable solutions.

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_30

- Proper components of the energy function can penalize solutions that do not reflect user expectations.

Active partitions can be considered as a generalization of active contours. Contours, localizing object in the images, divide the whole set of image pixels into subsets representing objects and background. Such a partition can be defined, however, for any set of elements (e.g. superpixels, semantic patches, segments, ellipses, etc.) describing image content. The change of image description can significantly reduce the space of analysed primitives without losing the significant semantic information (it can be encoded in attributes assigned to those primitives or in relations between them). The paper is organized as follows: the second section describes a problem of heart ventricle segmentation, in the third section the active partition approach with superpixel representation is discussed whereas in the fourth section the obtained results are presented, finally the last section contains a short discussion and summary of the those results.

## 2 Data

Pulmonary embolism is a disease in which a blockage of an artery in the lungs occurs. To perform an indirect, non-invasive assessment of those changes analysis of *computed tomography* CT heart examinations can be used. To gather images used in this work ECG-gated 64-row CT scanner was used in a standard chest protocol. Next, the reconstruction with a dedicated workstation software was performed to obtain 4D sequences - 3D video sequences where the heart cycle is reconstructed in 10 phases with 8 slices containing two chamber short axis view (80 images for every patient). All the examinations were performed after intravenous injection of contrast media which results in a better (bright areas) visibility of blood in vessels and heart ventricles (Fig. 1). For the ventricle and septum assessment purposes endocardial and epicardial contours must be found. In this work all those contours were drawn by an expert (Fig. 1). After closer analysis of those contours it appeared that the interior of the ventricles contains not only the blood with injected contrast but also fragments of heart muscle that grows into the chambers. It causes that to correctly interpret image content the additional knowledge must be used since the image information in those areas is insufficient.

## 3 Method

The application of active partition approach requires several elements to be defined: image representation, partition model, energy function and evolution scheme. In this work the content of the images was described using superpixel representation which reduces (approximately 80 times) the space of classified elements. The external knowledge required for proper segmentation is encoded both in energy function (border smoothness) and evolution procedure (solution generator). All the elements are described below.

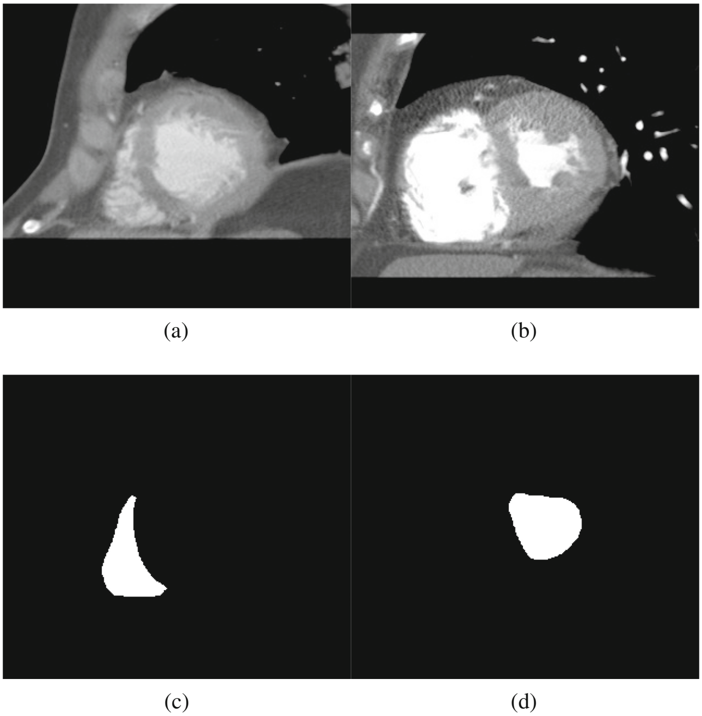
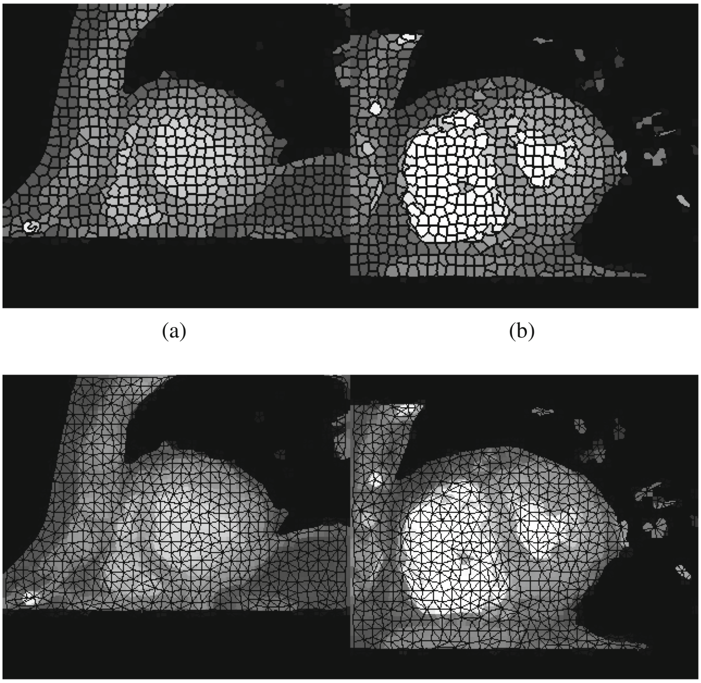


**Fig. 1.** Sample images and masks indicating localization of heart ventricles: (a), (b) - images, (c), (d) - masks generated basing on contours manually drawn by radiologist.

### 3.1 Representation

To represent image content with superpixels a *simple linear iterative clustering* SLIC technique was used [1]. This method adapts classic *k-means* clustering to find group of pixels that lie close to each other and have similar colour. This procedure is followed by additional postprocessing forcing all the superpixels to represent connected regions. The user can control a trade-off between the influence of pixel position and color and in consequence can obtain either more compact or more homogeneous superpixels. In this work the number of pixels (approximately 1000) as well as the characteristic (homogeneous but still compact) of the superpixels were chosen arbitrarily after visual inspection of sample results (Fig. 2). Superpixels describe only how the image area is divided into smaller regions. In order to use them in active partitions the actual knowledge about image content must be tied with this representation. By analogy to pixel representation that knowledge can be encoded either as the attributes of the individual elements (colour components in the case of pixels) or in the attributed relations between those elements (spatial relations in the case of pixels). In this work every superpixel is described by its: area, average image intensity, bounding box and the number of pixels representing blood within that superpixel. The last property was possible to calculate thanks to the earlier detection of pixels



(a) (b)

(c) (d)

**Fig. 2.** Representation of image content: (a), (b) - superpixels with average intensity assigned to them, (c), (d) - neighbourhood relation of the above superpixels.

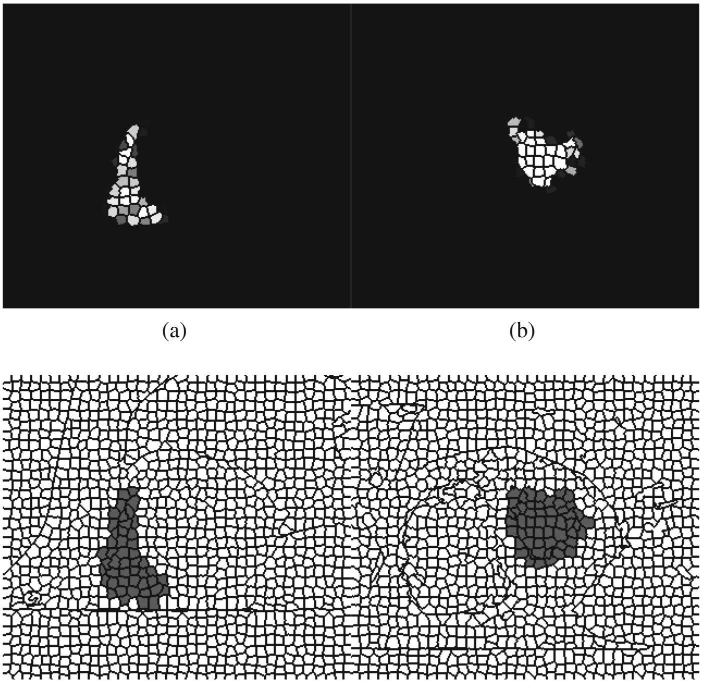
representing blood within ventricles [11]. Some of those attributes are later used for heart ventricle localization while the other enable acceleration of computations. As a relation between superpixels the neighbourhood relation was defined. Superpixels are considered to be neighbours if they are adjacent.

### 3.2 Model

In active contour techniques contour model determines what kind of partitions of pixel set are acceptable. Taking into account the number of pixels in the images considering all the possible partitions would lead to the enormous search space. That is why contour models usually reduce it allowing only partitions that describe connected regions or regions of reasonable shape. In the presented work the number of the partitioned elements is significantly smaller and consequently any partition is allowed (all the subsets are allowed). Additional constraints (external knowledge) will be imposed by optimization procedure (hard) and energy function (soft). Of course this is not the only possible partition model in the case of considered image representation. For example the approach in *Brownian strings* [5] could be adapted also for superpixels with spatial neighbourhood defined.

### 3.3 Energy

Energy function evaluates partitions and is an objective function in the evolution process. It should be defined in a way reflecting knowledge about object that is sought. For the perfect solution it should have a minimum value. As it was mentioned above with every superpixel the information about number of pixels representing blood is stored (Fig. 3). Of course all those pixels should lie within a the sought ventricle. This expectation can be expressed by the $E^o$ energy component which measures the number of superpixels with blood that are outside of the current solution (subset of superpixels). Finding of all such superpixels, however, is not sufficient since heart muscle growing into heart ventricles should also be identified as its interior. This is an additional, medical knowledge without which the proper interpretation of image content is not possible. The simplest solution of that problem can be adapted from classic active contour techniques where in the case of insufficient image information contour smoothness is forced. One of the approaches used in this case assumes minimization of contour length [9].


(a) (b)

(c) (d)

**Fig. 3.** Available knowledge: (a), (b) - the number of pixels representing blood inside a ventricle within a given superpixel (the brighter colour the more such pixels were found), (c), (d) - superpixels that according to expert should be identified as a ventricle (the superpixel was selected if there was at least one common pixel with the mask prepared by radiologist).

It is a reasonable approach since contour of minimal length will not penetrate cavities of the sought object. The same procedure can be used in this research if contour length will be replaced by the number of superpixels lying on the border of current solution $E^b$. Consequently, the final energy function considered further have the following form:

$$E = E^b + wE^o$$

A border is a set of superpixels that has neighbours which do not belong to the solution (Fig. 4). It can be defined since the neighbourhood relation is defined. Parameter $w$ controls a trade-off between those two components. It is required since partition with smallest border (one superpixel) will not contain all superpixels with blood and similarly partition containing all such superpixels will not have the smallest border.

**Table 1.** Average $F1$ scores (training set) for different values of parameter $w$ and different solution generators $G$: (a) - right ventricle, (b) - left ventricle.

(a)

| $w$ | 0.5 | 1 | 2 | 3 |
|---|---|---|---|---|
| $G^s$ | 0.0145 | 0.9102 | **0.9155** | 0.9122 |
| $G^c$ | 0.0131 | **0.9142** | 0.9038 | 0.8994 |

(b)

| $w$ | 0.5 | 1 | 2 | 3 |
|---|---|---|---|---|
| $G^s$ | 0.1031 | **0.8967** | 0.8924 | 0.8865 |
| $G^c$ | 0.1033 | **0.8989** | 0.8777 | 0.8824 |

## 3.4 Evolution

The considered problem of finding the optimal partition is a combinatorial optimization task (the subsets of superpixels representing heart ventricles are sought). Many techniques can be applied for that kind of problem. In this work a *simulated annealing* is used because of its simplicity and theoretical convergence to global solution [8]. In that approach in every iteration a new solution is

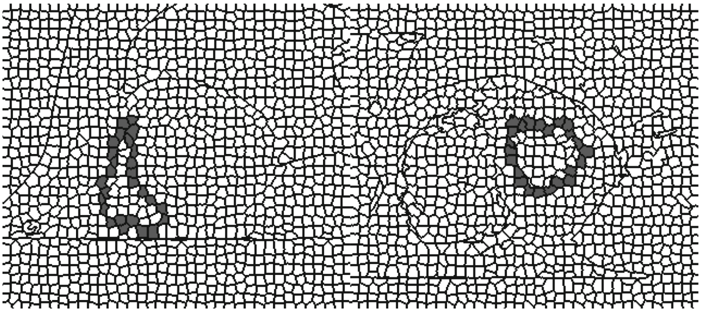
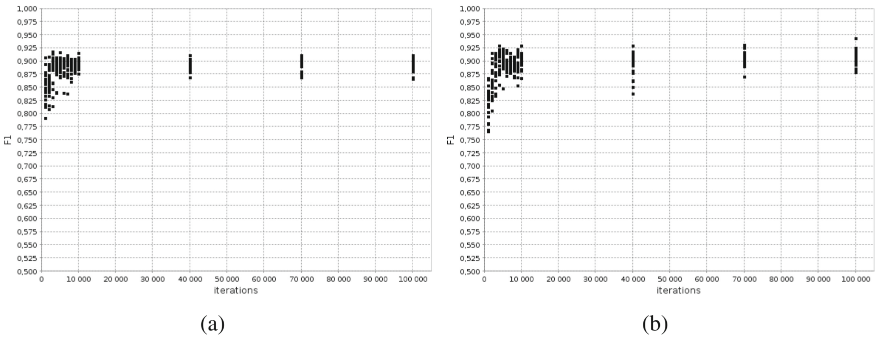


**Fig. 4.** Border of the regions identified by an expert as a ventricle.



(a)

(b)

**Fig. 5.** Sample $F1$ scores for different number of iterations $w = 2$ and generator $G^s$ (experiments were repeated 20 times): (a) - right ventricle, (b) - left ventricle.

proposed using a solution generator $G$. Solution generator should generate a random solution which is close to the current one and it should allow to explore the whole search space during optimization process (there should be always a possibility to generate a solution sequence transforming one solution to the other).

**Table 2.** Average $F1$ score for 400 images in a test set: (a) - right ventricle, (b) - left ventricle.

(a)

| | $F1$ | |
|---|---|---|
| $T$ | **0.9108** | |
| | $w = 1$ | $w = 2$ |
| $G^s$ | 0.8799 | 0.9075 |
| $G^c$ | 0.8976 | 0.9047 |

(b)

| | $F1$ | |
|---|---|---|
| $T$ | 0.8959 | |
| | $w = 1$ | $w = 2$ |
| $G^s$ | 0.9021 | **0.9124** |
| $G^c$ | 0.9059 | 0.9108 |

## 4 Results

In this section the conducted experiments are described together with discussion of obtained results. All the available data were split into training (10 patients) and test (5 patients) set. Training set (or its representative subset) was used to select parameters. Testing set was used to evaluate the quality of the methods. In all the experiments, to generate an initial solution, the superpixel in the centre of the image was sought. This superpixel and all the its neighbours in the range of 3 were considered as the partition in the first iteration of evolution process. To compare the results with some classic approach, which does not take into account additional knowledge, a simple thresholding procedure $T$ was used where only the superpixels containing blood inside ventricles were considered as superixels representing this ventricle. Within the *simulated annealing* two solution generators were considered:

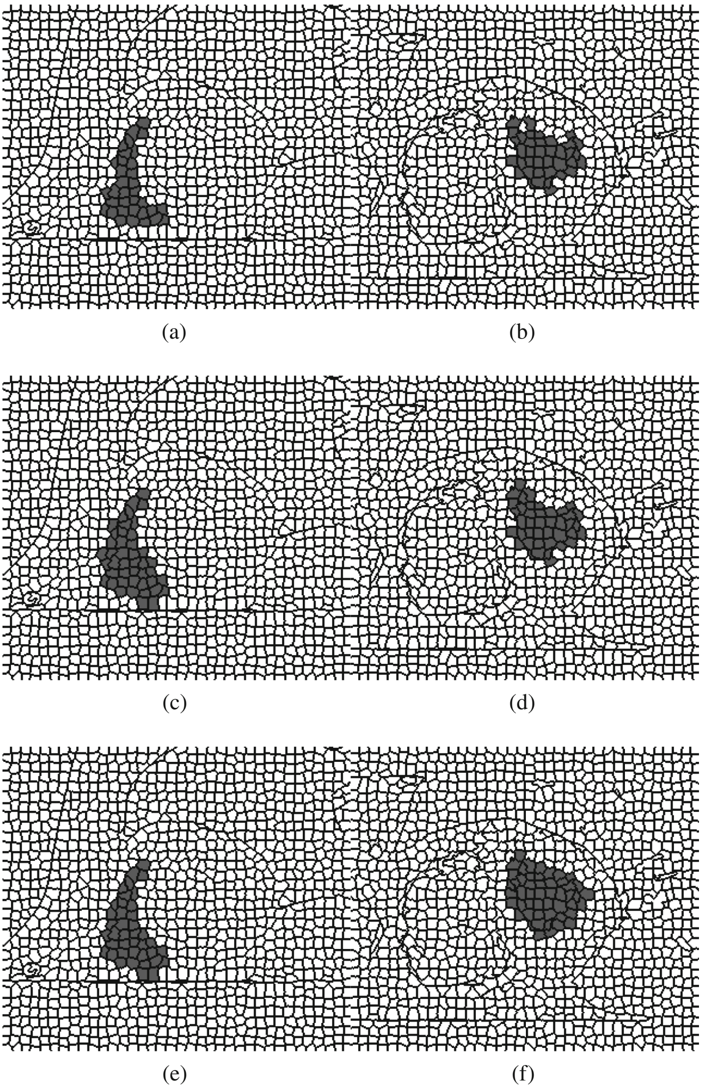


**Fig. 6.** Sample results: (a), (b) - thresholding, (c), (d) - active partitions with simple solution generator $G^s$, (e), (f) - active partitions with connected solution generator $G^c$.

- $G^s$ - simple generator which either removes one element on the border of the current set or adds one element which is adjacent to the border of the current set
- $G^c$ - connected generator which behaves in the same way as the simple generator but prevents situations where a new solution has either holes or is composed of two disconnected parts

**Table 3.** Distribution of $F1$ score differences ($G^s$ and $G^c$ was compared with $T$) for 400 images in a test set: (a) - right ventricle, (b) - left ventricle.

(a)

| | $< 0$ | $= 0$ | $> 0$ |
|---|---|---|---|
| | $w = 1$ | | |
| $G^s$ - $T$ | 229 | 3 | 168 |
| $G^c$ - $T$ | 237 | 4 | 159 |
| | $w = 2$ | | |
| $G^s$ - $T$ | 208 | 2 | 190 |
| $G^c$ - $T$ | 227 | 1 | 172 |

(b)

| | $< 0$ | $= 0$ | $> 0$ |
|---|---|---|---|
| | $w = 1$ | | |
| $G^s$ - $T$ | 125 | 16 | 259 |
| $G^c$ - $T$ | 132 | 9 | 259 |
| | $w = 2$ | | |
| $G^s$ - $T$ | **88** | **10** | **302** |
| $G^c$ - $T$ | 116 | 7 | 277 |

Here again border and topological properties of solutions can be considered because the neighbourhood relation is defined in the set of superpixels.

### 4.1 Evaluation

Partition can be treated as a binary classifier (object and background labels) of primitives representing image content. Consequently, to assess segmentation results, methods used for classifier evaluation can be applied. That is why further $F1$ score, which is a combination of precision and recall, will be employed. Precision measures how many primitives that do not represent ventricle were recognized as its part. Recall measures how many primitives within ventricle were not classified correctly. The optimal solution should give a $F1$ score close to 1. All those scores can be calculated for:

- pixels - label of superpixel is assigned to all the pixels composing it and masks given by expert are available (Fig. 1)
- superpixels - labels of superpixels can be determined using the pixel masks prepared by expert (Fig. 3)

Further only the values for superpixels are considered.

### 4.2 Repeatability

Because of random character of *simulated annealing* algorithm, before the actual experiments can be conducted, the proper number of iterations must be selected to ensure that the obtained results are repeatable. In theory it is possible if the optimization process is sufficiently long. For that purpose for selected images the evolution process was repeated 20 times with different number of iterations. Sample $F1$ values obtained for two different images are presented in Fig. 5. As it can observed in fact the deviation of the results, for a given number of iterations, decreases when the number of iterations increases. The stability of the deviation can be observed when the number of iterations is grater than 70000 and this value was used in further experiments. Closer analysis of presented charts leads to another interesting observation. The $F1$ values are not the same even if many iterations are considered. The reason of that situation is the specificity of the energy function and image representation. Changing the label of one superpixel may not change the value of the energy function but of course will influence the $F1$ score. Moreover, taking into account the number of superpixels representing ventricles such a small modification can result in noticeable change of this score. This observation should also be taken into account while interpreting further results.

### 4.3 Parameters

The only parameter that needs to be selected to conduct experiments is a weight $w$ controlling a trade-off between the energy components. An intuitive analysis of energy function suggests that optimal value should be greater or equal to 1 since otherwise almost all modifications of the partition with proposed solution generators $G$ will produce better solutions and consequently will be accepted. However, since the superpixel neighbourhood is not so typical as in the case of pixels, the intuition may fail. That is why to prove the above suppositions a supplementary experiment was performed where to find an optimal value a training set was used. Using different values of parameter $w$ the segmentation was performed for the images from this set and average $F1$ score was calculated. The results are presented in Table 1. They show that in fact values of 1 or 2 should be considered.

### 4.4 Discussion

Sample results of all considered approaches (thresholding and active partition with two different generators) for left and right ventricle are presented in Fig. 6. For those two cases the superiority of active partition approach with generator $G^c$ is especially evident for left ventricle where heart muscle grows into chamber interior. This observation was confirmed by statistical analysis of results for 400 images in a test set. In Table 2 average $F1$ score is shown. For right ventricle segmentation the thresholding was sufficient since blood was apparently filling almost the whole its interior and the size of superpixels allowed to notice that presence. In case of left ventricle, however, the image information was not always sufficient and the external knowledge, encoded in border size energy component, allowed to improve the results. Table 3 shows that for left ventricle, simple solution generator and $w = 2$ active partition was better then thresholding in 75% of cases. Authors are aware of the fact that the segmentation could be improved either by taking into account more complex energy function or by better granularity of image representation (smaller superpixels). A good idea could be considering the achieved coarse segmentation results as input to other, more precise, technique e.g. traditional active contour like *snake* or *potential active contour*. Nevertheless, the main goal of this work was to show, illustrating it with a simple example, that it possible to successfully transfer active contour approach to other than pixel based representations of image content. In opinion of the authors this goal was fully accomplished.

## 5 Conclusions

In the paper a method of applying concepts, well known from active contour techniques, to other than pixel based image representations was proposed. Here the image was represented as a set of superpixels with neighbourhood relation defined among them. This representation reduces significantly the search space in segmentation task without losing crucial, semantic image information (it can be encoded in superixel attributes). The proposed approach was called active partition. The crucial advantage of active contours, which distinguishes them among other segmentation techniques, is their ability to use external knowledge there were there is insufficient information in the image. In this work a transfer of this ability to active partitions was also proposed (contour length minimization and expected region characteristic). The approach was illustrated in a problem of heart ventricles localization in CT images. The experiments revealed that the external requirement forcing the minimal length of the partition border improves segmentation results if heart muscle grows into chamber interior (the improvement was observed mainly for left ventricle where such problems occured). Since the considered energy function was very simple, further work should focus on looking for more complex energy functions (in particular those with parameters that can be automatically selected using an available training set). The other work direction is a search for better image representations (in particular 3D representations). All those aspects are under further investigation.

**Acknowledgements.** This project has been partly funded with support from National Science Centre, Republic of Poland, decision number DEC-2012/05/D/ST6/03091. Authors would like to also express their gratitude to Mr Cyprian Wolski, MD, from the Department of Radiology of Barlicki University Hospital in Lodz for making heart images available and sharing his medical knowledge.

## References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2281 (2012)
2. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Comput. Vision **22**(1), 61–79 (2000)
3. Cootes, T., Taylor, C., Cooper, D., Graham, J.: Active shape models - their training and application. CVGIP Image Underst. **61**(1), 8–59 (1994)
4. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 484–498. Springer, Heidelberg (1998). doi:10.1007/BFb0054760
5. Grzeszczuk, R., Levin, D.: Brownian strings: segmenting images with stochastically deformable models. IEEE Trans. Pattern Anal. Mach. Intell. **19**(10), 1100–1113 (1997)
6. Ivins, J., Porrill, J.: Active region models for segmenting medical images. In: IEEE Transactions on Image Processing, pp. 227–231 (1994)

7. Kass, M., Witkin, W., Terzopoulos, S.: Snakes: active contour models. Int. J. Comput. Vision **1**(4), 321–333 (1988)
8. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
9. Siddiqi, K., Lauziere, Y., Tannenbaum, A., Zucker, S.: Area and length-minimizing flows for shape segmentation. IEEE Trans. Image Process. **7**(3), 433–443 (1997)
10. Tomczyk, A., Szczepaniak, P.S.: Adaptive potential active contours. Pattern Anal. Appl. **14**, 425–440 (2011a)
11. Tomczyk, A., Szczepaniak, P.S.: Knowledge extraction for heart image segmentation. In: Burduk, R., Kurzynski, M., Wozniak, M., Zolnierek, A. (eds.) Computer Recognition Systems 4. Advances in Intelligent and Soft Computing, vol. 95, pp. 579–586. Springer, Heidelberg (2011)

# The Method of Teeth Region Detection in Panoramic Dental Radiographs

Jakub Zak[1](✉), Anna Korzynska[1], Lukasz Roszkowiak[1], Krzysztof Siemion[1,2], Sebastian Walerzak[3], Monika Walerzak[3], and Konrad Walerzak[3]

[1] Laboratory of Processing Systems of Microscopic Image Information, Nalecz Institute of Biocybernetics and Biomedical Engineering Polish Academy of Sciences, Ks. Trojdena 4 Street, 02-109 Warsaw, Poland
jzak@ibib.waw.pl

[2] Department of Pathomorphology, Central Clinical Hospital of the Ministry of the Interior and Administration, Woloska 137 Street, Warsaw, Poland

[3] NZOZ Centrum Leczenia Wad Zgryzu, Luczek 4 Street, 02-434 Warsaw, Poland

**Abstract.** Radiographs, which visualise human dentition as a single panorama photograph, are commonly used in prognosis and diagnosis in dental care and biometric individual identification. The automatics analysis of these images can provide useful information about patients. So, this paper proposes a method of an introductory phase of teeth segmentation: selection of a region of interest which contains every tooth in the pantomograph. The proposed method finds an area containing all the teeth and the line dividing the upper and lower jaws, called teeth line, using adaptive power-law transformation, Hölder exponent, energy and entropy of the pantomograph. The teeth area is found correctly in 81.9%, while teeth line in 34.0%. The method is the introductory step in a single teeth segmentation, dentition evaluation according to presence of strong dental interventions and automatics generation of dental score for a patient.

## 1 Introduction

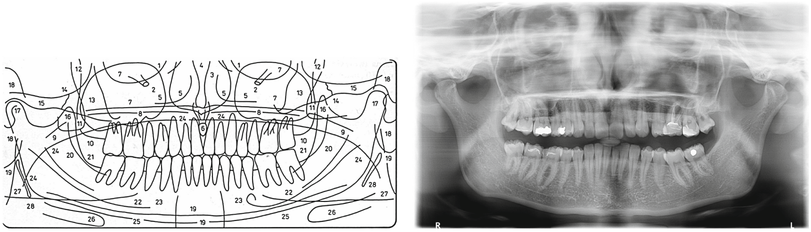
Dental X-rays, also known as radiographs, are commonly used in prognosis and diagnosis in dental care [4,6,16] and also biometric individual identification [2,7,9,11,12,15,18]. There are several types of radiographs: bite-wing (intraoral), periapical (intraoral), occlusal (intraoral), cephalometric (extraoral) and panoramic (extraoral) [8]. Panoramic radiographs (pantomographs) are one of the most complex, since they show the whole dentition of an individual, as well as parts of the skull and the cervical part of the spine [8]. Pantomographs are taken by the rotation of a film and an X-ray lamp around the head of a patient. The film and lamp are located opposite each other. This way, a single image as a panorama photograph is created. This gives dentists an opportunity to examine both the maxillary and mandibular teeth. The continuous digitalization process applied to medicine during last decades and benefits of automatic analysis of biomedical

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_31

data have also reached dental radiology [13,14,17]. Radiological images are used mainly in caries diagnosis, determining bone loss in periodontal diseases, diagnosis of periapical cysts, planning of operations in the maxillofacial region, planning of orthodontic treatment and also in individual identification, comparing post-mortem and archived ante-mortem radiographs. In both areas, a crucial step is the segmentation of teeth and each tooth separately. Several authors have already described numerous methods for teeth segmentation in various radiograms: bite–wing [5,11,18], periapical [4–6], occlusal [16], cephalometric [16] and panoramic [5,10,15]. Our study concerns pantomograph analysis, an issue, for which the literature is severely limited. However, the potential for the aid of diagnosis is large. Unfortunately, due to the complexity of pantomographs, it is very hard to extract single teeth from the whole image. What is more, caries diagnosis is also a complex issue: it would involve finding subtle differences in the shading of the teeth. Another problem to overcome is the existence of artefacts within the pantomograph. Patients with extensive history in dental procedures can possess multiple objects, usually very bright, which could cover the whole tooth, therefore increasing the difficulty of segmentation. In [15], the first steps of teeth segmentation (selecting the area with teeth) are omitted or prepared using statistic results from their own database [10]. Although this may work well for many cases, every variation in the data (due to the machine's, dentist's or patient's individual characteristics) may disturb the analysis. The aim of this study is the introductory phase of teeth segmentation: selection of a region of interest (ROI), which includes every tooth in the pantomographs (select an area of the teeth and reject an area of other hard tissue). This step is not easy because of inclusion of additional bone structures in pantomographs. The next step of the investigation will be separate teeth selection and analysis to fill a report about the condition of dentition of a patient, for the dentist to examine.

## 2 Materials

The algorithm is only expected to work on panoramic dental radiographs. This introduces several limitations and increases the complexity of the problem. In Fig. 1, a schematic (a) and exemplary (b) pantomograph are presented. As seen in Fig. 1, pantomographs include much more than teeth structures. All of those, marked (1–28) in Fig. 1a, are additions, which are superfluous to the diagnosis of dental caries. Not only do the pantomographs suffer the inclusion of unnecessary information, they are also subject of typical errors in images: mainly noise, reflection and incorrect position of patient's head during acquisition. Teeth are light gray objects, with inhomogeneous brightness. Sometimes they have a white rim or fillings, located partly on textured background in the region of the bone and partly on dark grey background in the mouth. The most prominent object in a pantomograph is palatal bone visible as white, almost horizontal line in the $\frac{1}{3}$ to $\frac{1}{2}$ of pantomograph height. The structures over this line are redundant and should be reduced as they are similar in texture and brightness to the teeth. All this makes any segmentation a nontrivial problem [16]. A database of 100 pantomographs has been created, courtesy of NZOZ Centrum Leczenia Wad Zgryzu



(a) (b)

**Fig. 1.** (a) Scheme of anatomical structures [8]: 1. orbital margin; 2. suborbital foramen; 3. nasal cavity; 4. nasal septum; 5. nasal concha; 6. incisive foramen, nasopalatine canal, nasal spine; 7. maxillary sinus; 8. nasal base; 9. soft palate; 10. maxillary tuberosity; 11 pterygoid processes; 12. pterygopalatine fossa; 13. zygomatic bone; 14. zygomaticofrontal suture; 15. zygomatic arch; 16. coronoid process; 17. condyloid process; 18. outer ear; 19. cervical part of the vertebral column; 20. mandibular crest; 21. oblique line of the mandible; 22. mandibular canal; 23. mental foramen; 24. dorsum of the tongue; 25. compact bone of the mandibular body; 26. hyoid bone; 27. contralateral structures of the viscerocranium; 28. styloid process. (b) Exemplary pantomograph.

in Warsaw. They were taken using Kodak 9000 pantomograph. Out of 100, 6 were rejected because of their quality (presence of the reflections (3), movement (2) or errors in pantomograph merging (1)). The rest are:

- 9 adults with noticeable gaps in teeth (mainly molar teeth)
- 33 adults with noticeable artefacts caused by the process of pantomograph acquisition or big changes in dentition (filled cavities and tooth canal treatment, presence of braces and the artificial bone inclusion of the jaw like implants and bone brackets)
- 48 adults (with all teeth and minor artefacts)
- 4 children (typical with deciduous teeth and permanent teeth buds)

One pantomograph of each type is presented in figures in this paper (Fig. 4). All the procedures are written in MATLAB code.

## 3 Methods

To perform single tooth segmentation in pantomographs, the region of interest should be defined as an area containing all of the teeth. Selection of the ROI and determination of the position of the gap between the upper and lower jaw is the goal of the study presented in this paper. This requires finding distinguishing features, which clearly divide the image into several segments, including the one containing only dentition and the inside of this region to determine the gap. All of the steps of the image processing are presented in following figures in this paper (Figs. 2 and 3). The final result is designation of the selected area containing all the teeth and the line dividing the upper and lower jaw, so called teeth line. The proposed algorithm is the first part of a method, called Individual Arch

Teeth Segmentation (IATS). The first phase of the proposed method is finding the line of the palatal bone, usually one of the brightest horizontal lines in the image, if the image acquisition is done correctly. To do that, the original image is processed to increase the contrast between bones and empty space in radiograph. It is done in 8 following steps (steps 5–8 were introduced by the authors):

1. At the beginning, top hat transform of the original image is added to the original, while the bottom hat of the original is subtracted. This step is required for contrast enhancement in the border between objects [18]. Images before and after this operation are not presented in this paper, as the differences between them are so delicate, that they will not be visible in the printed version of the paper.
2. Next, following the scheme presented in [6], adaptive power-law transformation (APLT) is performed on the image to further enhance it. APLT is a transformation, based on a local intensity of the pixels around the chosen pixel. It strongly enhances the contrast in the image, making the borders between the teeth more distinct and smoothing the texture within the teeth [6]. In this paper, the local window used in APLT is $3 \times 3$ pixels. To decrease the computation time, this and all other local operations are computed in parallel. For clarity, every operation will be explicitly marked as 'parallel' during its introduction, like this: adaptive power-law transformation (parallel).
3. Further step is also based on a procedure proposed in [6]. It involves texture analysis using Hölder exponent as the singularity measure. Singularity is a concept for representing complex signals [1]. Signals or structures can be presented as a superposition of singularities. Point-wise singularity may be observed by calculating Hölder exponent for every pixel, characterized as:

$$\alpha_i(x,y) = \frac{\ln \mu(x,y)}{\ln i} \tag{1}$$

   where $\mu$(x, y) is the amount of measure within the observed box with size $\epsilon =$ i centred at the pixel (x,y). Following [6], a mean (avg) measure is used, with local window: $3 \times 3$ pixels. Since the use of natural logarithm and division may result in imaginary, infinite or NaN (Not-a-Number) values, a cleanup procedure is required. All of those values are switched to 0, then the image is normalized to [0–1] range.
4. Bilateral filtering [3] is then used on the matrix of measures to preserve the edges and remove fine textures [6]. The resulting image is very bright, so is not presented in the paper. But the image being the result of subtraction (difference with absolute value) of the filtered image from the original is presented in Fig. 2a. The result of these 4 steps is an inverted brightness image, so the bones are darker than empty space, teeth fillings and palatal bone are almost black. The process selectively increases the contrast between strongly visible bones and background and decreases it between less visible bones and background.
5. An introductory ellipse mask is created to remove some parts of the image from consideration, which allows focusing on the palatal bone. An ellipse is

a proper approximation of the dental shape, which could include all of the teeth in pantomograph, while its location and size should be adjusted to the morphology of patient's dentition. The introductory ellipse is positioned in the centre of the image, its minor axis has the length of 550 pixels, major – 825 pixels. Minor axis is vertical, major one is horizontal. Its size and position have been chosen according to the knowledge that the teeth and the palatal bone should be located more or less in the middle of the image. The image after employing introductory mask is presented in Fig. 2b.

6. The next step is an adaptive thresholding (parallel) algorithm. In comparison to simple thresholding, adaptive thresholding is a local operation, which marks the parts of the image as either object (1/white) or background (0/black), depending on the chosen parameter. In this algorithm, the mean of the brightness in the window of 75 × 75 is used. Then, to clean up single white pixels and focus only on the largest objects within the ellipse, close (window 3 × 3) and erode (window 5 × 5) operations are used. The result of this step is presented in Fig. 2c.
7. Since the peripheral parts of the image are deemed unnecessary, the resulting border around the edges of the image is easy reduced. The remaining objects are sorted by their area (in pixels) and their orientation, in regards to the horizontal line (0°). In a typical pantomograph, the palatal bone is a long, very bright and approximately horizontal line, so the criteria for choosing an object are: size larger than 10000 pixels (to remove small groups) and orientation within [–8 ; 8] degrees. The result is a single object within the introductory ellipse: the palatal bone, presented in Fig. 2d.
8. The most important for next steps is finding a single horizontal coordinate of palatal bone, which is done using a median of all horizontal coordinates of the found object. This removes the errors introduced by other objects, like single teeth, merging with the palatal bone.

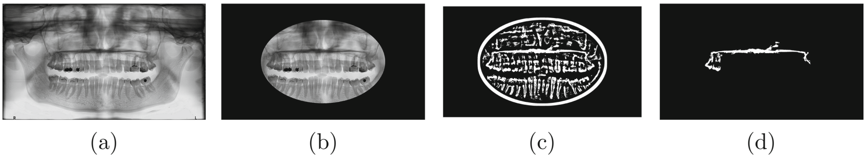

(a) (b) (c) (d)

**Fig. 2.** (a) (Original - bfilter); (b) $1^{st}$ ellipse masking; (c) Adaptive thresholding of (b); (d) Palatal bone object selection

Using the knowledge about palatal bone location and knowledge that teeth should be located in most cases directly beneath it, the ROI is constructed as final mask in the shape of a trimmed ellipse, adjusted to the morphology of a patient's dentition. For the final masking, a rectangular mask is considered. Its width would be limited by following borders: 200 pixels into the image from both sides, 200 up from the bottom. The top limit is the median of the Y coordinate of palatal bone. Using this rectangle's centroid, a final ellipse is located at the centre of the rectangle, its minor axis is 2.4 ($2 \times 1.2$) of the distance between the centre and the position of palatal bone and its major axis is equal to the rectangle horizontal size. The value of 1.2 ensures that all the teeth are left uncut (the curves would have otherwise removed parts of tooth's roots). The ellipse is then cut from the top, using the found palatal position. During the next steps, only those masked parts of the original image are processed.

The next goal is to find the line of the teeth. A novel method of finding this curve based on information about entropy and energy of the image is proposed. The following steps enhance the area chosen by the ROI towards increasing darkness of the empty space between the upper and lower jaws, which allows finding of the line of the teeth. It is done using mathematical operations of entropy and energy estimation in the image. Both energy and entropy may be used to increase the differences in the image, with different impact. The energy is defined as:

$$E(x, y) = I(x, y)^2 \tag{2}$$

While entropy:

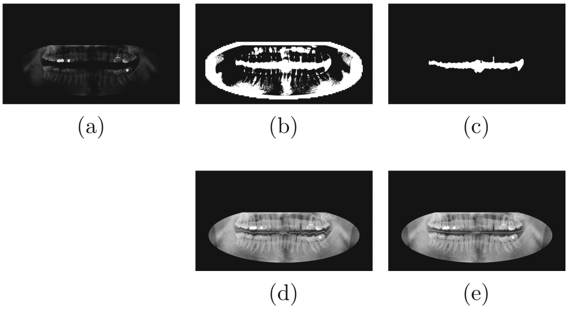
$$Ent = -\sum I(x, y) * \ln I(x, y) \tag{3}$$

where x, y – row and column index of the image, I – greyscale value of the image at point (x, y). From this local entropy value is derived:

$$Ent_{loc} = I(x, y) * \ln I(x, y) \tag{4}$$

The algorithm takes advantage of the fact that entropy changes between the image and its resized version. The resize method used in the proposed algorithm is bicubic. It produces result in 8 following steps:

1. First, an image with size reduced twice in every dimension is created and its local entropy is calculated pixel by pixel.
2. Then, energy of the original image is calculated.
3. Energy is resized (reduced) 16 times in every dimension and its local entropy is calculated pixel by pixel.
4. Energies of images created in steps 1 and 3 are added to the energy of the original image. In the resulting image, there bone structure and texture are clearly distinguished from empty space (very dark) between upper and lower teeth, presented in Fig. 3a.
5. The image is thresholded, using regular threshold operation with threshold level equal to 5 (Fig. 3b).
6. For cleanup, the binary image is eroded (window $7 \times 7$) and the holes within are filled. Then, all of the remaining objects are checked on their area and position of their centre. If their area in pixels in larger than 10000, they are considered as a candidate for the space between the teeth. The decisive factor is the distance between the centres of the second ellipse and the object. The one with the smallest distance is considered the space between the teeth (Fig. 3c).



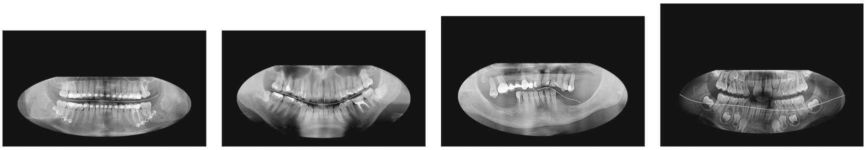
(a) (b) (c) (d) (e)

**Fig. 3.** (a) Mathematical contrast increase; (b) Thresholded entropy; (c) Gap between teeth object; (d) Complex teeth line; (e) Smooth teeth line

7. Using median filter (window $15 \times 15$) and erosion (window $3 \times 3$) the shape of the space between upper and lower teeth is smoothed and additional parts of the object, like the spaces between the teeth are removed.
8. The line between the teeth is found, as X position of mean value of calculated for each column in found space (local centres). This creates a very complex line, which sometimes diverges onto the teeth in the middle of the image, which is presented in Fig. 3d. This is caused by the middle teeth being relatively dark.
9. Therefore, a polynomial fitting is used to smooth the line. In the method, a polynomial of 5th degree is used. As a result, a line traversing the whole space between the teeth is created, which is presented in Fig. 3e.

## 4 Results and Discussion

The algorithm was tested on all 94 images. Since it is only an introductory step, the results were assessed visually by the authors. Expert assessment will be required after accomplishing teeth segmentation. In the analysis, the results are divided into 4 categories: Palatal bone found (P), Teeth line found with incorrect palatal bone finding (T), Both (palatal bone and teeth line) found ($B = P \cap T$), Incorrect results (I). Due to the fact that if palatal bone is not found, it is still possible to find the line of the teeth, the T group was introduced. Another note is the fact, that sometimes the teeth line can correctly traverse the actual teeth line, but is too short to go through the whole dentition. Figure 4c presents such an example. This kind of result is also marked as correct, if the position of the short line is correct. The extension of the line will require additional steps and fine tuning. Table 1. presents general statistics of the results. Clearly, the most effective step is the finding of the palatal bone. From among 12 images with incorrect results, 9 included very blurred the palatal bone which darkened it so significantly, that it hardly stood out from the background. If the pantomograph is acquired with wrong patient's head position, palatal bone



(a) (b) (c) (d)

**Fig. 4.** (a) Image with artefacts; (b) Typical image; (c) Image lacking teeth; (d) Image of paediatric patient

**Table 1.** Algorithm effectiveness.

| | Palatal bone found | Teeth line found | Both found | Incorrect |
|---|---|---|---|---|
| General (94) | 81.9% (77) | 34.0% (32) | 28.7% (27) | 12.8% (12) |

line is blurred or doubled with lower intensity. Another reason is image blurring, caused by patient's involuntary movement during acquisition. So, the first step for the improvement of the algorithm is finding a way to selectively enhance palatal bone region. Secondly, the teeth finding algorithm has not yet achieved satisfactory effectiveness. There are crucial areas for improvement, which will be implemented in the near future. The first one is making sure the line does not traverse the teeth, which disqualified 23 images form P category from putting them into B. Comparing to actual number of results in the B group, it could easily double the effectiveness. The second challenge is adapting the algorithm to varying brightness within the space between upper and lower teeth. The varying brightness (mostly because of presence of a mouthpiece between the teeth) is usually the reason for the shortening of a properly positioned line. In the paediatric group, the second part of the algorithm was 100% unsuccessful.

Another important factor is the computation time. Usually, end users are impatient to receive the results, therefore algorithms that perform quickly are in demand. Recording computational time of every run lead to statistical analysis of the performance of the algorithm. Out of 94 images, only 83 were processed fully. Sometimes, the analysis stopped after the palatal bone finding, therefore those times were omitted - they lowered the average. The mean of the accepted times was 152.38 ± 15.84 s. Rounded to minutes, it is 2:32:38 ± 0:15:84.

In comparison to other studies, the algorithm fares well. Using the published statistics, additional comparison tests were omitted. In [15], there were 218 images used. However, there were no statistical measures for comparison. The authors claim that the whole process took 213 s (1 s + 212 s) and that there were some problems with finding the teeth line and separate teeth, but overall their algorithm was said to be very effective. In [10], the statistical analysis is available. Out of 700 images, 95.7% had correct ROI and 92.6% ended with correct jaw partition (teeth line finding). However, the ROI definition step is based on statistical approach and actual distance measurements. That kind of approach can be subject to batch effect and not work, when presented with different set or an image from outside the statistical analysis. Although the data used in this study also comes from a single machine, the approach should be more flexible and uses the distances calculated for a particular image, rather than a global constant. No data was found about the speed of computations of the first two steps in this algorithm, but the process does not seem to be time-consuming.

## 5 Conclusions

In this study a novel approach to panoramic dental radiographs analysis was presented. It is the first step to completely automatic teeth segmentation from pantomographs, which can be applied to dental diagnosis and individual identification. The method results in a trimmed ellipse-shaped ROI, which encompasses all of the teeth in the pantomograph, and a line representing the middle of the space between the edges of lower jaw and upper jaw teeth. It was tested on a dataset of 94 different pantomographs. The effectiveness of separate ROI definition reached 81.9%, while teeth line finding – 34%. There was only 5.3% of images (5), where the successful line finding was coupled with incorrect ROI definition. The completely incorrect results were investigated regards the reason for errors and direction of the further study. In the next stage of this investigation, segmenting and numbering of the individual teeth, followed by filing out the dental report/treatment plan will be considered.

**Acknowledgement.** This study was supported by the 1PW13 grant from Nalecz Institute of Biocybernetics and Biomedical Engineering Polish Academy of Sciences.

We are grateful to the NZOZ Centrum Leczenia Wad Zgryzu (Luczek St, Warsaw) for sharing the radiographs.

## References

1. Farge, M., Schneider, K., Pannekoucke, O., van yen, R.: Multiscale representations. In: Handbook of Environmental Fluid Dynamics, vol. 2, pp. 311–332. Informa UK Limited, December 2012
2. Jain, A.K., Chen, H.: Matching of dental x-ray images for human identification. Pattern Recogn. **37**(7), 1519–1532 (2004)
3. Lanman, D.: Bilateral filtering (2006). https://uk.mathworks.com/matlabcentral/fileexchange/12191-bilateral-filtering
4. Li, S., Fevens, T., Krzyżak, A., Jin, C., Li, S.: Semi-automatic computer aided lesion detection in dental x-rays using variational level set. Pattern Recogn. **40**(10), 2861–2873 (2007)
5. Lin, P., Huang, P., Cho, Y., Kuo, C.: An automatic and effective tooth isolation method for dental radiographs. Opto-Electron. Rev. **21**(1), 126–136 (2013)
6. Lin, P., Huang, P., Huang, P., Hsu, H., Chen, C.: Teeth segmentation of dental periapical radiographs based on local singularity analysis. Comput. Methods Programs Biomed. **113**(2), 433–445 (2014)

7. Lin, P., Lai, Y., Huang, P.: An effective classification and numbering system for dental bitewing radiographs using teeth region and contour information. Pattern Recogn. **43**(4), 1380–1392 (2010)
8. Mlosek, K.: Radiologia Stomatologiczna i Szczekowo-Twarzowa. MEDDENT-PRESS (1995)
9. Nomir, O., Abdel-Mottaleb, M.: A system for human identification from x-ray dental radiographs. Pattern Recogn. **38**(8), 1295–1305 (2005)
10. Oliveira, J., Proença, H.: Caries detection in panoramic dental x-ray images. In: Tavares, J.M.R.S., Natal Jorge, R.M. (eds.) Computational Methods in Applied Sciences, pp. 175–190. Springer, Heidelberg (2010)
11. Said, E., Nassar, D., Fahmy, G., Ammar, H.: Teeth segmentation in digitized dental x-ray films using mathematical morphology. IEEE Trans. Inf. Forensics Secur. **1**(2), 178–189 (2006)
12. Shah, S., Abaza, A., Ross, A., Ammar, H.: Automatic tooth segmentation using active contour without edges. In: 2006 Biometrics Symposium: Special Session on Research at the Biometric Consortium Conference. Institute of Electrical and Electronics Engineers (IEEE), September 2006
13. Stelt, P.: Filmless imaging. J. Am. Dent. Assoc. **136**(10), 1379–1387 (2005)
14. Versteeg, C., Sanderink, G., van der Stelt, P.: Efficacy of digital intra-oral radiography in clinical dentistry. J. Dent. **25**(3–4), 215–224 (1997)
15. Wanat, R., Frejlichowski, D.: A problem of automatic segmentation of digital dental panoramic x-ray images for forensic human identification. In: Proceedings of CESCG 2011: The 15th Central European Seminar on Computer Graphics (non-peer-reviewed) (2011)
16. Wang, C.W., Huang, C.T., Lee, J.H., Li, C.H., Chang, S.W., Siao, M.J., Lai, T.M., Ibragimov, B., Vrtovec, T., Ronneberger, O., Fischer, P., Cootes, T.F., Lindner, C.: A benchmark for comparison of dental radiography analysis algorithms. Med. Image Anal. **31**, 63–76 (2016)
17. Wenzel, A., Gröndahl, H.: Direct digital radiography in the dental office. Int. Dent. J. **45**(1), 27–34 (1995). http://europepmc.org/abstract/MED/7607741
18. Zhou, J., Abdel-Mottaleb, M.: A content-based system for human identification based on bitewing dental x-ray images. Pattern Recogn. **38**(11), 2132–2142 (2005)

# Color Independent Quality Assessment of 3D Printed Surfaces Based on Image Entropy

Krzysztof Okarma(✉) and Jarosław Fastowicz

Department of Signal Processing and Multimedia Engineering,
Faculty of Electrical Engineering,
West Pomeranian University of Technology, Szczecin,
26. Kwietnia 10, 71-126 Szczecin, Poland
{okarma,jfastowicz}@zut.edu.pl

**Abstract.** The paper is focused on the issue of visual quality assessment of 3D printed surfaces which can be helpful in detection of quality decrease during the printing process as well as the quality inspection of previously printed objects. The basic assumption of the proposed approach is the fact that each distortion of the regular patterns, visible on the side surfaces of objects printed using Fused Deposition Modeling (FDM) technology, causes the increase of the local image entropy. However, due to different colors of the filaments used in our experiments, a reliable prediction of the absolute entropy values can be troublesome. The proposed solution utilizes the combined quality indicator based on the entropy and its variance calculated for the hue component, as well as for the RGB channels, depending on the color of the filament, allowing proper detection of low quality surfaces regardless of the filament's color.

**Keywords:** 3D prints · Entropy · Image quality assessment · Image analysis

## 1 Introduction

During several recent years various directions of progress in image quality assessment (IQA) methods can be observed. Nevertheless, some challenges in this area of science remain still up-to-date [3]. One of the most relevant problems is the assessment of natural images, another one is related to the presence of multiple distortions in images. Important issues correspond to run-time performance and memory requirements of algorithms and assessment of enhanced images as well. Such list can also be supplemented with the assessment of the HDR and 3D images as well as some other research challenges. Recently, a rapid progress has been made in the assessment of 3D images and several databases containing 3D images with subjective quality scores have been developed which are useful for verification of new proposed quality metrics. Some of such databases are: IVP Anaglyph Image Database, IRCCyN/IVC 3D Image Quality

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_32

Database [2], LIVE 3D Image Quality Database [5,6] or MMSP 3D Video Quality Assessment Database [10]. Many new ideas, often presented together with the development of 3D IQA databases, are devoted to stereopairs [11,12,21], also for synthesized views [1]. Some other applications are based on Depth-Image-Based-Rendering (DIBR) techniques used for free-viewpoint television purposes [17]. One of the currently unexplored fields of research related to IQA is the visual quality assessment of 3D prints. Such methods can be useful during monitoring of the 3D printing process in order to save the filament using the emergency stop in case of decreased quality. Another possibility is the visual inspection of already printed objects which can be made analyzing the consistency of the printed surfaces.

Considering the quality of 3D prints, the accuracy of object's 3D shape can be assessed by comparison of the detected edges with the edges extracted from the 3D model. Such an approach is quite similar to model based 3D visual servoing used for navigation of mobile robots [13]. Nevertheless, it requires the knowledge of the 3D model and in many cases distortions of the printed surfaces, which do not influence the external shape of the object can be omitted.

Due to the growing availability and popularity of relatively cheap 3D printers for home use, several factors influencing the final quality may play important role, causing noticeable surface distortions. According to our best knowledge, currently there are no comprehensive solutions for visual quality assessment of 3D prints and therefore a reliable method for classification of low and high quality prints is necessary.

## 2 Idea of the Entropy Based Surface Quality Assessment

The idea of use of computer vision and image analysis methods for monitoring the 3D printers has been presented in several earlier papers. Fang [8] has proposed on-line defect detection for fused deposition of ceramics whereas monitoring the top surface of the print based on fuzzy model has been described by Cheng [7]. The latter approach is based on the comparison of adjacent layers for identification of over- and under-filling during the printing process. Some other approaches are based on neural networks [20] or detection of known faults such as part jams and feeder jams implemented in LabVIEW [19]. Some other applications of machine vision inspection for fault detection and classification are presented in Chauhan's paper [4] and an interesting initial approach to automatic correction of detected errors during printing is presented by Straub [18]. This system is based on Raspberry Pi modules and five cameras, allowing proper detection of "dry printing" caused by the lack of filament. Nevertheless, its most relevant disadvantages are many required interruptions during the printing process, necessity of precise calibration, and sensitivity to changing lighting conditions as well as camera motions.

In some of our earlier papers related to the quality assessment of 3D prints we have focused on the analysis of the Gray Level Co-occurrence Matrix (GLCM) and chosen Haralick features [9,14] as well as some of the most promising IQA methods such as Feature Similarity [15] and Structural Similarity based metrics [16]. Obtained results are promising but results are dependent on the color to grayscale conversion method since all the above mentioned methods are typically applied for grayscale images. Since the ideal surface of the 3D print obtained using Fused Deposition Modeling (FDM) technology, observed by the side view camera, should contain regular linear patterns representing consecutive layers of the filament, the entropy of the image representing such a view should be relatively low. Each distortion of the structure visible on the image causes the increase of image entropy which is expressed as:

$$E = -\sum_{i=1}^{N} p_i \cdot log_2(p_i) \tag{1}$$

where the vector $p$ contains the histogram counts calculated assuming $N = 256$ bins for grayscale images. Entropy can be therefore considered as the statistical measure of randomness and interpreted as the measure of information amount. For corrupted patterns the amount of visual information is higher due to complications of the structures visible on the images. Therefore, relatively high correlation with perceived surface quality may be expected.

In order to minimize the impact of changing lighting conditions and verify the correctness of the proposed approach all test images have been obtained using a flatbed scanner assuming the scanning direction perpendicular to the visible pattern lines representing filament layers. The scanned 3D prints are flat plates printed using bio-degradable polylactic acid (PLA) filaments and two available 3D printers, namely Prusa i3 and RepRap Pro Ormerod 2. Some of such obtained images are presented in Fig. 1. Low quality samples (upper row) have been printed by forced changes of filament's delivery speed and temperature.

As the image entropy can be computed locally for the fragments of images, its potential application for on-line quality monitoring of the printed surfaces seems to be obvious. Nevertheless, one of the disadvantages of such approach is its sensitivity to filament's color as well as color to grayscale conversion method. Some exemplary results obtained using image entropy for different images (scans of 3D printed plates) are presented in Fig. 2.

Image entropy has been calculated for the hue component in HSV color space. All the high quality scans are marked with "**x**" and low quality ones with "**o**". First 4 samples have been obtained for white filament, samples 5–8 for the red one whereas 9 and 10 are for orange, 11–14 for silver, and 15–18 for a green filament. The same marks and order of samples have also been used in all plots in the paper.

Analyzing the results illustrated in Fig. 2, a strong influence of color on the absolute entropy value can be easily noticed. Although the entropy calculated for low quality samples is always higher in comparison to high quality prints obtained for the same filament, the prediction of the expected entropy can be troublesome. A practical implementation of such approach would require a calibration of entropy values for different colors since e.g. entropy equal to 1.5 may indicate low quality white or red sample but high quality green or silver print.

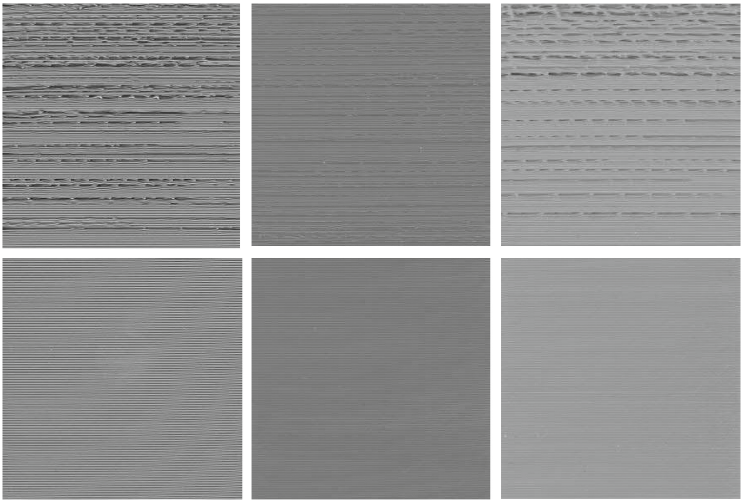


**Fig. 1.** Exemplary scans of the low and high quality 3D prints used in experiments (from left to right: silver, red, and green samples converted to grayscale)

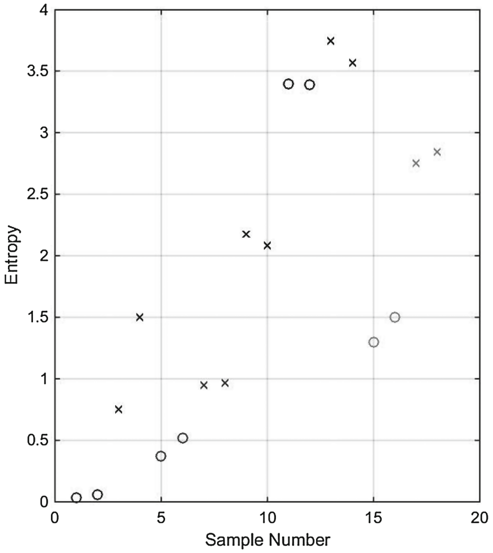


**Fig. 2.** Exemplary entropy results obtained in experiments for hue component

## 3 Color Independent Surface Quality Assessment

In order to overcome the problem related to color dependency of the image entropy values, an approach based on the combination of the local entropy and its variance is proposed. The motivation for such approach is based on the assumption, that from one side low entropy values are usually related to high quality prints, and on the other hand, regardless of these values, high variance of the local entropy is characteristic for degraded samples.

Since the image entropy can be calculated locally for the fragments of images, each sample has been divided into 16 parts of the same size. For each of such obtained blocks the local entropy values have been computed in order to determine the variance of entropy. Such obtained variance has been multiplied by the mean image entropy in order to achieve the new combined surface quality indicator.

Aiming to obtain the results independent on the color of the filament, several popular color spaces have been tested as well as several color to grayscale conversion methods. The most promising results have been obtained for the hue component obtained after conversion from RGB to HSV color model.

The results obtained for the proposed quality indicator EV are presented in Fig. 3. It can be easily observed that the values of such entropy and variance based metric EV higher than 0.04 occurs only for low quality prints. The only exception is the red filament and the expected reason for such a phenomenon may be the circular nature of the hue component where the purely red color is denoted as 0 and similar colors may be represented as small numbers close to 0 or large numbers close to 360 (or another maximum value depending on the data type used for storing the hue angular values).

On the other hand, much better results for the red filament (samples 5–8) can be obtained using the independent calculation of the proposed EV indicator for RGB channels forwarded by averaging of such obtained three values. Such obtained results for the same samples are illustrated in Fig. 4. In this case the improper detection can be observed for achromatic samples (white and silver) but the classification of red, green and orange 3D prints is correct. The dynamic

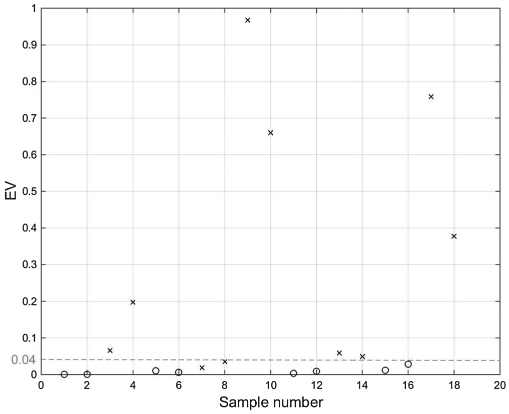


**Fig. 3.** Results of the combined EV indicator obtained experimentally for hue component

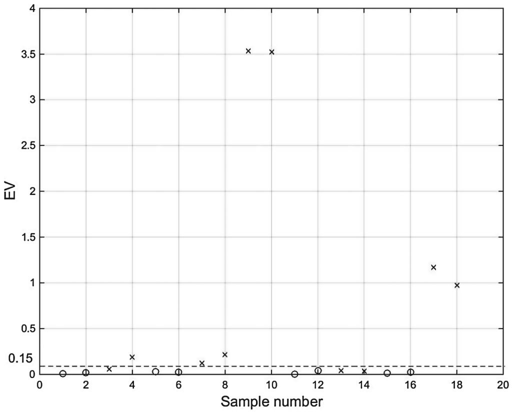


**Fig. 4.** Results of the combined EV indicator obtained experimentally for RGB channels after averaging

range of obtained results is different and therefore the threshold value for the classification of 3D prints according to surface quality should be set to 0.15.

The last step of the experiments has been the development of the indicator combining the advantages of two attempts described above based on hue and the average EV values for RGB channels. The proper classification with results illustrated in Fig. 5 of the high and low quality surfaced of 3D prints can be made using the natural logarithm of the product of two EV indicators:

$$EV_{combined} = ln(EV_{RGB} \cdot EV_{hue}) \tag{2}$$

As can be observed in Fig. 5, the discrimination of high and low quality 3D printed surfaces can be easily made by setting the threshold value of the cal-

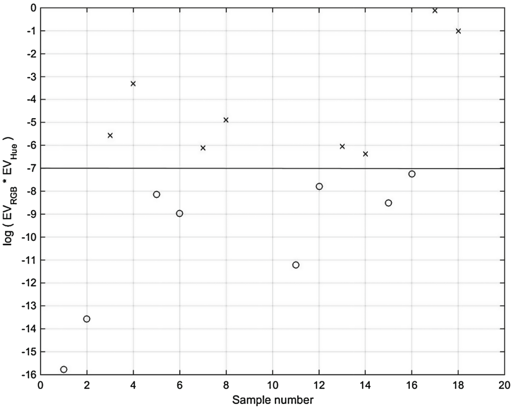


**Fig. 5.** Results obtained using the proposed combined EV indicator

culated metric to −7. Such computed indicator allows proper classification of the images representing scanned surfaces of 3D prints regardless of the filament's color.

## 4 Concluding Remarks

Proposed method of quality assessment of 3D printed surfaces independently on the filament's color can be useful for various purposes related to monitoring of the 3D printing as well as quality inspection of already printed elements. Presented approach has some limitations related to the assumptions made at the beginning of conducted experiments e.g. the use of scanned images. Nevertheless, in our opinion, it may be a good starting point for the development of more advanced methods useful also for some other types of 3D printers and materials. However, experimental verification of such approach requires the use of different types of printers e.g. based on stereolithography being one of the directions of our further research.

In the nearest future it is planned to adopt the presented approach to the assessment of the 3D printed surfaces using the natural images captured directly by cameras in different lighting conditions instead of scans.

## References

1. Battisti, F., Bosc, E., Carli, M., Callet, P.L., Perugia, S.: Objective image quality assessment of 3D synthesized views. Signal Process. Image Commun. **30**, 78–88 (2015)
2. Benoit, A., Le Callet, P., Campisi, P., Cousseau, R.: Quality assessment of stereoscopic images. EURASIP J. Image Video Process. **2008**(1), 659024 (2008)
3. Chandler, D.M.: Seven challenges in image quality assessment: past, present, and future research. ISRN Signal Process. **2013**, 53 (2013)
4. Chauhan, V., Surgenor, B.: A comparative study of machine vision based methods for fault detection in an automated assembly machine. Procedia Manufact. **1**, 416–428 (2015)
5. Chen, M.J., Cormack, L.K., Bovik, A.C.: No-reference quality assessment of natural stereopairs. IEEE Trans. Image Process. **22**(9), 3379–3391 (2013)
6. Chen, M.J., Su, C.C., Kwon, D.K., Cormack, L.K., Bovik, A.C.: Full-reference quality assessment of stereopairs accounting for rivalry. Signal Process. Image Commun. **28**(9), 1143–1155 (2013)
7. Cheng, Y., Jafari, M.A.: Vision-based online process control in manufacturing applications. IEEE Trans. Autom. Sci. Eng. **5**(1), 140–153 (2008)
8. Fang, T., Jafari, M.A., Bakhadyrov, I., Safari, A., Danforth, S., Langrana, N.: Online defect detection in layered manufacturing using process signature. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, vol. 5, San Diego, California, USA, pp. 4373–4378 (1998)
9. Fastowicz, J., Okarma, K.: Texture based quality assessment of 3D prints for different lighting conditions. In: Chmielewski, L.J., Datta, A., Kozera, R., Wojciechowski, K. (eds.) ICCVG 2016. LNCS, vol. 9972, pp. 17–28. Springer, Cham (2016). doi:10.1007/978-3-319-46418-3_2

10. Goldmann, L., Simone, F.D., Ebrahimi, T.: A comprehensive database and subjective evaluation methodology for quality of experience in stereoscopic video. In: 3D Image Processing (3DIP) and Applications. Proceedings of SPIE, No. 7526 (2010)
11. Guo, J., Vidal, V., Cheng, I., Basu, A., Baskurt, A., Lavoue, G.: Subjective and objective visual quality assessment of textured 3D meshes. ACM Trans. Appl. Percept. **14**(2), 11:1–11:20 (2016)
12. Lin, Y., Wu, J.: Quality assessment of stereoscopic 3D image compression by binocular integration behaviors. IEEE Trans. Image Process. **23**(4), 1527–1542 (2014)
13. Moughlbay, A.A., Cervera, E., Martinet, P.: Model based visual servoing tasks with an autonomous humanoid robot. In: Lee, S. (ed.) Frontiers of Intelligent Autonomous Systems. SCI, vol. 466, pp. 149–162. Springer, Heidelberg (2013). doi:10.1007/978-3-642-35485-4_12
14. Okarma, K., Fastowicz, J.: No-reference quality assessment of 3D prints based on the GLCM analysis. In: 2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 788–793 (2016)
15. Okarma, K., Fastowicz, J.: Quality assessment of 3D prints based on feature similarity metrics. In: Choraś, R. (ed.) Image Processing and Communications Challenges 8. AISC, vol. 525, pp. 104–111. Springer, Cham (2016). doi:10.1007/978-3-319-47274-4_12
16. Okarma, K., Fastowicz, J., Tecław, M.: Application of structural similarity based metrics for quality assessment of 3D prints. In: Chmielewski, L.J., Datta, A., Kozera, R., Wojciechowski, K. (eds.) ICCVG 2016. LNCS, vol. 9972, pp. 244–252. Springer, Cham (2016). doi:10.1007/978-3-319-46418-3_22
17. Starch, J., Kilner, J., Hilton, A.: Objective quality assessment in free-viewpoint video production. In: Proceedings of the 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, Istanbul, Turkey, pp. 225–228 (2008)
18. Straub, J.: Initial work on the characterization of additive manufacturing (3D printing) using software image analysis. Machines **3**(2), 55–71 (2015)
19. Szkilnyk, G., Hughes, K., Surgenor, B.: Vision based fault detection of automated assembly equipment. In: Proceedings of the ASME/IEEE International Conference on Mechatronic and Embedded Systems and Applications, Parts A and B, vol. 3, Washington, DC, USA, pp. 691–697 (2011)
20. Tourloukis, G., Stoyanov, S., Tilford, T., Bailey, C.: Data driven approach to quality assessment of 3D printed electronic products. In: Proceedings of the 38th International Spring Seminar on Electronics Technology (ISSE), Eger, Hungary, pp. 300–305, May 2015
21. Yang, J., Hou, C., Zhou, Y., Zhang, Z., Guo, J.: Objective quality assessment method of stereo images. In: Proceedings of the 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, Potsdam, Germany, pp. 1–4 (2009)

# Raster Maps Search Using Text Queries and Reasoning

Arkadiusz Cacko(✉) and Marcin Iwanowski

Institute of Control and Industrial Electronics, Warsaw University of Technology,
ul. Koszykowa 75, 00-662 Warszawa, Poland
{arkadiusz.cacko,marcin.iwanowski}@ee.pw.edu.pl

**Abstract.** Search engines have become one of indispensable elements of modern information processing systems. One of challenges to face in this area is to build an interface between pictorial and text data, that would allow for image search using text queries. In the paper, a concept is proposed, that combines inference, ontology and natural language processing to formulate queries to the pictorial data of raster map images. The proposed system analyses a user's query formulated in the natural language and finds place on map which meet this criteria. The proposed approach improves remarkably the easy of access to the raster map content.

## 1 Introduction

This paper describes some results of research oriented towards an application of queries formulated in the natural language to search for the meaningful information within the digital raster image presenting a digital map. Nowadays, in the era of information, search engines became one of crucial elements of data-processing systems. Vast majority of them is however oriented towards querying the textual data: pure text, hypertext, databases etc. Due to the complexity of digital images the methods of search in this domain is much less developed. The main problem is understanding of images, which is much more complicated issue than figuring out the meaning of a text. The search for particular information in raster images is usually performed by means of text-based queries of the accompanying metadata. Such metadata does not however contain the complete information available on the raster image. If available, it consists usually of the most important facts concerning the data themselves (e.g. file name, resolution, geotag etc.). The information on the *content* of the raster image is not available. In order to look for such an information, the image analysis tools are necessary. In the contemporary world, the digital maps became one of widely used data sources. The classic way of storing the geographic information are vector images similar to those used in computer graphics. However there exist some areas where available maps are stored in the raster format. For example, in popular web-based mapping services like e.g. Google maps, the visual output in the web-browser is a raster image (bitmap) that was generated on the

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_33

server side based on the original vector maps. Consequently the data available to the end-user is stored in a raster format. In this paper we focus on map image processing in conjunction with ontology-enhanced reasoning and natural language processing (NLP). The aim of our research was to develop an interface that allow for automatic investigation digital raster maps using natural language queries. Such interface has been implemented in a system able to find answers to users' map questions asked in the natural language. The result is returned in the form of a selected area on the map that meets the search criteria. System which operates in this way has to deal with many various problems. They starts with understanding the meaning of the user intention, goes through discovering specific objects on image and relationships between them and ends with formulating the system's answer which meets the original user query. Natural language understanding provides the basis for the discovery search criteria. This leads to information about which data needs to be acquired by a system. The data in conjunction with ontology is then converted to metadata which is used in inference operations. The paper consists of 5 sections. The current section contain introduction. Section 2 presents a related works' review of the literature. Section 3 contains description of the proposed methods used in whole query processing. The system proposal and example results are presented in Sect. 4. Finally, conclusions drawn from the experiments are presented in Sect. 5.

## 2 Related Works

Information becomes more valuable when combines various data types. Owing to this fact, it is important to find a way to integrate various information sources. One of the possible solutions is based on ontologies. They represent the structure of information included in the text data. On the other hand, a lot of information that relates to the content of a digital image, may be formulated using some ontological structure that describes both form and position of various elements of the visual scene. Ontology is an explicit formal specifications of the terms in the domain and relations among them. It is very useful especially during the construction of human-computer interfaces, because the origin of ontologies has something common with human experience. It allows also to explore digital images starting from the text query. Such approach leads to semantic annotation methods [5,7] where a search tool based on photos combined by semantic relationships were proposed. The solution that is investigated in this paper is based on image analysis combined with exploring external source of knowledge and inference. In order to provide with the easy-to-use human-computer interface, the above approach must be supported with NLP methods. As [3] shows, combining NLP with visual processing brings benefits in quality of search system performance. PROLOG as a powerful logic programming language is solution to deal with this problem [1,4]. Simple features extracting from maps combined with PROLOG reasoning rules based on ontology, brings much more valuable results than trivial approach where system tries to find answer from data visible directly in the map. As creating PROLOG queries is difficult for users, we propose a way of formulating image search queries in the natural language.

Such a question need to be converted to useful PROLOG predicates. To improve quality of system response, ontology is applied. In this way system has better ability to understand user questions formulated in Natural Language.

## 3 Methodology

### 3.1 Raster Maps

Map is a reflection of the real world. It includes simplified two-dimensional image of land cover consisting of objects that are in spatial relationship one with another. Modern digital maps are stored in a vector-format where particular regions on the ground are represented by polygons or other computer-graphics primitives. The vector-format has multiple advantages over thee raster-format, among which the most important is the ease of scaling of the vector data. There exist however some area where the digital maps, originally stored in the vector-format has been rasterized. One of them is displaying the map. Some of state-of-the-art map services are displaying the map as a raster image that has been downloaded from the server of a mapping service. Google Maps is one of such services. There exists thus some areas where the raster map format is the only one that is available. In the proposed approach we start from maps that are kept as a raster data where entire area of the map is subdivided into a grid of tiny cells. Raster data can be thought of as a matrix of values which are stored in each of these cells (label) to represent the nature of whatever is present at the corresponding location on the ground. In the rasterized maps, the labels describing the land-cover classes are related to image pixel values. Examples of such labeled objects can belong to any spatial class like forest, water, mountains. Apart from the extraction of such objects from the map, the system should also be able to discover shape of objects, their positions or distances between them. Having only valuable objects on image, system performs the analysis of positions of all objects within the scene [2]. This process may produce some image descriptors which can be used to find an answer to the user query.

### 3.2 Ontology and Reasoning

Combining knowledge from different sources always brings benefits. In case of map processing it may, for example, enrich the understanding of complex names of searched places. For example, the user can include in his/her search query the word “coast” what means that the system needs to find – using the appropriate ontologies – a region of the map, where water and land touch each other. Another example would be the way of recognizing the sense of a word describing mutual relations between objects in query sentence. In such a case, the knowledge is required, how to interpret words like “near”, “on the left” etc. in the particular case of rasterized map. System based on the ontology combines information referring directly to the image under study with external data sources. There are many types of ontologies and many ways to use it. In the current approach, ontologies are formulated as inference rules. Thanks to this fact, the possible relations between elements of a map – land-cover classes and objects – can be represented by a set of rules. In our approach rules was formulated in the form of PROLOG predicates. These predicates represent both properties of objects and relations between them [2]. They can be divided into two groups: low-level and hi-level facts. The low-level facts are discovered by means of image analysis tools. They derive directly from the image content. In the ontology rules they always belong to premises. The hi-level facts are, in turn, validated using rules based on both lo- and other high-level fact. They are always conclusions of rules, but they may also belong to premises. The low-level facts are evaluated by the computer-vision algorithm that finds objects within the image, extracts their features and evaluates also spatial relationships between them. Output of the image processing algorithm are PROLOG low-level facts about objects recognised in image – information describing each object separately. In the proposed approach, two types of low-level facts are introduced: attribute and position predicates. At first, every object is represented as a separate attribute predicate (*attr*) which contains basic information about each object extracted from the image. Such a information may be any useful discovered features of this object:

attr(idOfObject, typeOfObject, areaInPixelsInMap, ...otherFeatures).

For example, the following attribute predicates:

attr(1,'forest',49738).
attr(2,'water',54432).

describe two objects, the first is represented by a label '1' on a rasterized map, refers to the forest regions on a ground, and has the area of 49738 pixels. The second object is represented by the label '2', refers to waters and has the area of 54432 pixels. Apart from attribute predicates the position predicate (*pos*) are introduced. They contain an information on the mutual relations between objects already described using the attribute predicates. PROLOG predicates of position of each object also contains necessary data about any two objects on image:

pos(baseObject, relatedObject, relationType, distanceInPixels).

An example position predicate:

pos(1, 2, right, 200).

describes the relation between image region of label '1' (that is, according to the attribute predicate, a forest) and region of label '2' (water). This relation is the 'location on the right-hand side' i.e. the water is on the right from forest at distance of 200 pixels. Based on the low-level facts, by using PROLOG rules is is a possible to formulate ontologies as rules. For example, the word "between" – may be formulated as the predicates:

```
between(1,X,2):-pos(1,X,right,_),pos(2,X,left,_).
between(1,X,2):-pos(1,X,up,_),pos(2,X,down,_).
```

Using this mechanism, we can create PROLOG predicates for every word describing spatial relation. In a similar way other spatial relations may be defined. An application of PROLOG to formulate ontologies for spatial object relations is universal and can be applied with various image processing systems. Image processing methods that allow for evaluating the low-level facts has been described in [2]. All the external data which lead to discover not obvious (that cannot be derived directly from the image content) knowledge about map content is valuable and enrich user query processing in whole system.

### 3.3 NLP and Query Analysis

The reasoning and ontologies combined with image analysis allow to formulate queries related to the image content. However such an approach requires the user to know basic rules of creating PROLOG queries. In order to further facilitate human-computer interaction, PROLOG queries are automatically created using natural language processing (NLP), so that even a user who is not familiar with PROLOG may interact with the system using natural language. This is especially important when there is a need to find information that is not structured and there is not possible to create strict language of creating queries to databases. In case of map processing user describes what he/she wants to find. Often user doesn't know exactly the name of the given object but can give some criteria which the object should meet. Example of search question would be "Find hospital in Warsaw close to highway and is less than 500 m from river". Current map systems do not allow to ask these types of questions, although in many situations it might be very helpful. In order to translate the user query formulated in the natural language into the PROLOG predicate, the natural language processing tools are applied. It starts when the user formulates a question to the map. NLP processing finds words in sentence which can be candidates to map objects, separates criteria that each object should meet as well as relations between them. In addition, selected word/words should allow to localize the search area, it plays a role of a focal point. This mechanism leads to information on objects that should be localized and conditions that they must meet to find final answer to the user query. In order to interpret the text query formulated in the natural language, at the beginning it is tokenized (treebank tokenization) – the sentence is split into separated words. Next step is the process of marking up a word in a text as corresponding to a particular part of speech called word-category disambiguation (POS tagging) [6]. Tagged words are useful source of information for determining objects to find and relationships between them. To increase the quality of processing the lemmatization is performed to reduce different forms of a word to its primitive. It simplifies the mechanism of finding objects and relations between them in analysed sentence. Finding objects names is done by locating nouns in a query. Criteria are specified through the analysis of the other words that are between word objects.

### 3.4 Creating Map Query

Goal of the Map Service is to provide image of map containing only interesting objects which are needed in solution finding process. Google Maps allows for showing of chosen map labels. These labels are selected based on finding and recognising nouns in query. Some of sentence nouns equal names of Google Map API labels (forest, water, road), some need to be mapped into proper one ("lake" should be transformed to "water") and other must to be presented by multiple labels ("coast" should be represented as a combination of "water" and "terrain"). All this labels need to be presented (showed) in analysed map because search algorithm must operate on this data. Other type of available objects may be hidden because are not necessary to find the answer to the user question. Each map must be positioned in a particular place in the world – the focal point. In our system it is achieved by finding proper nouns in a sentence. These nouns are candidates to find name of a place where the user want to start the search of objects/spatial classes. To center map on this point, system needs to use geocoding service to translate i.e. name of city to latitude and longitude. Using this information it is possible to center map on this place and try to visualise labels which will lead us to find answer. On the basis of analysing user queries, finding objects appearing in the sentence and their counterparts in the map data, finding relations between them, the system can produce data query formulated in PROLOG which finds the answer (as a map with selected area/place).

### 3.5 System Proposal

Input data for whole framework is string query formulated in natural language. System analyse and process query, use external ontologies and map systems to produce result as map with selected object/area which meets user search criteria. Diagram showing the data flow is presented in Fig. 1. Starting from a given question, system separate and analyse meaning of the words. Nouns in sentence are considered as potentially meaningful objects on the map. 'Proper nouns' are candidates to define the focal point of the map analysis. Separated objects from the sentence – with use of ontology – are transformed to request to map provider. Ontology is used to explain meaning of the word in the context

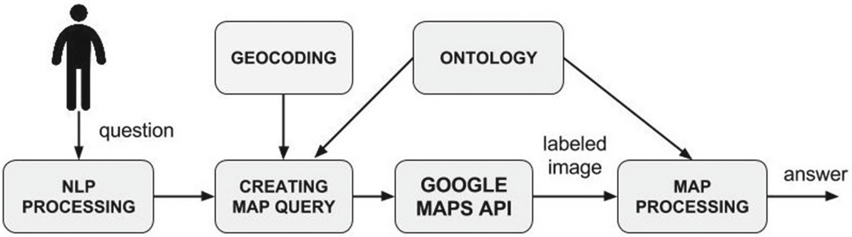


**Fig. 1.** Simplified algorithm graph

of map data. For query sentence: “Find forest near to the airport and between main road and airport in Bydgoszcz” objects are “forest”, “airport” and “main road”. “Bydgoszcz” is the focal point – a center of area of interests. Geographical coordinates of such area (eg. city) are found using geocoding system. Request to the map provider should return the map that contains only the required objects, centered on the focal point (Fig. 2). Thanks to label selecting, map contains only valuable types of objects. Each label can be separated by simple image segmentation of the label bitmap received from the map provider. All objects are converted to PROLOG predicates with additional descriptions and mutual positions between objects (according to algorithm presented in [2]). Thanks to the NLP and ontology processing – search criteria are extracted from the user query. They are next are transformed to the PROLOG search query further used in the inference process. Final result of the map processing is single complex PROLOG query. It allows to find the object that meets all user assumptions included in the question. If such a result exists – it is presented in standard map with highlighted object (Fig. 2b).

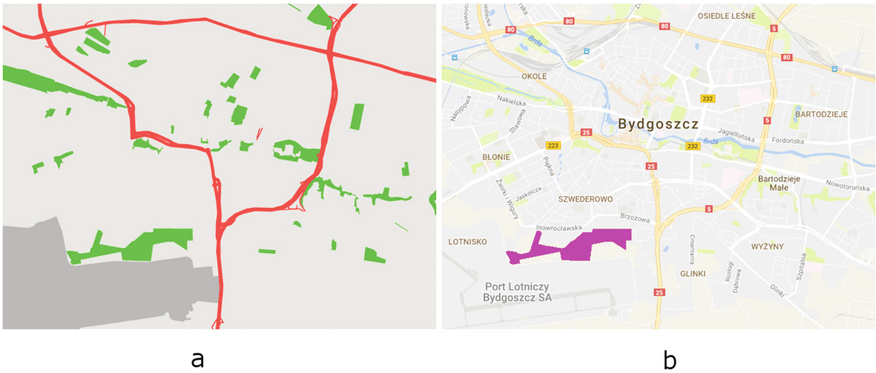


**Fig. 2.** Query: “Find forest near to the airport and between main road and airport in Bydgoszcz”: (a) Returned objects: grey – airport, red – main roads, green – forests, (b) Final result (pink color)

## 4 Examples and Tests

Final result of system question processing is an image of map with selected object. For a search query i.e. “Find greenery less than 100 m from coast in Puck” all steps are described below. NLP tools annotate every word in sentence extracting such information like part of speech (Table 1). Words: “greenery”, “coast” – are nouns which are candidates to map objects which should be used in the search process. “Puck” is a ‘Proper Noun’ considered as a name of city/place where system should start searching – the focal point. Using geocoding utilities

**Table 1.** Tagged question sentence

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Find | greenery | less | than | 100 | m | from | coast | in | Puck |
| Verb | Noun | Adjective | Preposition | Number | Noun | Preposition | Noun | Preposition | Proper Noun |

(i.e. Google Geocoding API) it is possible to find the exact geographic coordinates of this place/area. Words: "less than", "from" give information about relative position between objects found in sentence. "100 m" is an argument of this mutual position. Since system doesn't know what is "greenery" and "coast" – ontology needs to be applied. Ontology gives information that "greenery" may be forest or park. "Coast" is a place close to water. Thanks to this information system formulate query to Map Provider to show objects in the specific area which are only in a domain of "forest, park, water". Results are in a form of image where different colors represent other type of terrain. Such image can be labeled thanks to color segmentation (Fig. 3a). Having separated regions in the image – the algorithm starts to analyse the form of particular objects that results in attribute predicates. Next, mutual positions between all objects on the image in order to evaluate the position predicates. Every separated object on a map is thus represented as a one predicate with some features which were extracted from image (*attr*). Mutual positions between every pair of object is also generated as a PROLOG predicate (*pos*). The final result of this step is a set of PROLOG attributes and position predicates – low-level facts – presented in Sect. 3.2. All of this is complemented by the meaning of words from the ontology defined also as PROLOG predicates. Process of finding the final answer is based on data produced using the following computation steps:

- Tokenize words in sentense
- Find geographical coordinates of search area
- Select from sentence various classes of map objects (also using ontology)
- Extract from sentence mutual dependencies between objects
- Analyse map to produce PROLOG *attr* and *pos* about all objects found in map image

Final result – answer to the user question – is achieved by using the PROLOG query about predicates which was produced in last step. Since all information is available about objects and mutual positions between them in form of predicates, knowledge about which word in sentence represents the object on the map, and finally transformed search criteria from sentence into PROLOG rules – there is possibility to create one query to find what user asks. This query gives a result which object on map meets criteria formulated in a sentence. If such object exists – there is produced map with found object selected (Fig. 3b). Another example is presented in Fig. 4.

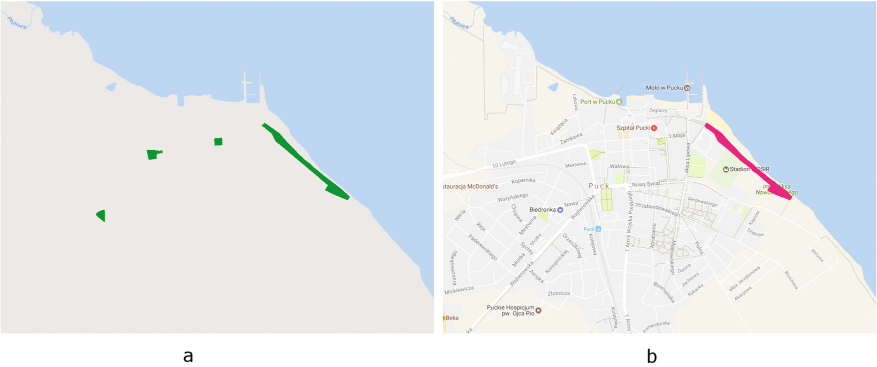


**Fig. 3.** "Find greenery less than 100 m from coast in Puck": (a) Image with selected objects (water, area, forest), (b) Final result (pink color)

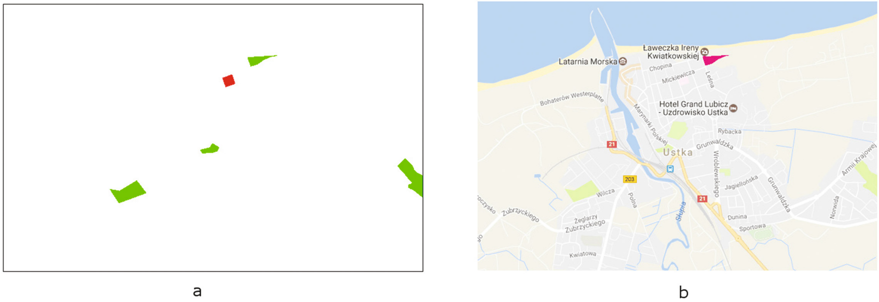

Latarnia Morska
Ławeczka Ireny Kwiatkowskiej
Hotel Grand Lubicz - Uzdrowisko Ustka
Ustka
a
b

**Fig. 4.** Example of query: "Find park in east of a hospital in Ustka": (a) Analyzed objects, (b) Answer (pink color)

## 5 Conclusions

Human-machine interface is a very important part of every computer system that interact with humans. Often underestimated, but necessary when end-user in not familiar with complex system processing domain. That's why NLP and ontology is valuable supplement of the image processing. In the paper, a concept was proposed, that combines inference, ontology and natural language processing to formulate queries to the pictorial data of raster map images. The proposed system analyses a user's query formulated in the natural language, use NLP tools and ontology to find objects in sentence and relation between them. Afterwards using external services offered by digital map providers, system acquires raster map with only such elements which are meaningful in context of search process. Image analysis in conjunction with reasoning finds place on map which meet user criteria. The results of experiments that were obtained are promising and shows the high potential of the proposed approach. The latter improves remarkably the easy of access to the raster map content. It may also be extended to other kinds of digital images.

## References

1. Batchelor, B.G.: Intelligent Image Processing in Prolog. Springer, London (1991)
2. Cacko, A., Iwanowski, M.: Evaluating the mutual position of objects on the visual scene using morphological processing and reasoning. In: Choraś, R.S. (ed.) Image Processing & Communications Challenges 6. AISC, vol. 313, pp. 13–20. Springer, Cham (2015). doi:10.1007/978-3-319-10662-5_2
3. Cantrell, R., Krause, E., Scheutz, M., Zillich, M., Potapova, E., et al.: Incremental referent grounding with NLP-biased visual search. In: Proceedings of AAAI 2012 Workshop on Grounding Language for Physical Systems (2012)
4. Gal, A., Lapalme, G., Saint-Dizier, P., Somers, H.: Prolog for Natural Language Processing. Wiley, Chichester (1991)
5. Hollink, L., Schreiber, G., Wielemaker, J., Wielinga, B., et al.: Semantic annotation of image collections. In: Knowledge Capture, vol. 2 (2003)
6. Merialdo, B.: Tagging English text with a probabilistic model. Comput. Linguist. **20**(2), 155–171 (1994)
7. Schreiber, A.T., Dubbeldam, B., Wielemaker, J., Wielinga, B.: Ontology-based photo annotation. IEEE Intell. Syst. **16**(3), 66–74 (2001)

# Nuclei Recognition Using Iterated Conditional Modes Approach

Marcin Skobel(✉), Marek Kowal, and Józef Korbicz

Institute of Control and Computation Engineering, University of Zielona Góra,
ul. Szafrana 2, 65-516 Zielona Góra, Poland
{M.Skobel,M.Kowal,J.Korbicz}@issi.uz.zgora.pl

**Abstract.** The paper presents Iterated Conditional Modes based method for nuclei recognition in cytological images. It approximates nuclei by circles and ellipses. The first step is to find coordinates and sizes of circles. To find good configuration of circles, Iterated Conditional Modes (ICM) approach is employed to maximize the probability of configuration given image data. However, nucleus shape appears to be more elliptical than circular. Unfortunately, the process of finding nuclei using ellipses is computationally expensive, because at one point ellipses have three parameters (minor axis, major axis and angle), while circle have only one parameter (radius). To tackle this problem, we proposed heuristic procedure to estimate ellipses based on previously determined circles. To test the effectiveness of the method, it was applied to recognize disks in synthetically generated images and nuclei in microscopic images of breast cancer tissue.

**Keywords:** Stochastic geometry · Nuclei recognition · Maximum-likelihood estimation · Iterated conditional modes · Marked point process

## 1 Introduction

Recent advances in digital microscopic imaging techniques gave a rise to quantitative cytopathology. More and more pathologists examine virtual slides on a computer screen instead of reviewing glass slides under a microscope. So, their work can be supported by image processing and machine learning algorithms [8,15]. In order to distinguish cancer cases, pathologists evaluate morphometric parameters of nuclei or cells. However, the amount of image data they need to review is huge because virtual slides can have size up to 200000 × 200000 pixels. To help them in this task, we need accurate algorithm of nuclei and cell segmentation. However, if we look at cytological specimen under a microscope, we will see a lot of clumps, which create complex, random and heterogeneous structures similar to nests. Their segmentation is rather a challenging task for existing segmentation methods. Especially, cytological images can be very challenging due to many overlapping nuclei. A number of scientific centers (e.g.: Department of

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_34

Biosystems Science and Engineering in ETH Zurich; College of Communication Engineering of Chongqing University; Institute of Pathology, Charité University Hospital Berlin) conduct an intensive research to develop accurate algorithms for cell and nuclei segmentation. The most common approaches are primarily based on intensity thresholding, feature detection, data clustering, region growing, deformable models and morphological mathematics [5–7,11,13]. Unfortunately, the most prospective methods are usually sensitive to nuclei seed initialization. The issue of nuclei seed generation is still open and exist a need for improved solutions in this area. Stochastic geometry is a branch of probability theory that deals with the analysis of random spatial patterns [3,4]. They are successfully applied wherever there are heterogeneous structures whose spatial distribution is random [12]. We can also observe such distribution for biological materials in microscopic images. Therefore, we can expect that stochastic models based on marked point processes can accurately determine the location and size of nuclei. Based on this information, seeded segmentation methods will be able to accurately segment nuclei. In this work, we limit our researches and discussion to the issue of nuclei detection in cytological images. We propose a two-stage procedure to determine the location and size of nuclei. Firstly, we build intensity model of nuclei and background based on training data from images [14]. Next, circular object recognition procedure was formulated as a maximum likelihood estimation task [1,9,10]. We applied ICM method in the form of Steepest Ascent and Coordinatwise Optimization algorithms to find the configuration of circles which maximizes likelihood [1,2,9,10]. However, circular approximation of nuclei is not always accurate because nuclei have usually elliptical shape. The undesired effect of such recognition is that several similar circles can approximate single nucleus. To tackle with this problem, the second stage of nuclei recognition is applied to extract elliptic objects based on the result of circular approximation. Simple clustering algorithm is applied to reveal circles lying close to each other. Then, for each cluster, line is fitted to circle centers. It represents major axis of ellipse. The length of this axis is calculated from two most distant intersections between line and circles. Similar procedure is applied to determine minor axis length. Finally, we get configuration of ellipses which represent nuclei. In order to verify the proposed method, it was applied to detect disks in artificially generated images and then to cytological images of breast cancer. The remainder of this paper is organized as follows. In Sect. 2, material used to experiments is presented. Section 3 gives the description of methods applied to recognize nuclei. Results of experiments are presented in Sect. 4 Concluding remarks are given in Sect. 5.

## 2 Materials

Two types of intensity images were used to carry out experiments to verify the effectiveness of the proposed method: synthetic images of disks and real microscopic images of nuclei. In the preliminary studies, we used artificially generated images. Each synthetic image includes a number of disks of the same or different size. Both, foreground pixel intensities (disks) and background pixel intensities were computed using a random number generator. Intensity values were scaled to interval $[0, 1]$. Background intensity follows a normal distribution $\mathcal{N}(0.8099, 0.0529)$ and foreground intensity follows a normal distribution $\mathcal{N}(0.5940, 0.0585)$. The size of computer generated image was $50 \times 50$ pixels. Cytological images of breast cancer were collected by pathologists from Regional Clinical Hospital in Zielona Góra, Poland. Biological material was acquired from affected breast tissue using fine needle biopsy under the control of an ultrasonograph. Some cases were malignant and some were benign (Fig. 1). Next, the material was fixed with Cellfix (Shadon) fixative spray and dyed with hematoxylin and eosin (h+e). Cytological preparations were then digitalized into virtual slides using the Olympus VS120 Virtual Microscopy System. For experimental studies, we used only small, selected fragments of these images. In order to distinguish nuclei from background, we build statistical models of intensity for both areas. It was assumed that intensity of nuclei and background follow normal distributions: $\mathcal{N}(\mu_N, \sigma_N^2)$ and $\mathcal{N}(\mu_B, \sigma_B^2)$ respectively. To estimate their parameters, we used training data taken form manually segmented cytological images. Estimated parameters are listed in the Table 1. Finally, we defined the interval for feasible radii of typical nucleus based on data from manually segmented nuclei. Maximal radius $r_{max}$, minimal radius $r_{min}$, average radius $r_{mean}$ of nuclei and standard deviation of radius $\sigma_r$ are listed in Table 2.

**Table 1.** Parameters of intensity models of nuclei and background

| Lesion | Nuclei | | Background | |
|---|---|---|---|---|
| | $\mu_N$ | $\sigma_N$ | $\mu_B$ | $\sigma_B$ |
| Benign | 0.594 | 0.059 | 0.810 | 0.053 |
| Malignant | 0.540 | 0.077 | 0.820 | 0.051 |

**Table 2.** Typical nuclei size [pix]

| Lesion | $r_{max}$ | $r_{min}$ | $r_{mean}$ | $\sigma_r$ |
|---|---|---|---|---|
| Benign | 26.88 | 22.00 | 24.89 | 01.27 |
| Malignant | 37.00 | 25.30 | 29.72 | 03.12 |

## 3 Methods

The main task of the proposed ICM based approach is to detect elliptical objects in the image. However, the method first estimates the location of such objects using circles. Finally, elliptical objects are extracted from detected circles.

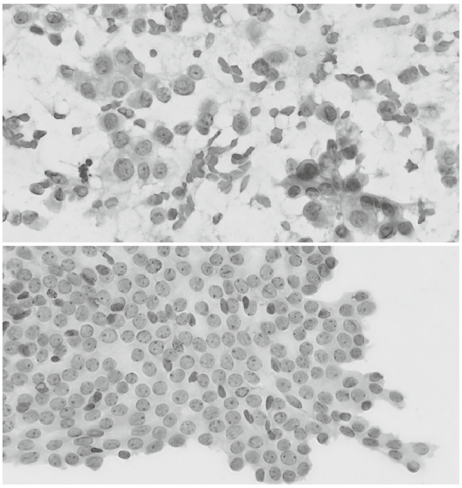


**Fig. 1.** Cytological images used in experiments: malignant case (left), benign case (right).

### 3.1 Circular Object Recognition

The input of the proposed procedure is intensity image y and $y_i$ denotes the pixel value at position $i$ on a finite pixel lattice $S$. Pixel values represent intensities $V = \{0, \ldots, 1\}$. Unknown configuration of circles is given by $\mathrm{x} = \{x_1, x_2, ..., x_n\}$. Each circle $x_i \in U$ is described by 3 parameters which are the coordinates of the center of the circle and its radius. Space $U$ of all possible parameters of circles is finite due to discrete finite image area $S$ and discrete set of feasible radii $r \in \{r_{min}, \ldots, r_{max}\}$ of circles. The objective is to determine the unknown circle configuration x given observation y. To do this, we assumed that the circle configuration x depends on the observed image y through a known probability distribution. So the crucial step of this method is to construct suitable conditional probability mass function $p(\mathrm{x}|\mathrm{y})$. Given conditional distribution and observation y, we must find such configuration $\hat{\mathrm{x}}$ that maximizes its probability. In Bayesian approach the process of finding unknown configuration $\hat{\mathrm{x}}$ may be viewed as maximum *a posteriori* estimation:

$$\hat{\mathrm{x}} = arg\, max_x l(\mathrm{y}|\mathrm{x}) p(\mathrm{x}). \tag{1}$$

To simplify the problem, it was assumed that *a prior* term $p(\mathrm{x})$ has uniform distribution. In this case, procedure boils down to maximum likelihood estimation:

$$\hat{\mathrm{x}} = arg\, max_x l(\mathrm{y}|\mathrm{x}). \tag{2}$$

Unfortunately, the solution space is huge because there is no limit put on the number of circles in the configuration. To find local maximum in the solution space, we applied ICM strategy [1,2,9,10]. It updates single state of $U$

by maximizing the conditional probability given the information about the state of all other points of $U$ and given data y. During sequential searching, likelihood never decreases at any stage and eventual convergence is guaranteed. However, algorithm will stuck in nearest local maxima. There are different variants of ICM method [1,9,10]. Here, we used Steepest Ascent and Coordinatewise Optimization algorithms [1,9,10]. The pseudocodes of these algorithms are presented below.

**Algorithm 1.** ICM using Coordinatewise Optimization

```
Initialization: $x^{(0)} = \emptyset, u = \{u_1, u_2, \ldots, u_n\}$ ;
for $i = 0, 1, 2, \ldots,$ $N$ do
    for $j = 1, \ldots, n$ do
        $k = j + in$;
        if $u_j \notin x^{(k-1)}$ and $\frac{l(y|x^{(k-1)} \cup u_j)}{l(y|x^{(k-1)})} > C$ then
            $x^{(k)} = x^{(k-1)} \cup u_j$;
        else if $u_j \in x^{(k-1)}$ and $\frac{l(y|x^{(k-1)} \setminus u_j)}{l(y|x^{(k-1)})} > C$ then
            $x^{(k)} = x^{(k-1)} \setminus u_j$;
        else
            $x^{(k)} = x^{(k-1)}$;
        end
    end
end
```

Coordinatewise optimization procedure starts with the empty configuration $x^{(0)}$. Then current configuration is modified by addition or deletion successive circles taken from set $u$. Threshold value $C$ is chosen arbitrary but its value does not significantly affect results. The procedure is repeated $N$ times.

**Algorithm 2.** ICM using Steepest Ascent

```
Initialization: $x^{(0)} = \emptyset$ ;
for $i = 0, 1, ...;$ do
    for $All$ $u$ do
        if $u \notin x^{(k-1)}$ then
            $w(u) = \frac{l(y|x^{(k-1)} \cup u)}{l(y|x^{(k-1)})}$;
        else if $u \in x^{(k-1)}$ then
            $w(u) = \frac{l(y|x^{(k-1)} \setminus u)}{l(y|x^{(k-1)})}$;
    end
    $find\ object\ u_k\ which\ maximizes\ w(u)$;
    if $u_k \notin x^{(k-1)}$ then
        $x^{(k)} = x^{(k-1)} \cup u_k$;
    else if $u_k \in x^{(k-1)}$ then
        $x^{(k)} = x^{(k-1)} \setminus u_k$;
end
```

Likelihood function $l(\mathrm{y}|\mathrm{x})$ evaluates how likely is to obtain the image y from given configuration x. Assuming that pixels $y_t$ are conditionally independent, it takes the following form:

$$l(\mathrm{y}|\mathrm{x}) = \prod_{y_t \in S(\mathrm{x})} g(y_t|\theta_N) \prod_{y_t \in S \setminus S(\mathrm{x})} g(y_t|\theta_B), \tag{3}$$

where $g(y_t|\theta_N)$ and $g(y_t|\theta_B)$ are probability mass functions which define the statistical models of nuclei and background intensities, $\theta_N$ and $\theta_B$ represent parameters of these models, $S(\mathrm{x})$ is a silhouette of configuration x:

$$S(\mathrm{x}) = \bigcup_{i=1}^{n} S(x_i), \tag{4}$$

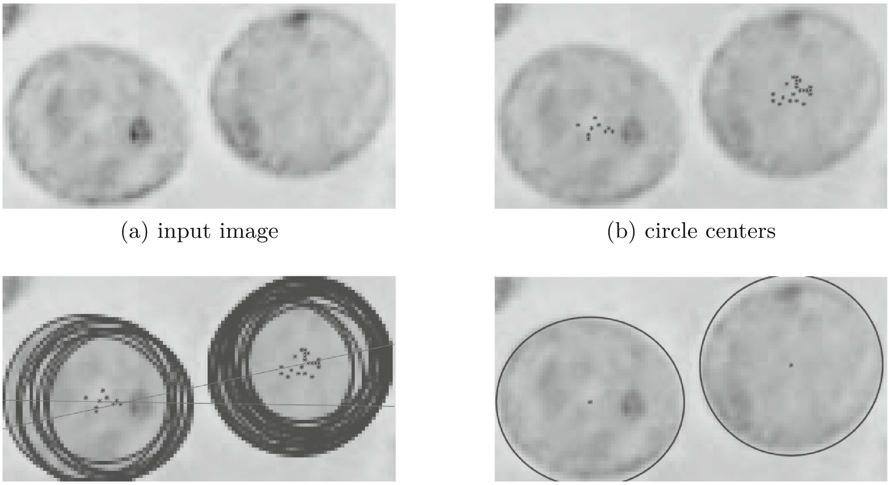
and $S(x_i)$ is the part of the image $S$ occupied by the disk generated by the circle $x_i$. Likelihood very often takes very small values, what causes numerical problems. Therefore, it is more convenient to use log-likelihood ratios instead of likelihood ratios while running Steepest Ascent or Coordinatewise Optimization:

$$\begin{aligned} \ln\left(\frac{l(x^{(k-1)} \cup u_j|y)}{l(x^{(k-1)}|y)}\right) \\ = \sum_{y_t \in S(\mathrm{x} \cup u_j)} \ln\big(g(y_t|\theta_F)\big) + \sum_{y_t \in S \setminus S(\mathrm{x} \cup u_j)} \ln\big(g(y_t|\theta_B)\big) \\ - \sum_{y_t \in S(\mathrm{x})} \ln\big(g(y_t|\theta_F)\big) - \sum_{y_t \in S \setminus S(\mathrm{x})} \ln\big(g(y_t|\theta_B)\big). \end{aligned} \tag{5}$$

It was assumed that both intensity models $g(y_t|\theta_N)$ and $g(y_t|\theta_B)$ follow normal distribution and their parameters are estimated from training data.

### 3.2 Ellipse Recognition

The maximum likelihood estimation has tendency to produce configurations of clustered and overlapping circles. This behavior can be explained by the fact that every circle configuration which covers properly nuclei gets high log-likelihood value. When given no constraints on disk placement, we usually obtain many redundant and overlapping circles. To deal with such problem, we propose to replace every cluster of closely spaced circles on single ellipse. It can be observed that circles covering single nucleus are approximately centered inside nucleus. Moreover, due to the fact that typical nucleus has elliptical shape, centers of such circles are approximately distributed along major axis of potential ellipse. A group of circles is considered to be a cluster if mutual distances between their centers are below a predetermined threshold. Threshold value is chosen based on the information about typical size of the nuclei (Table 2). With grouped circles, we can estimate major axis of ellipse for each cluster. The procedure is repeated for each cluster which includes at least two circles. First, the direction of major axis is determined by fitting line to circle centers. Line intersects with many circles belonging to the current cluster. The length of major axis is calculated on the basis of two most distant intersections between line and circles. A perpendicular line which represents minor axis, passes through the midpoint of major axis. The length of the minor axis is determined in the same way as major axis. Finally, the minor axis is shifted to get its midpoint exactly in the midpoint of major axis. Sample result of this procedure is shown in Fig. 2.


(a) input image

(b) circle centers

(c) line fitting

(d) final result

**Fig. 2.** Ellipse recognition procedure

## 4 Experimental Results

In order to verify the effectiveness of the proposed procedure, we applied it to detect circular objects and nuclei in test images. Below, we present some chosen results. Preliminary experiments were conducted using artificially generated test images. The first image contains 2 disks, both with radius $r = 11$ [pix]. Both variants of ICM, Coordinatwise Optimization and Steepest Ascent recognized the circular objects correctly (Fig. 3). The second test image contains 2 disks of different sizes. Smaller disk has radius $r = 10$ [pix] and greater disk has radius $r = 12$ [pix]. Coordinatwise Optimization returned a lot of redundant circles for greater disk and smaller disk was recognized correctly. Steepest Ascent recognized both disk correctly (Fig. 4). We observed, that generally the Steepest Ascent algorithm perform better than Coordinative Optimization. Especially in the case of circular objects which vary in the size. Satisfactory results of experiments carried out on synthetic images allowed us to proceed to the second stage of experiment. This time, the task boils down to detect nuclei in cytological images. However, problem is not as trivial as for artificially generated disks because nuclei have irregular shapes and are more like an ellipses than a circles. Due to high computational cost of ICM procedure, we was not able to process large images. Therefore, we selected for experiments only small fragments of cytological images. Below, we present 2 selected results obtained for benign (Fig. 5) and malignant (Fig. 6) case. Steepest Ascent version of ICM procedure was applied to recognize circles and then heuristic procedure presented in subsect. 3.2 was used to recognize ellipses. In Fig. 5, we can see the results of ellipse recognition for benign case. In the first step, Steepest Ascent procedure generated many overlapping circles. However, during the second step method detected ellipses which accurately estimates nuclei. One more experiment was carried out for malignant case. Results are presented in Fig. 6. As with the previous experiment, Steepest Ascent generated many overlapping circles but finally, method was able to recognize nuclei.

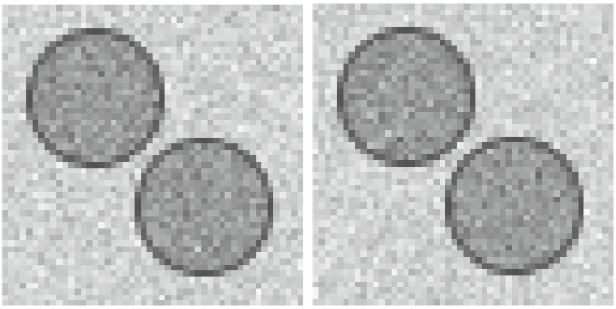


**Fig. 3.** Results for artificially generated image: Coordinatewise Optimization (left), Steepest Ascent (right).

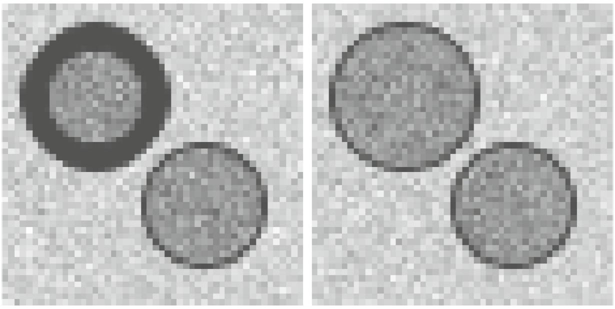


**Fig. 4.** Results for artificially generated image: Coordinatewise Optimization (left), Steepest Ascent (right).

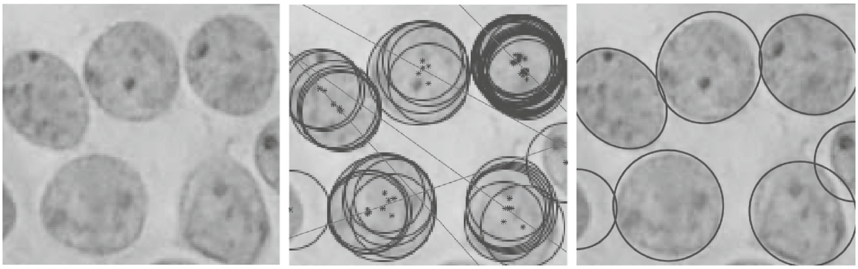


**Fig. 5.** Nuclei recognition - benign case: input image (left), circle recognition results (middle), ellipse recognition results (right)

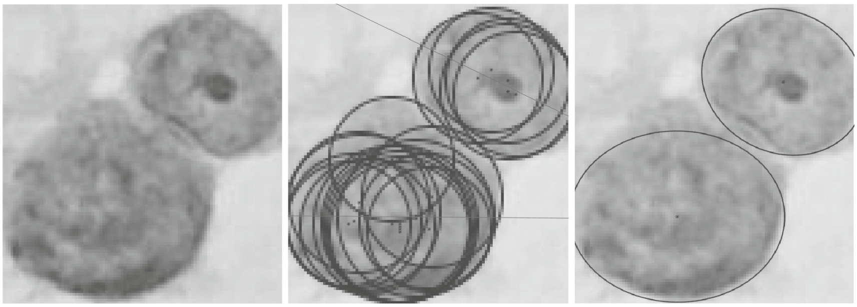


**Fig. 6.** Nuclei recognition - malignant case: input image (left), circle recognition results (middle), ellipse recognition results (right)

## 5 Conclusions

Content of cytological images is highly complex and its analysis is difficult in an automated way. The crucial step of this analysis is nuclei segmentation. Generally such methods of image processing as image thresholding, edge detection or active contours are not able to extract nuclei with satisfactory accuracy. This paper presents an alternative way of nuclei recognition which is based on ICM approach. Preliminary results are promising because method is able to detect nuclei even for noisy and low contrast scenes. The method requires the training data to build statistical models of objects to be detected. To avoid the tedious process of obtaining the training data, we propose that pathologist begins the process of image analysis by marking single nucleus and background area. Next, system is learning to build a preliminary model of nuclei and background. Based on these models, preliminary results of nuclei recognition are presented to pathologist. He can accept properly detected nuclei or rejects erroneous detections. The process is repeated until the correct result is presented to pathologist. In future work, we plan to adapt the algorithm to process large images and design *a priori* term $p(\mathrm{x})$, which allows us to penalize solutions with highly overlapping circles.

**Acknowledgments.** The research was supported by National Science Centre, Poland (2015/17/B/ST7/03704).

## References

1. Baddeley, A., Lieshout, M.N.M.: Stochastic geometry models in high-level vision. J. Appl. Stat. **20**(5–6), 231–256 (1993)
2. Besag, J.: On the statistical analysis of dirty pictures. J. Roy. Stat. Soc. Series (Methodol.) **48**(3), 259–302 (1986)
3. Chatelain, F., Descombes, X., Zerubia, J.: Parameter estimation for marked point processes. Application to object extraction from remote sensing images. In: Cremers, D., et al. (eds.) Lecture Notes in Computer Science, vol. 5681, pp. 221–234. Springer, Heidelberg (2009)
4. Descombes, X.: Stochastic Geometry for Image Analysis. Willey, London (2012)
5. Filipczuk, P., Fevens, T., Krzyżak, A., Monczak, R.: Computer-aided breast cancer diagnosis based on the analysis of cytological images of fine needle biopsies. IEEE Trans. Med. Imaging **32**(12), 2169–2178 (2013)
6. Kowal, M., Filipczuk, P.: Nuclei segmentation for computer-aided diagnosis of breast cancer. Int. J. Appl. Math. Comput. Sci. **24**(1), 19–31 (2014)
7. Kowal, M., Jacewicz, P., Korbicz, J.: Combaining image thresholding and fast marching for nuclei extraction in microscopic images. Advances in Intelligent Systems and Computing, vol. 525, pp. 195–202 (2016)
8. Krawczyk, B., Galar, M., Jeleń, L., Herrera, F.: Test article sample title placed here. Appl. Soft Comput. **38**, 714–726 (2016)
9. Lieshout, M.N.M.: A Bayesian approach to object recognition. In: Eckhardt, U., et al. (eds.) Geometrical Problems of Image Processing, vol. 4, pp. 185–190. Akademie, Berlin (1991)
10. Lieshout, M.N.M.: Markov point processes and their applications in high-level imaging. Bull. Int. Stat. Inst. **56**, 559–576 (1995)
11. Plissiti, M.E., Nikou, C.: Overlapping cell nuclei segmentation using a spatially adaptive active physical model. IEEE Tran. Image Process. **21**(11), 568–4580 (2012)
12. Perrin, G., Descombes, X., Zerubia, J.: 2D and 3D vegetation resource parameters assessment using marked point processes. In: Proceedings of the International Conference on Pattern Recognition (ICPR), Hong-Kong (2006)
13. Piórkowski, A., Mazurek, P., Gronkowska-Serafin, J.: Comparison of assessment regularity methods dedicated to isotropic cells structures analysis, In.: Choraś, R.S. (ed.) Image Processing & Communications Challenges 6. Advances in Intelligent Systems and Computing, pp. 169–178. Springer (2015)
14. Savchenko, A.V., Belova, N.S.: Statistical testing of segment homogeneity in classificatin of piecewise-regular objects. Int. J. Appl. Math. Comput. Sci. **25**(4), 915–925 (2015)
15. Trajdos, P., Kurzynski, M.: A dynamic model of classifier competence based on the local fuzzy confusion matrix and the random reference classifier. Int. J. Appl. Math. Comput. Sci. **26**(1), 175–189 (2016)

# Feature Selection in Texts

Magdalena Wiercioch(✉)

Faculty of Mathematics and Computer Science,
Lojasiewicza 6, 30-348 Kraków, Poland
magdalena.wiercioch@ii.uj.edu.pl

**Abstract.** Feature selection is used in many application areas relevant to expert and intelligent systems, such as machine learning, data mining, cheminformatics and natural language processing. In this study we propose methods for feature selection and features analysis based on Support Vector Machines (SVM) with linear kernels. We explore how these techniques can be used to obtain some interesting information for further exploration of text data. The results provide satisfactory observations which may lead to progress in feature selection field.

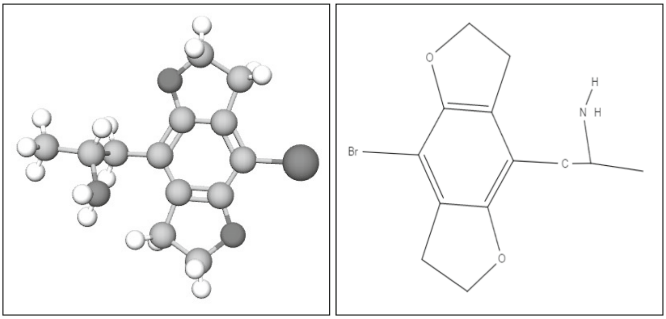
**Keywords:** Feature selection · Text classification · Dimension reduction · Support Vector Machines

## 1 Introduction

High dimensional data is a significant problem in both supervised and unsupervised learning [7]. It is becoming even more prominent with the recent explosion of the size of the available datasets both in terms of the number of features in each sample and the number of data samples [17]. The main motivation for reducing the dimensionality of the data and keeping the number of features as low as possible is to decrease the training time and enhance the classification accuracy of the algorithms [3,11].

Take for instance, cheminformatics. Chemical compounds are usually represented by fingerprints, i.e. high dimensional binary strings where a given bit indicates the absence or presence of particular feature of compound [8]. Since a lot of features can be taken into account and it increases the computational costs, another models are considered as well. Chemical databases often store data in form of SMILES. A SMILE (Simplified Molecular Input Line Entry System) is a string of ASCII characters associated with atoms and bonds which build the molecule (see Fig. 1). It is usually visually presented as a graph. Feature selection has received considerable attention in the machine learning and data mining communities. It is typically performed by sorting linguistic features according to some weighting measure [13,16] and then setting up a some kind of threshold on the weights or specifying a percentage or number of highly scored features to be retained. Features with lower weights are omitted as having less significance

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_35



(a) A molecule in 3D space.

(b) A graph-like representation of the molecule.

**Fig. 1.** The SMILES for this molecule is CC(N)Cc1c2CCOc2c(Br)c2CCOc12.

for the classification decision. Furthermore, [14] proposes the method that uses frequent graphs. In addition, much of the past research in cheminformatics has focused on substructure based approaches [10,15]. *The main contribution of this paper is selecting the most valuable fragments of molecules.* However, we also test our approaches on text documents with text stories. In this study we propose techniques for feature selection and analysis based on Support Vector Machines (SVM) [4] with linear kernels. Our experimental results on real world chemical datasets and a collection of Reuters documents [9] show that the demonstrated approaches are promising tools. As it shows, the methods will also be useful for further investigations in the fields connected with representation construction. In the following sections we first describe the basic concepts used in further study. Then, we give a brief explanation of our feature selection methods. Section 3 introduces the experiments and presents the results. We conclude with a brief summary and the outline of future research.

## 2 Methodology

Our hypothesis in feature selection is that the spatial distribution of text features carries important information regarding the importance of the feature. In presented approaches we consider text data, i.e. SMILES and stories. Figure 2 gives an overview on our approach. In step 1, the text is preprocessed. The type of preprocessing depends on data.

- As previously mentioned, according to SMILES notation, the molecule is represented as a graph. Thus, for each data-molecule random SMILES walks of given length $l$ are performed. In consequence, we obtain many fragments of the graph. Additionally, each fragment has the same label as the graph it belongs to (see Subsect. 3.1). Figure 3 shows two fragments extracted from the original chemical compound. What is more, the preprocessing stage includes a simple tokenization. We assume that SMILES are considered as 2 character

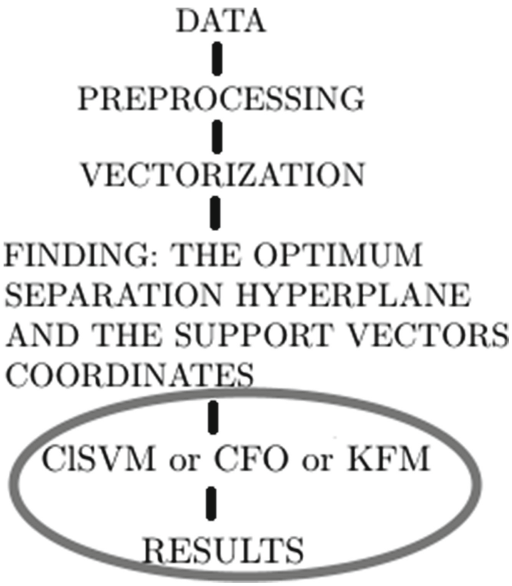

DATA
PREPROCESSING
VECTORIZATION
FINDING: THE OPTIMUM SEPARATION HYPERPLANE AND THE SUPPORT VECTORS COORDINATES
ClSVM or CFO or KFM
RESULTS

**Fig. 2.** The procedure workflow.

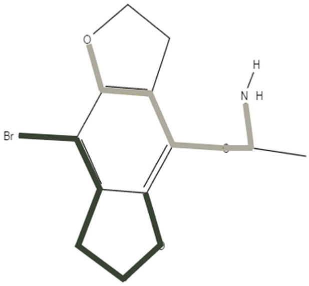


**Fig. 3.** Visualization of two paths of length 6.

strings. Nevertheless, the chemical properties enforce some exceptions. For instance, [Br-] is treated as a single token.
- As far as texts are considered, they sometimes consist of useful data which has to be removed. These are stopwords or special characters (#, %).

In order to perform some machine learning operations on texts, the conversion of text data into numerical data is required (see the vectorization phase in Fig. 2). Next the LibSVM package [2] is used to train a Support Vector Machine (SVM) [4] for classification. We use the Support Vector Machine with linear kernels. After the vectorization [5], training examples are described by vectors $x_i = (x_i^1, \ldots, x_i^d)$, where $d$ represents the dimensionality of the feature space. The subsequent step 2 is 1 out of 3 different approaches: Clustered SVM (ClSVM), Counting Features Occurrences (CFO) and Key Features-to-Model (KFM).

**Clustered SVM (ClSVM).** The classification step provides a set of points, namely support vectors. The randomly selected support vectors are initial cluster centers for $k$-mean clustering [12]. The goal of such an approach is to gather some features into clusters. It enables to verify the kind of features (in sense of their chemical properties) included together. The whole workflow of ClSVM with the previous steps for chemical data is summarized schematically in Fig. 4.

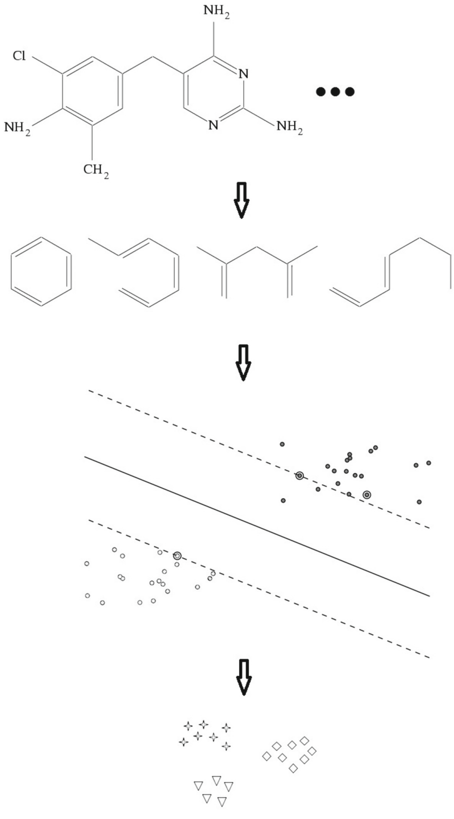


**Fig. 4.** The procedure workflow with ClSVM.

**Key Features-to-Model (KFM).** We assume the support vectors provide the most valuable features. Thus, these features are used to create a new representation. To be more precise, for each class we select $n$ most common points from the set of support vectors and concatenate them (see Fig. 5).

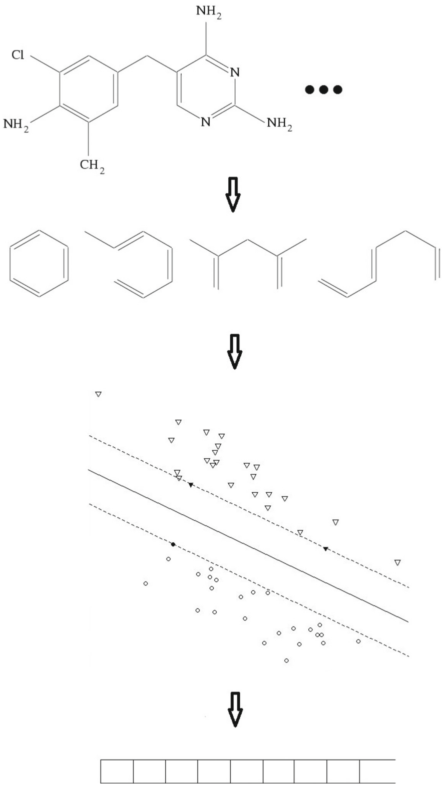


**Fig. 5.** The procedure workflow with KFM.

**Counting Features Occurrences (CFO).** Given a linear hyperplane $H$ which divides a dataset $X$ into two regions $X_{-1}$ and $X_{+1}$ we perform a classification of a new point - $x_{new}$ (a new word or a molecule fragment) based on the following system:

- The number of occurrences of the new element within each separate class is calculated: $num_{-1}$, $num_{+1}$.
- Finally, the point $x_{new}$ is assigned to the class where it occurs more often.

Figure 6 illustrates this approach.

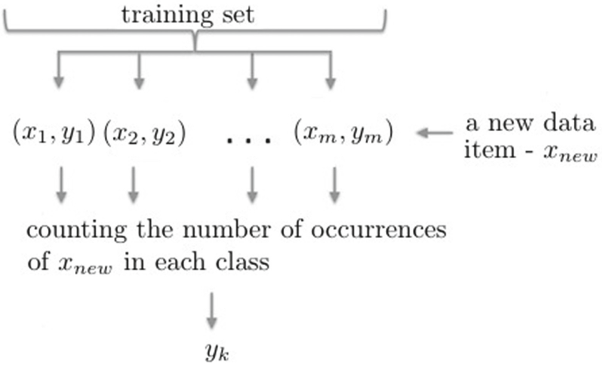

training set
$(x_1, y_1)$ $(x_2, y_2)$ . . . $(x_m, y_m)$
a new data item - $x_{new}$
counting the number of occurrences of $x_{new}$ in each class
$y_k$

**Fig. 6.** The procedure workflow with CFO.

# 3 Experimental Study

In this section we introduce the datasets used in the experiments. Then the presentation of the results is given.

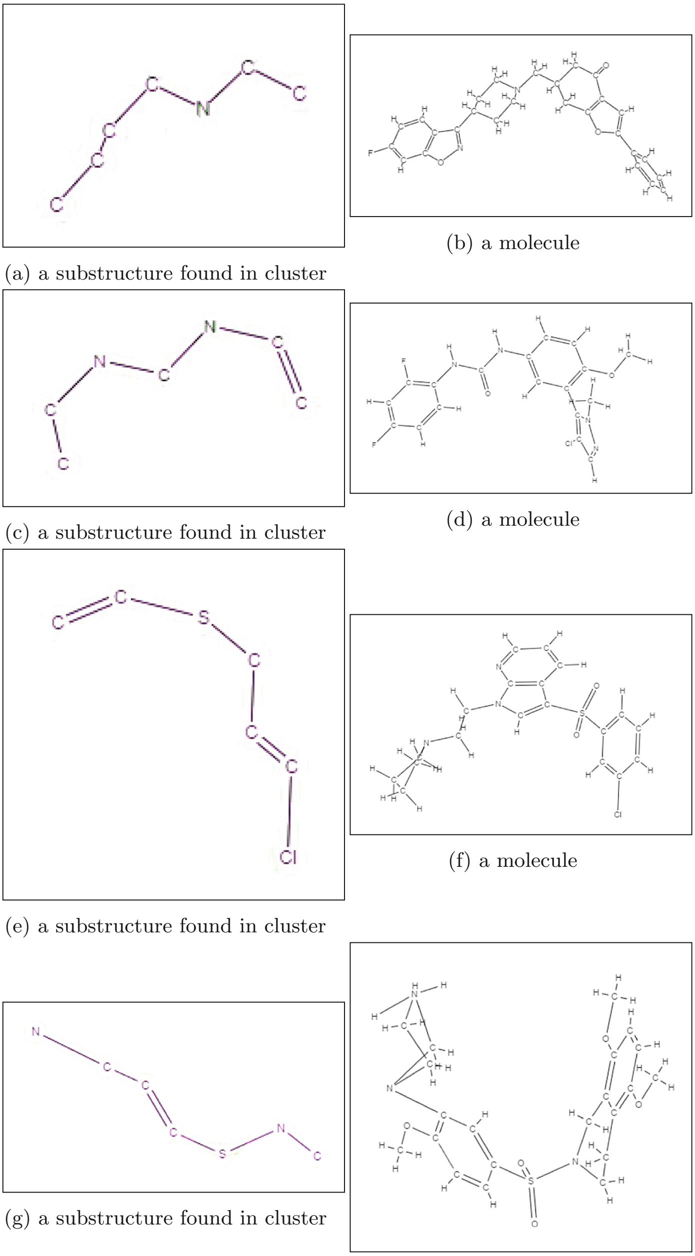
**Table 1.** Overview of considered data sets. Table contains the names and the number of active and inactive compounds included in initial dataset.

| Receptor name | Actives | Inactives |
|---|---|---|
| M1 | 759 | 938 |
| H1 | 635 | 545 |
| $5HT_7$ | 704 | 339 |
| $5HT_{2A}$ | 1835 | 851 |
| $5HT_6$ | 1490 | 341 |

## 3.1 Data Sets

For experiments we used 2 different types of datasets, i.e. SMILES representation of chemical compounds and text data. As molecules are considered, five biological receptor ligands were used [6], each represents a one receptor ligands, Table 1. Note that there exists an inhibition constant $K_i$ as a kind of activity threshold. For a given molecule, if this factor is less or equal 100nM, the compound is treated as active. However, if $K_i$ is higher than 1000nM, the molecule is seen as inactive. Since the majority of molecules in the real world are inactive, an imbalance dataset problem appears. To tackle it, in our experiments we randomly select inactive molecules as many as actives.

The second collection, namely the Reuters-2000 collection [9] includes a total of 806,791 documents, with news stories covering the period from 20 Aug 1996



(a) a substructure found in cluster

(b) a molecule

(c) a substructure found in cluster

(d) a molecule

(e) a substructure found in cluster

(f) a molecule

(g) a substructure found in cluster

(h) a molecule

**Fig. 7.** The examples of substructures found in clusters.

through 19 Aug 1997. We selected documents from these corpus and divided this data into a training and test sets. The finally obtained corpora includes 2 groups of documents on different subjects, i.e. Earn and Acquisition. We modeled a textual document as a kind of a graph of words which corresponds to a graph. Each vertices are associated with unique terms of the document and edges show cooccurrences between the terms within a fixed-size sliding window. The background of this concept is presented in [1].

### 3.2 Experimental Results

For the tests where chemical data are studied, the obtained graph fragments are the effects of walks of length $l = 6$. The task of conversion of text data into numerical form, namely vectorization, is carried out using CountVectorizer [5]. Moreover, C and other parameters of SVM were tuned using cross validation.

**Clustered SVM (ClSVM).** The first experiment investigates the results of ClSVM procedure. Indeed, our observations show the obtained clusters include fragments which enable to partially recreate the original molecule. Figure 7 demonstrates only the examples of such fragments found in groups. In is quite interesting since the empirical studies suggest that the fragments included in one cluster enable to reproduce the real compound up to 60% of its real form. Although it is difficult to try to recreate the stories since some data was removed at preprocessing stage, in fact the single clusters contain a set of words associated with the given documents.

**Counting Features Occurrences (CFO).** Table 2 demonstrates the comparison of CFO classifier with SVM on test data set. The classification quality is calculated with F-measure. As one can see, our approach has performed slightly better in 2 out of 5 cases. It suggests that similar methods are worth to be further explored. For Reuters data we have achieved F-measure of 0.62 which is less than the F-measure of SVM (0.85). This may be caused by type of data.

**Key Features-to-Model (KFM).** Finally, for KFM we tested the final representation of length 1000. Table 3 shows text classification accuracy for chemical

**Table 2.** Classification results measured with F-score for CFO and SVM.

| Receptor name | CFO | SVM |
|---|---|---|
| M1 | **0.67** | 0.65 |
| H1 | 0.68 | **0.72** |
| $5HT_7$ | 0.65 | **0.69** |
| $5HT_{2A}$ | 0.73 | **0.8** |
| $5HT_6$ | **0.73** | 0.72 |

**Table 3.** Molecules classification accuracy using SVM. The compound are represented with KFM model and typically, as SMILES.

| Receptor name | KFM | SVM |
|---|---|---|
| M1 | 0.5 | **0.6** |
| H1 | 0.6 | **0.7** |
| $5HT_7$ | 0.55 | **0.69** |
| $5HT_{2A}$ | 0.7 | **0.8** |
| $5HT_6$ | 0.68 | **0.72** |

data using SVM with and without applying the new representation. In fact our model has not achieved better results. However, it still provides a valuable insight into features properties. It should be noticed, the outcomes are not random, so the technique is worth to extend. The experiments outcomes raise an interesting question why it may work. The first reason seems to be connected with the fragments properties - the more fragments, the information is more valuable. What is more, document vectors are sparse and SVM-based approaches are well suited for such problems. Despite the fact our preliminary experiments show encouraging results, this research leaves open a few issues.

- On the exemplary data the methods perform quite satisfactory. However, the more reliable comparison with other similar approaches is absolutely necessary. At this stage, we have not found any analogous techniques.
- Most of the results which are not described in this work are somewhat empirical, i.e. they are based on visual inspection or analysis. Thus, the further exploration should be done.

## 4 Conclusion

In this paper we have presented three techniques for feature selection and features analysis. The proposed methods seemingly have an influence on text data exploration. In the future, we plan to extend the idea of support vectors-based clusters and analyze another points as potential centroids.

**Acknowledgments.** This research was partially supported by National Centre of Science (Poland) Grants No. 2016/21/N/ST6/01019.

## References

1. Blanco, R., Lioma, C.: Graph-based term weighting for information retrieval. Inf. Retr. **15**(1), 54–92 (2012). http://dx.doi.org/10.1007/s10791-011-9172-x
2. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**, 27:1–27:27 (2011). http://www.csie.ntu.edu.tw/~cjlin/libsvm

3. Chen, R. (ed.): ICICIS 2011, Part II. CCIS, vol. 135. Springer, Heidelberg (2011)
4. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
5. Garreta, R., Moncecchi, G.: Learning Scikit-learn: Machine Learning in Python. Packt Publishing (2013)
6. Gaulton, A., Bellis, L.J., Bento, A.P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., Overington, J.P.: Chembl: a large-scale bioactivity database for drug discovery. Nucleic Acids Res. **40**(D1), D1100 (2011). http://dx.doi.org/10.1093/nar/gkr777
7. Janecek, A., Gansterer, W.N., Demel, M., Ecker, G.: On the relationship between feature selection and classification accuracy. FSDM **4**, 90–105 (2008)
8. Klekota, J., Roth, F.P.: Chemical substructures that enrich for biological activity. Bioinformatics **24**(21), 2518–2525 (2008)
9. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: Rcv1: a new benchmark collection for text categorization research. J. Mach. Learn. Res. **5**, 361–397 (2004)
10. Kramer, S., De Raedt, L., Helma, C.: Molecular feature mining in HIV data. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 136–143 (2001)
11. Lim, T.S., Loh, W.Y., Shih, Y.S.: A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Mach. Learn. **40**(3), 203–228 (2000)
12. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, Oakland, CA, USA, pp. 281–297 (1967)
13. Mladenic, D., Grobelnik, M.: Feature selection for unbalanced class distribution and naive Bayes. In: Proceedings of the 16th International Conference on Machine Learning (ICML), pp. 258–267. Morgan Kaufmann Publishers (1999)
14. Thoma, M., Cheng, H., Gretton, A., Han, J., Kriegel, H.P., Smola, A., Song, L., Yu, P., Yan, X., Borgwardt, K.: Near-optimal supervised feature selection among frequent subgraphs, pp. 1076–1087. Max-Planck-Gesellschaft/Society for Industrial and Applied Mathematics, Philadelphia, May 2009
15. Wale, N., Watson, I.A., Karypis, G.: Comparison of descriptor spaces for chemical compound retrieval and classification. Knowl. Inf. Syst. **14**(3), 347–375 (2008)
16. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: ICML 1997, pp. 412–420 (1997)
17. Zhang, Y., Yang, C., Yang, A., Xiong, C., Zhou, X., Zhang, Z.: Feature selection for classification with class-separability strategy and data envelopment analysis. Neurocomputing 166, 172–184 (2015), http://www.sciencedirect.com/science/article/pii/S0925231215004609

# Image Processing and Analysis in Lung Cancer Cells Growth

Przemysław Jędrusik[1(✉)], Łukasz Walusiak[1,2], Ilona Bednarek[3], Robert Koprowski[1], and Zygmunt Wróbel[1]

[1] Department of Computer Biomedical Systems, Institute of Computer Science,
University of Silesia, ul. Będzińska 39, 41-200 Sosnowiec, Poland
pjedrusik@o2.pl, {lwalusiak,robert.koprowski}@us.edu.pl

[2] Institute of Technology, Pedagogical University,
ul. Podchorążych 2, 30-084 Cracow, Poland

[3] Department of Biotechnology and Genetic Engineering,
School of Pharmacy with the Division of Laboratory Medicine in Sosnowiec,
Medical University of Silesia, ul. Jedności 8, 41-200 Sosnowiec, Poland
ibednarek@sum.edu.pl

**Abstract.** Modern diagnostic methods allow to get multiple information regarding research material. Patients diagnosis is performed using highly specialized tools, effectively supporting any medical diagnostic processes. This paper focuses on the analysis of lung cancer cell cultures growth and migration in vitro. Most of the publications on the growth rate of cells is based on the analysis of changes in surface area, less wide cracks. This study determined there are additional parameters like cells angle, number of cells and distance between cells, with separate cells up and down the scratch in all parameters, that affect how the migration of cells which have not been considered previously. Analysis on the arrangement of the cells and the distances between them, allow for determination of the level of cell migration. Experience has shown that on the first day a high proliferation of cells, and then clear their migration, increasing the distance. It was also noted changing the angle of the cells that begin migration. The performed analysis confirmed that those additional parameters differentiate correctly evaluated a group of images. Developed algorithm of image processing and analysis, operates on data from a collection of microscopic images of lung cancer cells in vitro propagation, acquiring data on the growth and migration of cancer cells. As a result the data contain a description of parameters studied images in the form of growth profiles over time and the type of growth.

**Keywords:** Image processing · Algorithms · Lung cancer · Cell culture · Cell migration · Wound healing

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_36

## 1 Introduction

Modern diagnostic systems computer-based tools are becoming more widely used in medicine. The process of medical analysis in multiple stages is closely linked with the use of specialized computer algorithms. The whole group studies such as medical ultrasonography, X-ray, computed tomography scan [11], magnetic resonance imaging, echocardiography, provides information about the patient in the form of medical images. Cell migration has a essential role in development and in physiological repair processes, and is also a component in many pathological disorders, including cancer invasion and metastasis, angiogenesis, and inflammatory reactions [5]. The use of support as an image analysis, effectively improves the accuracy and speed measurements. This eliminates the typical measurement errors resulting from fatigue or inattention. A lung cancer develops in most cases without any visible symptoms. This is the main problem with the early detection and diagnosis of lung cancer. Therefore, it is important to understand how the growth of lung cancer cells, as well as to propose appropriate drug therapy [12]. The lung cancer is one of the most important medical problems of today's world [10]. Worldwide, diagnosed with lung cancer is approximately 13% of all cancers diagnosed [4]. This represents approximately 1.6 million cases [4]. They are the most commonly diagnosed cancer among men and the fourth to the number of women. Any types of lung cancer kill more than 1.4 million (18%) men in the world [4]. In Europe in 2012 were just over 3.4 million new cases of cancer, lung cancer was a 410000 [7]. In men, incidence was highest in Central and Eastern European countries. In contrast, women observed increased incidence in Northern European countries and low rates in Eastern Europe. In 2010 in Poland, the number of cases of lung cancer is approximately 21000, of which about 15000 are men and 6000 women [4]. According to the World Health Organization, globally the world, lung cancer is the leading cause of cancer death in men and the second leading cause of cancer death in women, with an estimated 951000 deaths in men and 427400 deaths in women in 2008 [2]. Most of the articles relating to cell migration focuses on data analysis of scratch changes over time, placed in the middle of the culture plate. Parameters obtained by using a combination of analysis and image processing methods usually includes a surface area or width of the scratches [16,17]. The analysis of these parameters can determine whether there were changes in the culture of individual measuring points or allowed to specify the rate of change, growth acceleration. The rate of changes was also based on the changes of surface area scratches or cell area. Some studies also determine the number of cells in the different measuring points [15]. The research mainly relate to the entire surface of the image, without distinguishing between the areas on either side of previously created scratches.

## 2 Materials in Experiment

The study of this work is based on microscopic analysis of medical images A549 lung cancer cells ATCC CCL-185TM. Lung cancer cell line originated from the group ATCC, grown in accordance with the manufacturer's instructions. Cell cultures were routinely propagated in a humidified incubator Hera-Cell (Heraeus), at 37° C, in 5% CO2 atmosphere. Cells were grown in RPMI-1640 medium supplemented with 10% FBS and gentamicin (20[$\mu$g/ml]). Cells were seeded in 12-well plates to reach confluency. Then, in each well scratch line was made by applying a sterile pipette tip across the center of the well to produce a 1 mm-wide wound area. Unattached cells were removed by washing with PBS. Cells were allowed to growth and migrate for 24, 48 and 72 h. Kinetics of migration within the scratch lines were visualized under inverted microscope (OLYMPUS BX-60 microscope with DP50 digital camera) at indicated time points; magnification 100x. The acquisition of microscopic images was performed in the Department of Biotechnology and Genetic Engineering Medical University of Silesia in Katowice. Were registered 48 different samples containing digital images in 4 measurement time points. All obtained images is provided in each of the same area of the cell culture. The images in digital form are saved as a bitmap with a resolution of 2080 × 1540 pixels - 150 dpi. Pixel of acquired images was dimension of approximately 10 nm in real. Images of all groups were analyzed using a proprietary algorithm, written in Matlab version R2013b software with Image Processing Toolbox version 8.3. The software was running on a computer with an Intel Core i5 third generation (Ivy Bridge) processor clocked at 2.6 GHz, along with 8 GB of DDR3 RAM, serial ATA hard disk drive and operating system Microsoft Windows 10 Pro (version 1511, compilation 10586.63).

## 3 Methods

In the initial step of the designed algorithm was applied pre-processing method for all input images. Due to the presence of artifacts which affect for further analysis and calculation, the initial phase of the algorithm is responsible for its elimination [1,3]. An example of input images can be found in Fig. 1. Prepared in this way images of lung cancer cells treated with the different steps of the algorithm, which all steps are shown in Fig. 2. For each of the images were used the median filter with local window size 19 × 19 pixels [9]. The median filter mask size has been selected based to remove unnecessary elements (artifacts) from an image in the initial phase of the algorithm. Because the image parameters refer to both scratches as well as individual cells, the process was carried out independently for finding scratches and cells. In the case of scratches was used median filter with a respectively large window. The size of the windows was good in a way corresponding to the largest width of the scratch. Then it was appointed the lower threshold binarization based on the histogram [1]. On the other hand, to determining the cells was used adaptive binarization with local window. Window size was selected based on analysis of changes in the brightness of the image pixel groups. The use of such an operation eliminates uneven lighting image and the correct determination of individual cells. The sample images after performing binarization was shown in Fig. 3. The next step of the algorithm was to use the morphological operations: erosion and dilation,

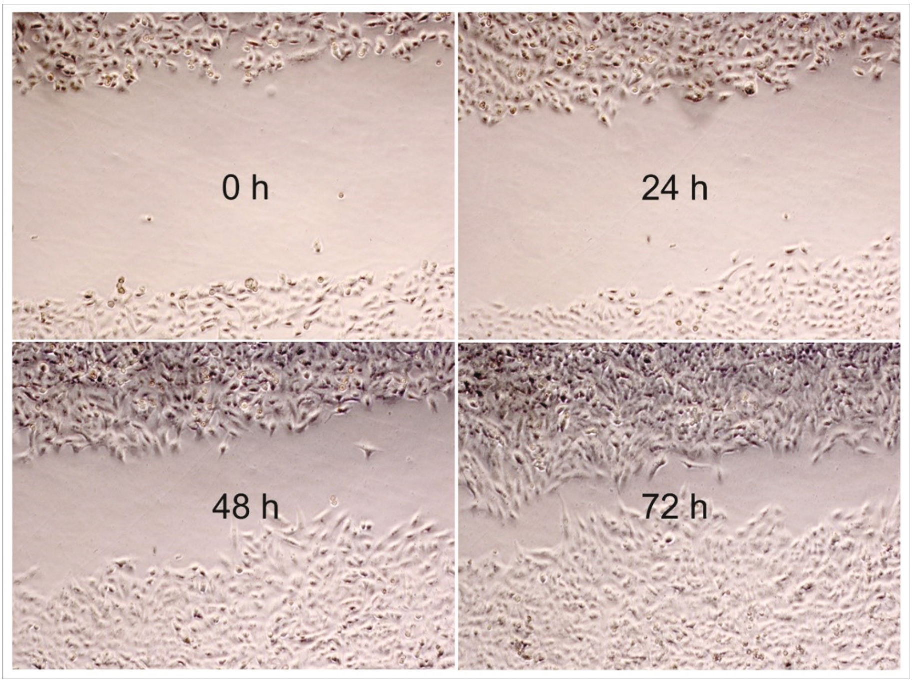


**Fig. 1.** Example source images of lung cancer.

as well as of their respective assemblies. Submission of these operations were operations morphological opening and closure to allow for the elimination of undesirable elements in the image. As in the case of determining the size of the median filter mask, the structuring element size was chosen to remove from the image elements considered artifacts, leaving only the essential elements of the image. Before the measurements started, has been made the process of edge detection, separately for scratch and cancer cells. Figure 4 shows the contours of the scratch and Fig. 5 shows cancer cells contours. In addition, has been marked curve which is the middle of scratch. This curve is the boundary between the cells occurring above and below scratch, especially important for images in the last stages of cells growth. The standard has been set image parameters such as surface area of the cells and width of scratches [13]. Additionally they have been appointed the new parameters which define how migration cells growth in subsequent phases, where scratch was not visible in full width of image. The first additional parameter was to determine the number of cells present above and below the scratch. The next parameter was to measure distance between the cells according to the position above and below the scratch, appointed based on the distance matrix of the center of gravity cells. On the images, where scratches was not fully visible, they were taken into account only those cells that were over or under fixed in advance on the curve (the middle scratches). Another

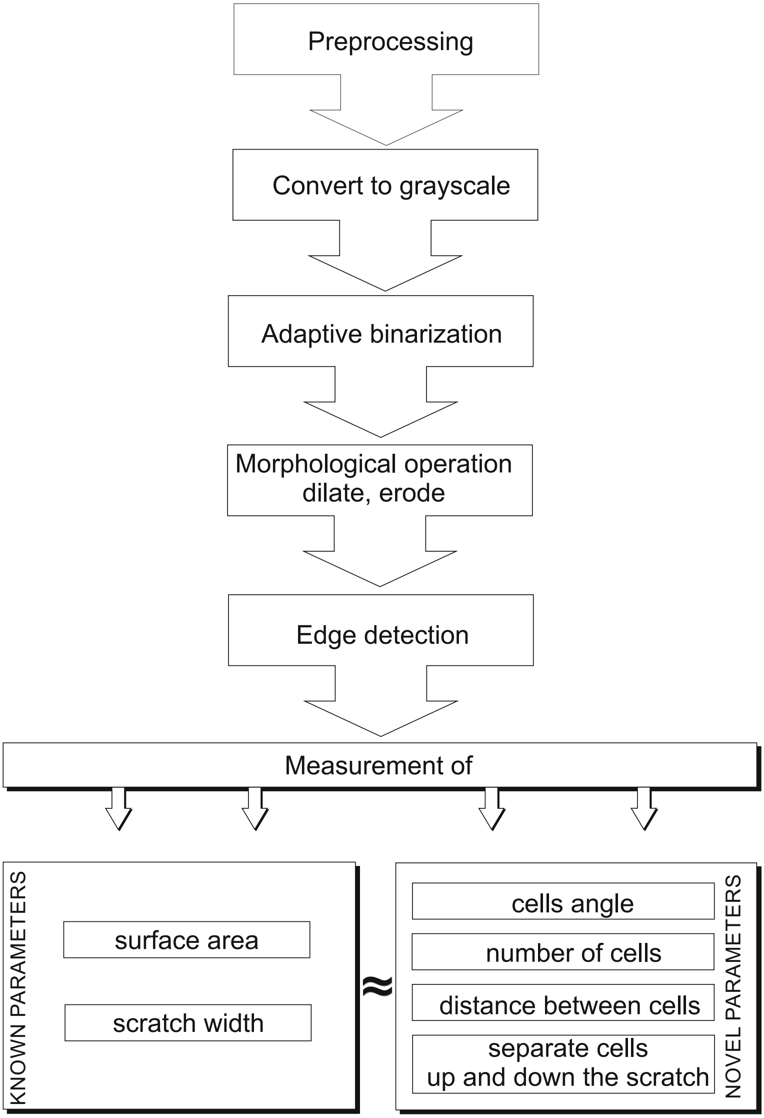


**Fig. 2.** Schematic of the proposed algorithm for image processing and analysis rate of changes in lung cancer cells.

important parameter affecting the growth of cancer cells was the alignment of the tumor cells in the immediate vicinity of the crack. The measurement on the arrangement of objects under study was carried out by analyzing the position of a diagonal set of cells in the horizontal axis of the image. Let them be microscopic image data objects $C_1, C_2, \ldots, C_n$, directly adjacent to the designated contour features. Next let respectively $d_1, d_2, \ldots, d_n$ are the longest diagonal (chords) of objects $C_1, C_2, \ldots, C_n$. Then it was measured the average angle formed by the diagonals $d_1, d_2, \ldots, d_n$ horizontal axis of the image. The idea of measuring presents a Fig. 6a. Analysis of the average angle includes cells located under and above to the scratch, similar to the previous calculations. Figure 6b shows the marked diagonal cells that are in close the neighborhood scratch in a sample part of image. Finally, each of the 42 cases containing images in 4 measuring points (0 h, 24 h, 48 h, 72 h), has been evaluated by an expert. In this phase of the experiment they were separated the two groups, which are characterized by the evaluation cell growth rate. The first group was characterized by moderate growth and low chaotic, and the second group by fast growth and high chaotic.

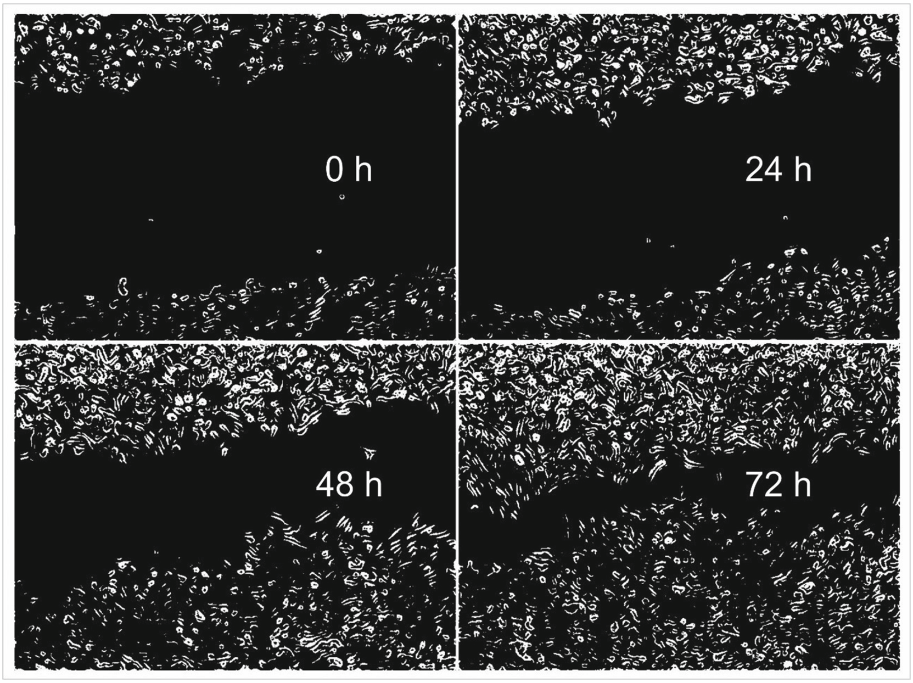


**Fig. 3.** Example images after binarization.

## 4 Results

Obtained in this study the results of analyzes concerning the planned measurements included 42 cases of measurements. Results of a part of the cases of measurement are shown in Table 1. Analysis of the results has shown that selecting additional parameters allows correct differentiation of specific groups of cases. Calculated changes in the number of cells divided into the areas above and below the scratch, allowed for determine the exact rate of growth. In addition, taking

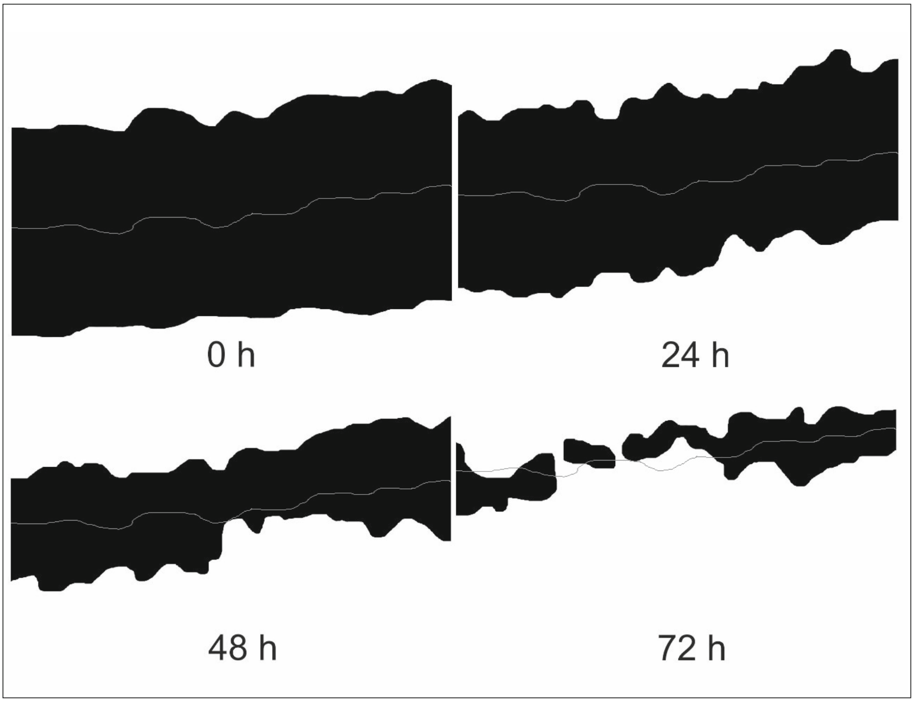


**Fig. 4.** Contours of the scratchs with marked curve.

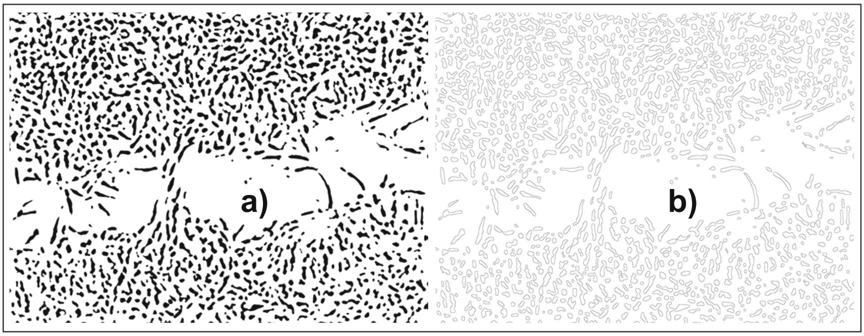


**Fig. 5.** The results of the subsequent steps of the algorithm: (a) extracted cells, (b) extracted cells contours.

into account another parameter, which is the average distance between cells, can be show what is the appearance of cell migration and growth. And finally analyzing the degree of stacking of cells relative to the horizontal axis of image, can be determined areas, where the cells will migrate faster, and in which more slowly. The results obtained show how these parameters differentiate each of

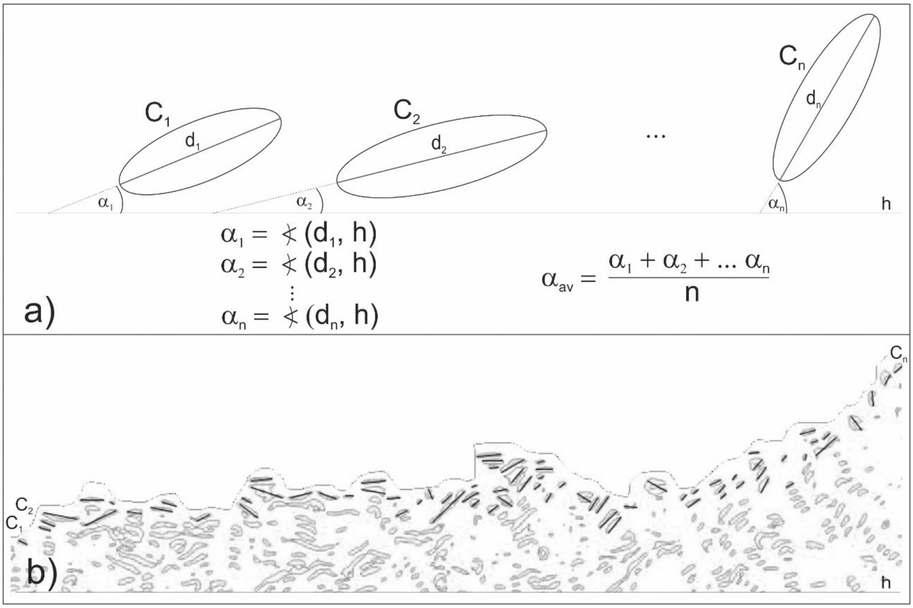


**Fig. 6.** Idea of set average angle of object diagonal and horizontal axis of the image: (a) description of the calculation, (b) apply the calculate to sample microscopic image.

**Table 1.** Part of the results of measurements: cells surface area, scratch average width, average angle of cells (horizontal rule of image), av. distance cells, number of cells.

| case | cells surface area [%] | | | | mean scratch width [pixels] | | | | mean angle cells near scratch [°] | | | | mean cells distance [pixels] | | | | number of cells | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| no. | 0h | 24h | 48h | 72h | 0h | 24h | 48h | 72h | 0h | 24h | 48h | 72h | 0h | 24h | 48h | 72h | 0h | 24h | 48h | 72h |
| 1 | 36.47 | 37.07 | 36.51 | 36.50 | 759 | 771 | 759 | 759 | 27 | 29 | 32 | 35 | 45 | 52 | 55 | 58 | 425 | 395 | 395 | 358 |
| 2 | 42.11 | 54.26 | 78.30 | 92.06 | 1204 | 951 | 452 | 165 | 33 | 38 | 56 | 76 | 34 | 39 | 58 | 75 | 443 | 611 | 808 | 1129 |
| 3 | 35.53 | 47.12 | 70.02 | 88.45 | 1341 | 1100 | 624 | 240 | 24 | 29 | 68 | 82 | 52 | 73 | 81 | 96 | 426 | 519 | 728 | 1020 |
| 4 | 42.42 | 50.71 | 76.46 | 88.87 | 1198 | 1025 | 490 | 232 | 32 | 39 | 48 | 69 | 37 | 49 | 77 | 88 | 455 | 720 | 953 | 1271 |

cases. Differentiating between groups was shown in Fig. 7, including the ratio number of cells above and below the scratch ($\frac{L_{above}}{L_{below}}$), ratio the mean distance between the cells and the average angle formed by the cell edge with a horizontal axis of the image.

## 5 Discussion

The results define the way the migration and growth of tumor cells. For cases assessed by experts designated parameters have been correctly differentiating each group growth. Average number of cells broken down into sub-positions relative to a scratch, show strong variation according to the expert's assessment. Changing the number of cells does not proceed in a linear fashion, but on the

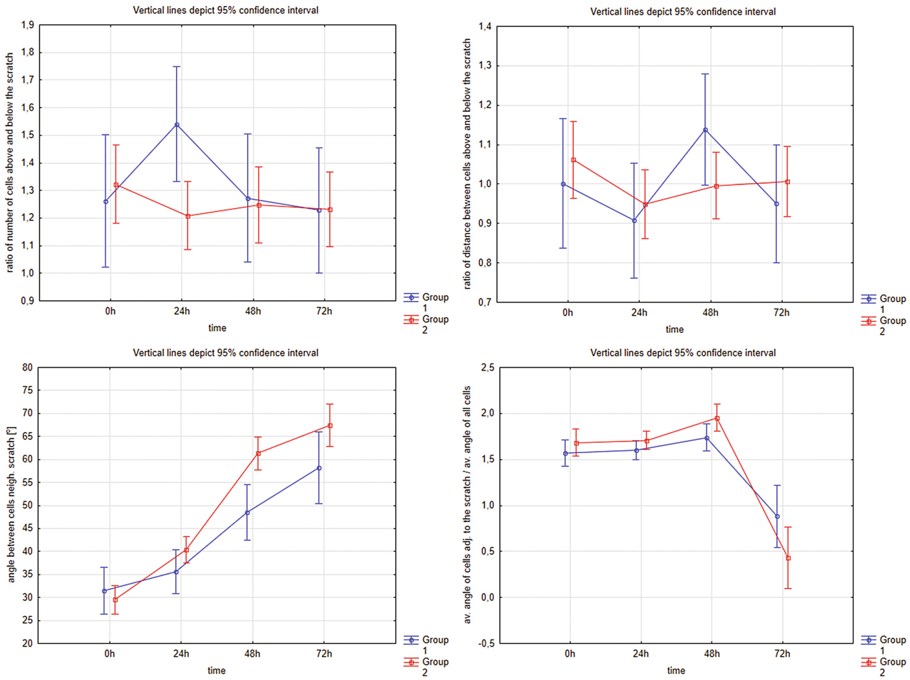


**Fig. 7.** Diversity groups: (a) in relation to the quotient of the number of cells, (b) in relation to the quotient terms of the average distance between cells, (c) in relative to the average angle of the neighboring cells to the scratch, (d) in terms the quotient of the average angle of the neighboring cell to the scratch to the angle of all cells.

first day (24 h) there is a strong increase in their number. At the same time, it was observed that in the next period of time (from first 24 h to about 48 h) followed by a clear change in the distance between cells. These two arguments to show that the cancer cells in the early stages highly proliferates and in the next phase, change his size and move in the direction of the crack. In addition, it was noted that a strong change in the number of cells, and the change in the distance between the cells, induced a phenomenon of changes in the angle of the cells, beginning near the scratch, then the remaining. Changes of the ratio of angle cells near scratch, to the angle of all cells is shown in Fig. 7d. Where the diagonal angle of the cell was higher, there would be expected fast and strong cell growth and migration in the direction of scratch.

## 6 Conclusions

Modern research for testing the growth of tumor cells largely based on software that calculates how changing the area of cell cultures. The most commonly used is a specialized software [6,8,14]. The results of changes of the surface area using the designed algorithm were compared to the results of the TScratch software.

The direction of changes in both cases coincided, but it not include important changes of cells migration and growth. The use of available and well-known analysis allows to determine the growth rate, cell migration globally throughout the image. They are determined parameters such as speed overgrowing scratches. However, all operate only on the changes of surface area. The study shows that it can be additionally determine the parameters that define the ways in which the migration of cancer cells. Designation of areas where clusters of cells change the angle of permits to conclude that in these places will be faster migration to the scratches. In addition, to determine if the cells themselves increase during the migration process, determining their number and the distance between them. Given the above results we can assume what will be the direction of migration in the next stages of experience. This allows you to determine the predictors, indicating a high probability of projected direction of the changes. The next stage of research will be to propose a system of automatic classification groups of images, using the parameters listed in this article. The proposed solution can be used not only in the study of the behavior of cancer cells, but also to examine the nature of changes in the wound healing processes. This of course will require the implementation of further research.

## References

1. Acharya, T., Ray, A.K.: Image Processing: Principles and Applications. Wiley InterScience, New York (2006)
2. American Cancer Society: Global Cancer Facts & Figures, 2nd edn. American Cancer Society, Atlanta (2011)
3. Angenent, S., Pichon, E., Tannenbaum, A.: Mathematical methods in medical image processing. Bull. Am. Math. Soc. **43**, 365–396 (2006)
4. Ridge, C.A., McErlean, A.M., Ginsberg, M.S.: Epidemiology of lung cancer. Semi. Intervent. Radiol. **30**(2), 93–98 (2013)
5. Eccles, S.A., Box, C., Court, W.: Cell migration/invasion assays and their application in cancer drug discovery. Biotechnol. Annu. Rev. **11**, 391–421 (2005)
6. Flate, E., Stalvey, J.R.D.: Motility of select ovarian cancer cell lines: effect of extracellular matrix proteins and the involvement of PAK2. Int. J. Oncol. **45**(4), 1401–1411 (2014)
7. Ferlay, J., Steliarova-Foucher, E., Lortet-Tieulent, J., Rosso, S., Coebergh, J.W.W., Comber, H., Forman, D., Bray, F.: Cancer incidence and mortality patterns in Europe: estimates for 40 countries in 2012. Eur. J. Cancer **49**, 1374–1403 (2013)
8. Peng, J., Ramesh, G., Sun, L., Dong, Z.: Impaired wound healing in hypoxic renal tubular cells: roles of hypoxia-inducible factor-1 and glycogen synthase kinase $3\beta/\beta$-catenin signaling. J. Pharmacol. Exp. Ther. **340**(1), 176–184 (2012)
9. Koprowski, R., Korzyńska, A., Zieleźnik, W., Wróbel, Z., Małyszek, J., Stępień, B., Wójcik, W.: Influence of the measurement method of features in ultrasound images of the thyroid in the diagnosis of Hashimoto's disease. BioMed. Eng. OnLine **11**, 91 (2012)
10. Hollings, N., Shaw, P.: Diagnostic imaging of lung cancer. Eur. Respir. J. **19**, 722–742 (2002)
11. Purandare, N.C., Rangarajan, V.: Imaging of lung cancer: implications on staging and management. Indian J. Radiol. Imaging **25**(2), 109–120 (2015)

12. Moller, S., Kristensen, C.S., Poulsen, L.K., Cartensen, J.M., Molin, S.: Bacterial growth on surfaces: automated image analysis for quantification of growth rate-related parameters. Appl. Environ. Microbiol. **61**, 741–748 (1995)
13. Sivamani, R.K., Pullar, C.E., Manabat-Hidalgo, C.G., Rocke, D.M., Carlsen, R.C., Greenhalgh, D.G., Isseroff, R.R.: Stress-mediated increases in systemic and local epinephrine impair skin wound healing: potential new indication for beta blockers. PLoS Med. **6**(1), e1000012 (2009)
14. El-Hamoly, T., Hegedűs, C., Lakatos, P., Kovács, K., Bai, P., El-Ghazaly, M.A., El-Denshary, E.S., Szabó, É., Virág, L.: Activation of poly(ADP-Ribose) polymerase-1 delays wound healing by regulating keratinocyte migration and production of inflammatory mediators. Mol. Med. **20**(1), 363–371 (2014)
15. Aneknan, P., Kukongviriyapan, V., Prawan, A., Kongpetch, S., Sripa, B., Senggunprai, L.: Luteolin arrests cell cycling, induces apoptosis and inhibits the JAK/STAT3 pathway in human cholangiocarcinoma cells. Asian Pac. J. Cancer Prev. **15**(12), 5071–5076 (2014). Article 54
16. Chae, D.S., Han, S., Son, M., Kim, S.W.: Stromal vascular fraction shows robust wound healing through high chemotactic and epithelialization property. Cytotherapy, pii: S1465-3249(17)30032-4, 13 February 2017. doi:10.1016/j.jcyt.2017.01.006
17. Zismanov, V., Drucker, L., Gottfried, M.: ER homeostasis and motility of NSCLC cell lines can be therapeutically targeted with combined Hsp90 and HDAC inhibitors. Pulm. Pharmacol. Ther. **26**(3), 388–394 (2013)

# Frame-to-Frame Visual Odometry: The Importance of Local Transformations

Aleksander Kostusiak(✉)

Institute of Control and Information Engineering,
Poznań University of Technology, ul. Piotrowo 3A, 60-965 Poznań, Poland
aleksander.m.kostusiak@doctorate.put.poznan.pl

**Abstract.** Trajectory estimation is of pivotal importance for mobile robots. Visual Odometry (VO) allows localizing a robot from passive vision data in frame-to-frame fashion. The VO problem can be solved in different ways, hence an evaluation of these algorithms in the context of real benchmark data is interesting. We focus on feature-based $n$-point methods based on RGB images. These methods used in monocular vision allow for camera rotation estimation, but only a few of them provide translation estimates up to the unknown scale. In the context of the use of commodity RGB-D cameras, we also compare these methods with the Kabsch algorithm, which uses full depth information.

**Keywords:** Sparse visual odometry · Trajectory estimation · Monocular vision · Comparison · Benchmark

## 1 Introduction

The trajectory estimation problem is in the spotlight for many years, yielding many different solutions. The best approaches belong to the dense solution group and use all the information seen by the camera. Because this is computationally expensive and requires GPU acceleration, it is almost impossible to use such an approach in mobile robotics in a larger scale. For these reasons, we are concerned with the existing feature-based (sparse) approaches. While working with a single passive (RGB) camera it is impossible to obtain full depth information about the scene and in turn to accurately determine the scale of translation between two frames. However, it is still possible to compute rotational components of the trajectory and to use them for constraining the orientation of the robot/camera. This is of high importance, as small errors here can result in large translational errors. It was shown in [2] that a SLAM system can benefit from being augmented with the visual estimation of the frame-to-frame orientation change. In the last few years, the focus in visual trajectory estimation was put on Visual SLAM and Visual Odometry. The former type of approach builds a map, and then estimates the robot position with respect to it. The first working real-time Visual SLAM was demonstrated by Davison and Murray [11]. The latter type of approach

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_37

tries to achieve the same goal without building a map, however with a larger number of frame-to-frame measurements. A comprehensive overview of the VO methods is presented in [12,13], and in [14], where VO and SLAM algorithms are nicely and briefly described. The rest of this paper is structured as follows: Sect. 2 presents the used libraries and related work, Sect. 3 briefly describes the n-point relative pose computation algorithms, Sect. 4 details the methodology used to obtain both the quantitative and qualitative results, Sect. 5 describes and comments the obtained results, and Sect. 6 concludes the paper.

## 2 State of the Art

Currently, there are several libraries that help in obtaining rotation from RGB images, like OpenCV [8], OpenMVG and OpenGV. All of the mentioned libraries but OpenCV contain the 5-point Nister [3,4], Stewenius [5], and 7-point and 8-point methods, which are in OpenCV, for finding the essential matrices from 2D-2D correspondences. OpenCV has two last methods to return fundamental matrix only. OpenMVG contains a 5-point method that uses Nister and Stewenius constraint expansions that find the essential matrix. The richest library of that kind of algorithms is OpenGV allowing for [9]: calculation pose of the camera from 2D-3D correspondences between points in the world frame and bearing vectors in the camera frame (this states as central absolute pose problem), and in multiple camera frames (non-central absolute pose case), finding the pose of one camera with respect to another camera given a number of 2D-2D correspondences between the bearing vectors in the camera frames (central relative pose), and in multiple camera frames (non-central relative pose problems). Some algorithms may return only rotation, only translation, several different probable rotations, rotation and translation, several probable rotations and translations, essential matrix, several probable essential matrices, or several probable complex essential matrices. In the past Murphy et. al. presented [14] an evaluation of three VO techniques: patch-based, Structure from Motion, and the VO approach of [15], but in a monocular version. Recently, there is a work in semi-dense approaches [1] that labels itself as VO method claiming frame-to-frame trajectory estimation based on semi-dense inverse depth map based. Nonetheless, to create those semi-dense depth maps their approach uses the history of RGB frames to find a possibly oldest frame, in which the pixel from the current frame was seen. That leads to building a form of map, and to the indirect use of it. Consequently, this type of solution does not fit well to the VO definition, being partially a SLAM system, yet without trajectory optimization. Worth mentioning are the experiments trying to combine both worlds of visual SLAM and VO. Earlier, Schmidt et al. [2] proposed to augment SLAM method by replacing orientation change every few steps of EKF prediction function with the one provided by separately running VO system. A comparison of SLAM and VO approaches for RGB-D data was presented in [6]. As for now, there is very little work done concerning evaluation of methods that estimate the frame-to-frame orientation change. Recently, Hartman et. al. [7] used 5-point Stewenius algorithm with a modified RANSAC approach. Ground truth data was collected with the use of IMU and GPS devices while the one used in experiments presented here were collected by a high-resolution multi-camera system [21], or the Vicon motion capture system [19].

## 3 Used Algorithms

In this section, we briefly describe used algorithms for solving geometric vision problem. We used SURF, ORB and AKAZE feature detector-descriptor pairs for preliminary image processing. 8-point [16,17] algorithm is here the oldest algorithm for computing essential or fundamental matrix, depending if the calibration has been performed, based on 8 or more matched points. The essential matrix can be defined as:

$$\mathbf{u'Fu} = 0 \qquad \text{(a)} \qquad \text{or} \qquad \mathbf{u'Eu} = 0 \qquad \text{(b)} \qquad (1)$$

where u' and u are matched homogeneous image points, because of that the last element equals to 1, and E and F stand for the Essential and Fundamental matrix (without calibration data). The entry points are normalized to ensure that all entries of respective matrices will be treated approximately equally. After representing matrix Eq. 1 as a vector, set of linear equations is obtained:

$$\mathbf{Af} = 0 \qquad \text{(a)} \qquad \text{or} \qquad \mathbf{Ae} = 0 \qquad \text{(b)} \qquad (2)$$

where A is the equation matrix, e and f are 9-vector containing the entries of the matrix E or F. The equation is solved under the singularity constraint $\|f\|$or $\|e\|$= 1, where $\|e\|$and $\|f\|$are the norms, by checking if it is of rank 2 and if it is singular and by enforcing if it is not the case. Finally, the result matrix is replaced by closes singular matrix under Frobenius norm. Nister's [3,4] algorithm computes Essential matrix given 5 correspondences, which gives a rise to essential matrix constraint of the following form:

$$\mathbf{EE}^T\mathbf{E} - \frac{1}{2}trace(\mathbf{EE}^T)\mathbf{E} = 0 \qquad (3)$$

After representing matrix Eq. 3 as vector, set of the linear equation a 5×9 matrix is obtained. Next, four vectors $\tilde{\mathrm{X}}$, $\tilde{\mathrm{Y}}$, $\tilde{\mathrm{Z}}$, $\tilde{\mathrm{W}}$ that spans the right nullspace of that matrix are calculated.

$$\mathbf{E} = x\mathbf{X} + y\mathbf{Y} + z\mathbf{Z} + w\mathbf{W} \qquad (4)$$

They correspond directly to four matrices constraining the form of essential matrix 4 which is in turn inserted into nine cubic constraints 3. Calculations are performed by using Gröwner basis. Scalars x, y, z, w are defined up to a common scale factor and it is assumed that w = 1. As the output algorithm can return up to 10 essential matrices, but some of them are internally rejected. Kneip [18] proposed a direct method for finding exact rotation between two images independently of the translation. This results from the fact that rotational optical flow is fundamentally different from translational one. The constraints on the rotation are then as follows:

$$|(f_1 \times \mathbf{R}f_1')(f_2 \times \mathbf{R}f_2')(f_3 \times \mathbf{R}f_3')| = 0 \tag{5}$$

Because rotation encodes 3° of freedom at least 3 epipolar plane normal coplanarity constraints are required to fully constrain rotation. Two additional features allow for building the necessary system of equation:

$$\begin{aligned} |(f_1 \times \mathbf{R}f_1')(f_2 \times \mathbf{R}f_2')(f_3 \times \mathbf{R}f_3')| = 0 \\ |(f_1 \times \mathbf{R}f_1')(f_2 \times \mathbf{R}f_2')(f_4 \times \mathbf{R}f_4')| = 0 \\ |(f_1 \times \mathbf{R}f_1')(f_2 \times \mathbf{R}f_2')(f_5 \times \mathbf{R}f_5')| = 0 \end{aligned} \tag{6}$$

To solve this Eq. 6 Gröne basis is used. As the output algorithm can return up to 20 different rotation matrices, resulting from 10 essential matrices, but some of them are internally rejected. Eigensolver method tries to find the eigenvalue that minimizes the fallowing function 7:

$$\mathbf{R} = argmin_{\mathbf{R}} \lambda_{M,min} \tag{7}$$

where R is the rotation transforming the i-th bearin vector $f_i$ to the corresponding one seen from the second frame $f_i'$. M has rank at most 2 and is a real symmetric and positive- defined matrix of fallowing form: $M = \sum_{i=1}^{n}(f_i \times \mathbf{R}f_i')(f_i \times \mathbf{R}f_i')^T$. At best a non-linear optimization over three parameters is needed because rotation has 3° of freedom. To solve the problem Kneip et.al. [18] are enforcing the first-order partial derivates of $\lambda_{M,min}$ to be zero and then are using Levenberg-Marquardt scheme. Also, this algorithm computes translation direction (but does not explicitly return it) as automatically given by the eigenvector that corresponds to the smallest eigenvalue. We have implemented the Kabsch algorithm [22,23] with the help of information found in [24]. As this algorithm computes transformation of two aligned points sets, the camera movement is the opposite.

## 4 Experimental Methodology

The whole program is constructed similarly to the work presented in [10]: firstly detector/descriptor algorithm is used to extract silent features. Then RANSAC technique, with varying ejection threshold (and for Eigensolver method also varying confidence) is performed twice to obtain best results. At this stage, two different error measures are used: reprojection error and a simple Euclidean distance between bearing vectors. The second method omits the translation, assuming it is sufficiently small. Finally, a trajectory is computed with all the matches remaining. Kneip algorithm is an exception- it only accepts from 5 to 8 bearing vectors pairs. It is also the only algorithm that does not compute translation, thereby to be able to compute reprojection error we employ a two-point algorithm, that is also contained in the OpenGV library. Because some algorithms, namely Kneip, Nister and Eightpoint return several results, the best one is chosen based on appropriate error measure. In a case of an empty result set, resulted from in-algorithm suppression, 8 points are randomly chosen from the inliers set to compute rotation and translation as long as it does not exceed maximal RANSAC iterations. If the number of remaining inliers is insufficient for trajectory estimation the whole pipeline breaks as unable to correctly recover trajectory. For experiments we used 2 different trajectories: *putkk_Dataset_1_Kin_1* data set described in [20], is further referred as dataset1, and *fr3_long_office_household* which will be referred as dataset2 from the TUM RGB-D Benchmark [19]. Dataset1 was collected by the Kinect sensor mounted on the moving in the laboratory wheeled robot. Ground truth was obtained from multicamera vision system PUT Ground Truth (PUT GT) [21]. The acquisition time of robot and GT cameras have been synchronized so no interpolation was performed. Dataset2 was collected by the hand-held Kinect camera. Ground truth was also acquired by a multicamera system but a time of acquisition was not enforced what resulted in the need for data interpolation. Timestamps are not perfectly aligned in the ground truth and data collected by the camera. To measure the quality of retrieved trajectory we use well known Absolute Trajectory Error (ATE) and Relative RPE metrics as described in [19]. The first error is the Euclidean distance between corresponding points of the ground truth trajectory and estimated one. RPE is a relative translational or rotational error of successive frames. As for evaluation Root Mean Squared Error (RMSE) of mentioned metrics are used.

## 5 Results

In this section, in order to fit the data into the tables, we abbreviated some names. The translation recovery approaches that are based on the use of Centroid are abbreviated to C. and the Scale-based methods (depth is used only in the first two frames to determine the scale) are denoted as S. Consequently, the Reprojection error is denoted as R. and the Euclidean error as E. In the first experiment, it occurred that the scene was too demanding, or the found points were insufficient for Eigensolver and Nister algorithms to retrieve full trajectory in 3 out of 4 cases. It is possible, that those algorithms could restore trajectory if a different set of points was provided but regardless of this fact it shows that those algorithms are not as robust as rest of the investigated methods. With respect to rotational RPE RMSE errors, it is questionable if Kneip algorithm managed to retrieve the trajectory. The best of n-point relative pose algorithms was the 8-point, achieving best results with a constant scale, mainly if it was ignoring the translational part in RANSAC scheme. While this depends on the initialization stage and the detected points, the differences are not large. Still, it is outperformed by Kabsch algorithm, which is using full depth information, what is demonstrated in Fig. 1. This method combined with SURF detector/descriptor works twice better in ATE error terms than with other detector/descriptor pair.

**Table 1.** Comparison of absolute trajectory errors (ATE RMSE) and relative pose errors (RPE RMSE) for the putkk_Dataset_1_Kin_1

| Algorithm | AKAZE | | | ORB | | | SURF | | |
|---|---|---|---|---|---|---|---|---|---|
| | ATE [m] | RPE [m] | RPE [deg] | ATE [m] | RPE [m] | RPE [deg] | ATE [m] | RPE [m] | RPE [deg] |
| C.E.Eigensolver | 7.267 | 0.021 | 0.372 | 7.292 | 0.021 | 0.372 | 7.261 | 0.021 | 0.372 |
| C.E.Kneip | 3.866 | 3.863 | 114.451 | — | — | — | 3.583 | 3.885 | 114.153 |
| C.E.nister | — | — | — | — | — | — | — | — | — |
| C.E.eight-point | 4.044 | 0.047 | 4.683 | 4.067 | 0.042 | 0.928 | 4.064 | 0.042 | 0.926 |
| C.R.Eigensolver | — | — | — | — | — | — | — | — | — |
| C.R.Kneip | 4.316 | 3.835 | 115.062 | 3.833 | 3.772 | 114.803 | 3.725 | 3.869 | 116.144 |
| C.R.nister | 4.090 | 0.283 | 9.031 | 3.759 | 0.230 | 6.189 | 4.123 | 0.355 | 10.094 |
| C.R.eight-point | 3.843 | 0.063 | 4.691 | 4.033 | 0.042 | 0.973 | 3.968 | 0.042 | 0.966 |
| S.E.Eigensolver | 2.043 | 0.014 | 0.372 | 2.044 | 0.014 | 0.372 | 2.032 | 0.014 | 0.372 |
| S.E.Kneip | 11.806 | 0.932 | 115.418 | 1.858 | 0.212 | 115.817 | 3.699 | 0.434 | 115.101 |
| S.E.nister | — | — | — | 2.044 | 0.014 | 3.900 | 1.974 | 0.014 | 4.375 |
| S.E.eight-point | 1.745 | 0.018 | 0.922 | 1.735 | 0.0176 | 0.9236 | 1.715 | 0.018 | 4.685 |
| S.R.Eigensolver | — | — | — | — | — | — | — | — | — |
| S.R.Kneip | 2.651 | 0.424 | 113.133 | 4.553 | 0.5737 | 113.945 | 2.869 | 0.277 | 114.681 |
| S.R.nister | 2.143 | 0.021 | 11.260 | 4.463 | 0.063 | 9.325 | 1.932 | 0.017 | 8.919 |
| S.R.eight-point | 1.861 | 0.025 | 0.956 | 1.768 | 0.023 | 0.963 | 1.699 | 0.021 | 0.955 |
| Kabsch | 0.724 | 0.019 | 0.413 | 0.714 | 0.020 | 0.430 | 0.470 | 0.020 | 0.417 |

The full comparison is further given in Table 1. In the second experiment, all but Nister coupled with AKAZE detector/descriptor were able to reconstruct trajectory. The best results were achieved by the Kabsch algorithm with the use of depth data. The best algorithm for finding a rotation from solely RGB data is 8-point algorithm if ATE error is considered but if the RPE metrics is evaluated, the Eigensolver seems to be a bit better. It seems as 8-point algorithm works slightly better with proposed in-RANSAC Euclidean error measurement than with reprojection error. This is not the case for Eigensolver algorithm, as it benefits if combined with SURF algorithm, and suffers if combined with AKAZE in the context of constant scale. Also, Eigensolver sometimes returns no solution and thus is not as reliable as the 8-point algorithm. Kneip algorithm has the biggest RPE error from all of the investigated algorithms. This is probably due to its inability to perform computations on more than 8 points. It can be assumed that it "reconstructed" trajectory, as Fig. 1 shows, when was working without depth data, having only a scale and reprojection error but this is not the case if it is used with depth information in centroid fashion or Euclidean error. Still, RPE errors are high. The reason for this are residual outliers, gaining a lot of importance in the final computation of the estimate. Other algorithms perform some form of least-squares optimization, and thus the residual outliers are strongly suppressed. The results for all investigated algorithms for fr3_long_office_household sequence are collected in Table 2 and enhanced with Fig. 2.

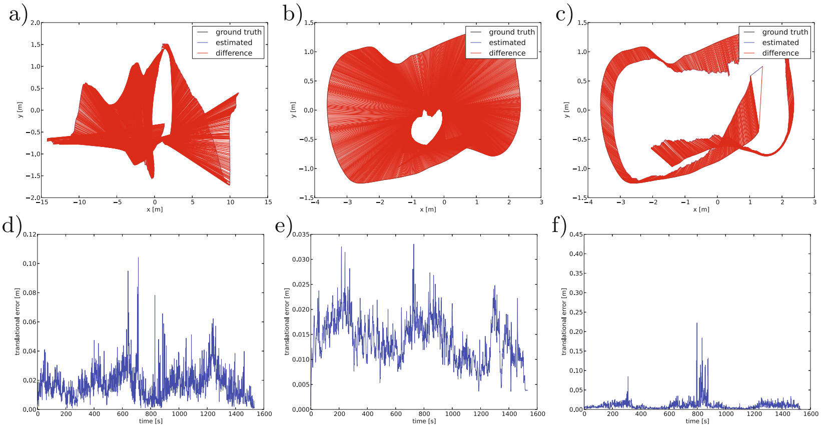


**Fig. 1.** Trajectories estimated for the putkk_Dataset_1_Kin_1 (a) ORB ATE C.E. Eigensolver, (b) SURF ATE S.E. Nister, (c) SURF ATE Kabsch, (d) ORB RPE C.E. Eigensolver, (e) SURF RPE S.E. Nister, (f) SURF RPE Kabsch

**Table 2.** Comparison of absolute trajectory errors (ATE RMSE) and relative pose errors (RPE RMSE) for the fr3_long_office_household sequence

| Algorithm | AKAZE | | | ORB | | | SURF | | |
|---|---|---|---|---|---|---|---|---|---|
| | ATE [m] | RPE [m] | RPE [deg] | ATE [m] | RPE [m] | RPE] [deg] | ATE [m] | RPE [m] | RPE [deg] |
| C.E.Eigensolver | 1.683 | 0.376 | 19.313 | 1.822 | 0.320 | 11.743 | 1.790 | 0.302 | 11.743 |
| C.E.Kneip | 2.571 | 2.628 | 131.423 | 3.211 | 2.925 | 131.514 | 3.334 | 2.918 | 130.940 |
| C.E.nister | — | — | — | 2.066 | 1.224 | 43.236 | 1.636 | 1.128 | 35.555 |
| C.E.eight-point | 2.745 | 0.808 | 22.853 | 1.957 | 0.503 | 29.023 | 1.615 | 0.489 | 22.757 |
| C.R.Eigensolver | 1.467 | 0.371 | 18.975 | 2.290 | 1.275 | 48.219 | 3.211 | 0.924 | 46.888 |
| C.R.Kneip | 2.933 | 2.653 | 132.805 | 3.996 | 3.016 | 131.065 | 3.579 | 2.949 | 131.599 |
| C.R.nister | 2.028 | 1.131 | 45.637 | 2.459 | 1.842 | 73.429 | 2.316 | 1.138 | 42.402 |
| C.R.eight-point | 2.865 | 0.815 | 22.956 | 2.177 | 0.515 | 37.691 | 2.152 | 0.490 | 29.017 |
| S.E.Eigensolver | 1.972 | 0.260 | 105.122 | 1.984 | 0.269 | 11.743 | 1.913 | 0.286 | 11.743 |
| S.E.Kneip | 1.957 | 0.367 | 131.377 | 1.737 | 0.465 | 132.218 | 1.920 | 0.508 | 131.092 |
| S.E.nister | — | — | — | 1.857 | 0.349 | 42.416 | 1.600 | 0.400 | 33.549 |
| S.E.eight-point | 1.181 | 0.368 | 22.819 | 1.467 | 0.352 | 22.895 | 1.202 | 0.358 | 22.777 |
| S.R.Eigensolver | 1.803 | 0.244 | 21.157 | 1.998 | 0.253 | 49.985 | 1.935 | 0.269 | 49.098 |
| S.R.Kneip | 7.865 | 3.266 | 131.034 | 2.076 | 0.623 | 131.848 | 2.105 | 1.173 | 132.467 |
| S.R.nister | 1.916 | 0.345 | 46.895 | 2.140 | 0.543 | 71.752 | 1.869 | 0.513 | 39.916 |
| S.R.eight-point | 1.246 | 0.354 | 22.887 | 1.786 | 0.394 | 29.294 | 1.925 | 0.391 | 29.788 |
| Kabsch | 0.560 | 0.042 | 1.267 | 0.576 | 0.046 | 1.372 | 0.619 | 0.044 | 1.434 |

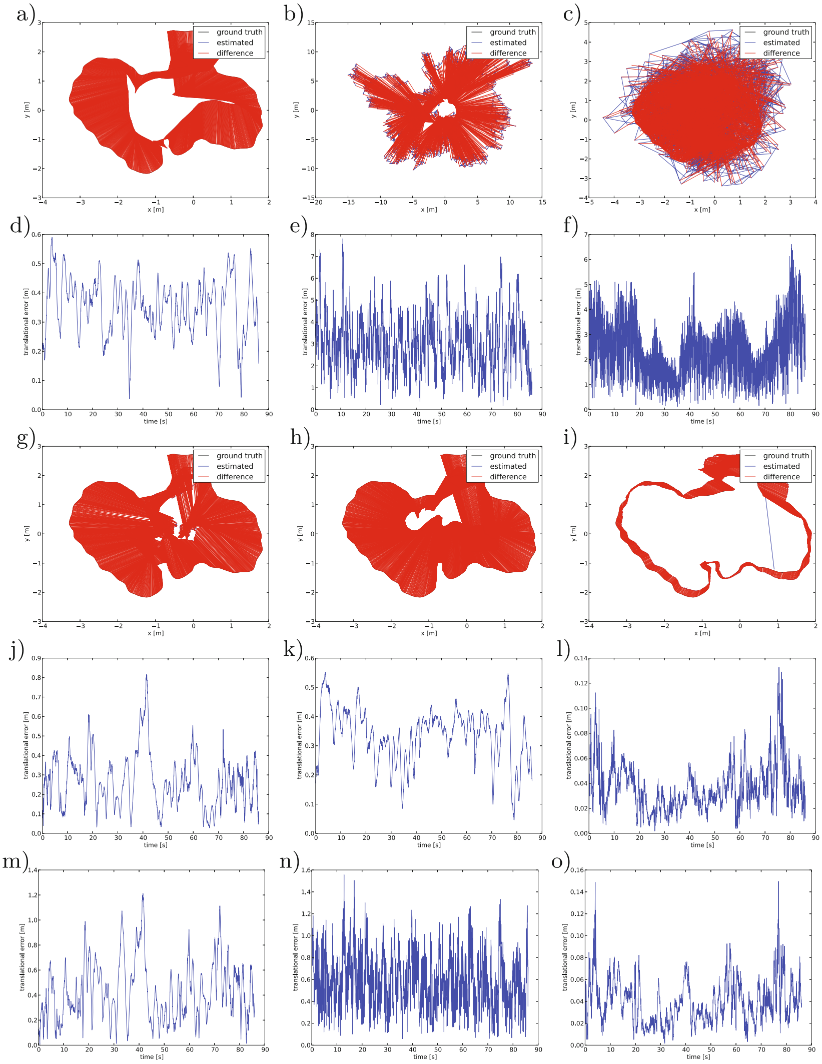


**Fig. 2.** Trajectories estimated for the fr3_long_office_household, (a) AKAZE ATE S.E. 8-point, (b) AKAZE ATE S.R. Kneip, (c) AKAZE ATE C.R. Kneip, (d) AKAZE RPE S.E. 8-point, (e) AKAZE RPE S.R. Kneip, (f) AKAZE RPE C.R. Kneip, (g) SURF ATE S.E. Eigensolver, (h) ORB ATE S.E. Nister, (i) AKAZE ATE Kabsch, (j) SURF RPE S.E. Eigensolver, (k) ORB RPE S.E. Nister, (l) AKAZE RPE Kabsch, (m) SURF RPE C.E. 8-point, (n) ORB RPE S.R. Kneip, (o) SURF RPE Kabsch

## 6 Conclusions

Taking into account the presented experimental results we can conclude, that the best algorithm for solving the frame-to-frame motion estimation problem is the Kabsch algorithm, which however requires depth data, e.g. from an RGB-D sensor. In passive monocular systems, the 8-point algorithm won the competition, as the Eigensolver, although quite accurate, not always was able to deliver a solution. Whenever no depth data is available, the 8-point algorithm can restore a reasonable trajectory in comparison with other methods. The remaining algorithms usually performed by far worse than the two previously mentioned methods. The superiority of the 8-point algorithm may result from a better least-square optimization. Also, when working with real, imperfect data, the assumptions underlying the 8-point algorithm are more realistic than the stricter assumptions of the 5-point algorithm. It seems that the choice of the detector/descriptor has little influence compared to the error metrics used in RANSAC. There is no significant difference in performance resulting from using different detector/descriptor algorithms. The Kabsch algorithm in one scene works better with SURF, but in another scene, AKAZE and ORB perform slightly better. This shows that for different scenes, a different approach to select the salient point features is needed and that there is no perfect feature detector yet. It seems that the 8-point algorithm works better with AKAZE or SURF features than with ORB. The ORB is known from its fast computation and thus is considered for the computational efficiency, but its performance in our experiments was rather disappointing. Taking into account that AKAZE is an open-source software, we also advise considering this detector/descriptor pair instead of SURF. In our future work, we plan to consider also some detector/descriptor combinations, such as ORB-AKAZE, which seem to be interesting.

## References

1. Engel, J., Sturm, J., Cremers, D.: Semi-dense visual odometry for a monocular camera. In: IEEE International Conference on Computer Vision (ICCV) (2013)
2. Schmidt, A., Kraft, M., Fularz, M., Domagala, Z.: Comparative assessment of point feature detectors and descriptors in the context of robot navigation. J. Autom. Mob. Rob. Intell. Syst. **7**(1), 11–20 (2013)
3. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell. **26**, 756–770 (2004)
4. Nistér, D.: An efficient solution to the five-point relative pose problem, In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), vol. 2, pp. 195–202 (2003)
5. Stewénius, H., Engels, C., Nistér, D.: Recent developments on direct relative orientation. ISPRS J. Photogrammetry Remote Sens. **60**, 284–294 (2006). http://www.vis.uky.edu/ stewe/FIVEPOINT/
6. Belter, D., Nowicki, M., Skrzypczyński, P.: On the performance of pose-based RGB-D visual navigation systems. In: Computer Vision – ACCV 2014. LNCS, vol. 9004, pp. 407–423. Springer (2015)
7. Hartmann, W., Havlena, M., Schindler, K.: Visual gyroscope for accurate orientation estimation. In: Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision (WACV 2015), pp. 286–293. IEEE Computer Society, Washington, DC (2015)

8. Bradski, G.: Opencv_library. Dr. Dobb's J. Softw. Tools (2000). http://opencv.org/
9. Kneip, L., Furgale, P.: OpenGV: a unified and generalized approach to real-time calibrated geometric vision. In: Proceedings of The IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China (2014)
10. Kostusiak, A.: The comparison of keypoint detectors and descriptors for registration of RGB-D data. In: Szewczyk, R. et al. (eds.) Challenges in Automation, Robotics and Measurement Techniques. AISC, vol. 440, pp. 609–622. Springer (2016)
11. Davison, A.J., Murray, D.W.: Simultaneous localisation and map-building using active vision. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 865–880 (2002)
12. Fraundorfer, F., Scaramuzza, D.: Visual odometry: Part I the first 30 years and fundamentals. IEEE Rob. Autom. Mag. **18**(4), 80–92 (2011)
13. Fraundorfer, F., Scaramuzza, D.: Visual odometry: Part II matching, robustness and applications. IEEE Rob. Autom. Mag. **19**(2), 78–90 (2012)
14. Murphy, L., Morris, T., Fabrizi, U., Warren, M., Milford, M., Upcroft, B., Bosse, M., Corke, P.: Experimental comparison of odometry approaches. In: Experimental Robotics: The 13th International Symposium on Experimental Robotics. Springer Tracts in Advanced Robotics, vol. 88, pp. 877–890 (2013)
15. Warren, M., McKinnon, D., He, H., Upcroft, B.: Unaided stereo vision based pose estimation. In: Australasian Conference on Robotics and Automation, Brisbane, ARAA (2010)
16. Longuet-Higgins, H.: Readings in Computer Vision: Issues, Problems, Principles, and Paradigms. Morgan Kaufmann Publishers Inc., San Francisco (1987)
17. Hartley, R.: In defense of the eight-point algorithm. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **19**(6), 580–593 (1997)
18. Kneip, L., Siegwart, R., Pollefeys, M.: Finding the exact rotation between two images independently of the translation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2012)
19. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D., A benchmark for the evaluation of RGB-D SLAM systems. In: Proceedings of the IEEE RSJ International Conference on Intelligent Robots and Systems, Vilamoura, pp. 573–580 (2012)
20. Kraft, M., Nowicki, M., Schmidt, A., Fularz, M., Skrzypczyński, P.: Toward evaluation of visual navigation algorithms on RGB-D data from the first- and second-generation Kinect. Mach. Vis. Appl. **28**(1), 61–74 (2017)
21. Schmidt, A., Kraft, M., Fularz, M., Domagala, Z.: The registration system for the evaluation of indoor visual SLAM and odometry algorithms. J. Autom. Mob. Rob. Intell. Syst. **7**(2), 46–51 (2013)
22. Kabsch, W.: A solution for the best rotation to relate two sets of vectors. Acta Crystallogr. **32**, 922 (1976)
23. Kabsch, W.: A discussion of the solution for the best rotation to relate two sets of vectors. Acta Crystallogr. **A34**, 827–828 (1978)
24. Nghia, H.: Finding optimal rotation and translation between corresponding 3D points (2011). http://nghiaho.com/?page_id=671

# Chiron: A Robust Recommendation System with Graph Regularizer

Saber Shokat Fadaee[1], Mohammad Sajjad Ghaemi[2], Hossein Azari Soufiani[3], and Ravi Sundaram[1](✉)

[1] College of Computer and Information Science,
Northeastern University, Boston, USA
{saber,koods}@ccs.neu.edu

[2] École Polytechnique de Montréal, Montreal, Canada
mohammad-sajjad.ghaemi@polymtl.ca

[3] Columbia Business School, New York City, USA
Hazarisoufiani18@gsb.columbia.edu

**Abstract.** Recommendation systems have been widely used by commercial service providers for giving suggestions to users. Collaborative filtering (CF) systems, one of the most popular recommendation systems, utilize the history of behaviors of the aggregate user-base to provide individual recommendations and are effective when almost all users faithfully express their opinions. However, they are vulnerable to malicious users biasing their inputs in order to change the overall ratings of a specific group of items. CF systems largely fall into two categories - neighborhood-based and (matrix) factorization-based - and the presence of adversarial input can influence recommendations in both categories, leading to instabilities in estimation and prediction. Although the robustness of different collaborative filtering algorithms has been extensively studied, designing an efficient system that is immune to manipulation remains a challenge. We propose a novel *hybrid* recommendation system with an adaptive graph user/item similarity-regularization - **Chiron**. Chiron ties the performance benefits of dimensionality reduction (via factorization) with the advantage of neighborhood clustering (through regularization). We demonstrate, using extensive comparative experiments, that Chiron is resistant to manipulation by large and lethal attacks.

## 1 Introduction

Users of commercial service providers such as Netflix, Spotify, and Amazon are provided with a large selection of recommended choices while using these online services. Recommendation systems aid users in the challenging task of finding the best video, music, book, or product out of all the possible options that they can have while using these systems. In this regard, collaborative filtering-based recommendation systems play an increasing role in helping people locate their favorite items in an immense dataset. In addition to providing helpful recommendations to users, these systems are also beneficial for the companies in raising

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_38

their sales. However, since a good recommendation usually results in increased sales, some might find it profitable to shill recommendation systems by providing false information. “Collaborative filtering” algorithms predict how much a user prefers a set of items, and produce a ranked list of items that would benefit or match her interests the most. In recommendation systems based on collaborative filtering, users rate specific items and receive recommendations for unrated ones. All the different systems are vulnerable to malicious attackers intending to manipulate the recommendations to suit their needs. Such attackers are known as “shills” and those attacks have been referred to as “shilling” or “Sybil” attacks [16]. We introduce Chiron[1], a robust recommendation system. We conduct extensive experimental studies to compare the robustness of our algorithm with current state-of-the-art methods. Our experimental results indicate that Chiron is the most robust recommendation system, and the presence of an attack does not affect its performance. While many collaborative filtering methods are prone to overfitting, we prevent over-fitting by introducing a smart regularization technique which takes users’ and items’ similarities into account in the context of local graph estimation of the marginal probability of users and items.

### 1.1 Related Work

Many psychological studies have shown that people tend to agree with opinions of others regardless of their factual correctness. Cosley et al. [7] showed that prediction manipulation in a recommendation system can affect people in that system and, in some cases, mislead people into accepting a negative and unfitting recommendation. Therefore, people’s perceived value of items are influenced by the ratings of a recommendation system. Lam et al. [8] and O’Mahony et al. [15] showed that many of the well-known recommendation systems are vulnerable to attacks and proposed different methods to distinguish honest raters from attackers. However, using detection algorithms as a preprocessing step can be computationally expensive. Therefore, others have proposed building robust recommendation systems [13,14]. While recommendation systems have been widely investigated, less attention has been devoted to studying their vulnerability to manipulations. Mehta et al. [12] discussed a robust recommendation system’s characteristics. We propose a new generative model for recommendation systems that not only considers the users’ evaluations of items, but also takes into account the items’ evaluations of users. In some cases, items could be informative, and have their own evaluations of users. This extra information could be used to add priors to the system in order to promote users who give honest ratings. We incorporate these evaluations as regularization terms inspired by the local similarities underlying the graph structure.

[1] Chiron was the most important Centaur in Greek mythology, and centaurs are hybrid creatures. Since our model is a hybrid-recommendation system that factorizes the user/item matrix and uses the neighborhood information, we picked this name.

## 2 Contribution

We propose a new model subject to certain constraints which considers the quality of a rater (e.g. customers providing feedback), as well as the quality of an item (e.g. a restaurant or a product). Then we apply the neighborhood information using a graph regularizer to approximate a geometrical structure of the distribution of users and items in the latent space. Fitting an arbitrary model to the observed data is usually prone to over-fitting by increasing the variance of the error term. In order to prevent a model from over-fitting, researchers deploy several techniques, including cross validation and regularization. Regularization deals with a trade-off between bias and variance of an estimator. It has to ensure a model is complex enough to encompass the observed data smoothly, simultaneously, keeping the model as simple as possible in order to generalize the unobserved data. We have two regularization terms in our model (4). One controlling the user similarity in the intrinsic space of their underlying rated items. The other tunes the item's rating measured by the geometry of the users' distribution. The intuition behind this model is simple yet efficient. Whether someone likes an item or not depends on the affinity between her latent preferences and the item's latent attributes (shown by $\theta$ in our model), and the influence of her friends on her choices and her decision regarding similar items in the past. We add extra information regarding a person and an item local neighborhood to this model by adding regularization to the log likelihood of our model. Our research is inspired by the work of Belkin et al. [3], and Cai et al. [6]. We construct our model by first computing a graph from the observed ratings and creating an affinity matrix by measuring the similarities between ratings of different users. Then we incorporate a regularization term into the model which is sufficiently smooth with respect to the intrinsic structure collectively revealed by both observed and missing data. Our model has two basic assumptions:

1. Local assumption: nearby users are likely to have the same rating.
2. Global assumption: users on the same structure (considered as an underlying manifold) are more likely to have the same rating.

The relationship between users can be thought of as an undirected weighted graph, in which the weights reflect the affinity between the ratings of those users. Our regularization propagates the rating's value through the edges attached to it. The value transferred to each user is proportional to its weight (similar to the graph construction in [18]). We demonstrate how this approach is robust to attacks by malicious users. Additionally, we examine our model's performance on three real-world datasets, and compare the success rate of attacks on our system as well as the state-of-the-art collaborative filtering recommendation systems. We also compare the running time of different methods on a different dataset and conclude that Chiron is not only robust to attacks, but is also the second fastest algorithm among the varied collection of current recommendation systems.

## 3 Model

Suppose we have $m$ raters who rate $n$ items with a score in the range of 1 to $K$. Let $P_i = (p_{i1}, .., p_{iK})$ be the vector of probabilities of different ratings for item $i$, and $Q_j = (q_{j1}, .., q_{jK})$ be the vector of probabilities of different ratings by rater $j$. From now on we use index $j$ to refer to a user, and index $i$ to refer to an item in our model. Let $r_{ij} = k$ if user $j$ gives rating of $k$ to item $i$ and let $z_{ijk}$ denote the corresponding indicator variable, i.e. $z_{ijk} = 1$ if $r_{ij} = k$, and $z_{ijk} = 0$ otherwise. Now in Chiron, the probability that an item $i$ receives rating $k$ by user $j$ is calculated as follows:

$$Pr(r_{ij} = k) \approx \theta_{ijk} = \frac{p_{ik} q_{jk}}{\sum_{l=1}^{K} p_{il} q_{jl}} \tag{1}$$

Different combinations of $P_i$, $Q_j$, and corresponding $Pr(r_{ij} = k)$ for three cases are illustrated in Fig. 1. For example in the first case, we have a user $j$ who gives a high rating to most items (e.g. rates items only when she loves them). For an item $i$ which receives a poor rating by most users and other ratings with the same probability, Chiron predicts that user $j$ gives either the lowest or the highest rating to that item with a high probability. In the second case, user $j$ rates items as 1 to 5 with an ascending probability, and item $i$ is rated with a descending probability. In this case, Chiron predicts that this user would rate this item as 1,2,4, or 5 with the same probability and rates it as 3 with a lower probability. In the last case, user $j$ tends to rate items as either very good or very bad. On the other hand, we have an item $i$ which is mostly rated as average (3). Chiron predicts that user $j$ gives any rating to item $j$ with the same probability. $\theta$ captures users' rating habits. The log likelihood function of 1 is:

$$\begin{aligned} \mathcal{L}(P, Q; Z) &= \sum_{ijk} z_{ijk} \log \frac{p_{ik} q_{jk}}{\sum_l p_{il} q_{jl}} && (2) \\ &= \sum_{ijk} z_{ijk} \log p_{ik} q_{jk} - \sum_{ij} \log \sum_l p_{il} q_{jl} && (3) \end{aligned}$$

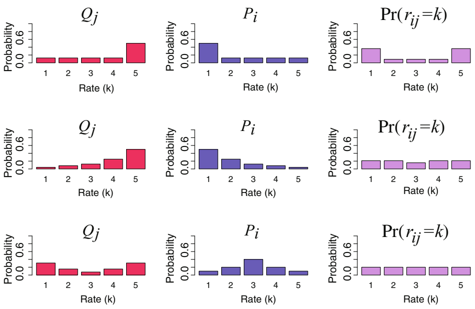


**Fig. 1.** Examples of different combinations of probabilities of ratings given by user $j$ (as $Q_j$) and her ratings for an item $i$ (as $P_i$), and the corresponding probability that item $i$ is rated as $k$ by user $j$ as $Pr(r_{ij} = k)$.

## 4 Estimation

The model defined in 3 is our fitting constraint, which means an appropriate estimating function should not change too much from the initial observed values. Now we add the smoothness constraints, which imply that a reasonable estimating function should not change too much between nearby users and corresponding ratings. The trade-off between these two competing constraints is captured by a positive parameter $\lambda_1$, and its counterpart $1 - \lambda_1$ to make a convex combination of two regularization terms. This way, one can increase the influence of users so that the effect of items will decrease. We set the value of hyper-parameters $\lambda_1$ using cross-validation.

$$\mathcal{L}'(P, Q; Z) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{K} \Bigg( z_{ijk} \log \frac{p_{ik} q_{jk}}{\sum_l p_{il} q_{jl}} \\ + \lambda_1 q_{jk} (d^Q_{jj} q_{jk} - \sum_{t=1}^{m} q_{tk} w^Q_{tj}) + (1 - \lambda_1) p_{ik} (d^P_{ii} p_{ik} - \sum_{t=1}^{n} p_{tk} w^P_{ti}) \Bigg) \tag{4}$$

In formula 4 $w^Q$ and $w^P$ represent the graph underlying the data for users and items respectively. We represent those graphs with $G^Q = (V^Q, E^Q)$, and $G^P = (V^P, E^P)$. In $G^Q$ and $G^P$ users or items with similar ratings are connected to each other. To construct the weight matrix $w^Q$ (we use similar method for computing $w^P$), we find 10 nearest users for each user in $Q$ using Pearson correlation (to construct $E^Q$). Then we apply the following kernel for each connected pair $(j_1, j_2)$:

$$w^Q_{j_1 j_2} = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ij_1k} z_{ij_2k}$$

$w^Q_{j_1 j_2}$ measures how many times both users $j_1$ and $j_2$ give the same rating to the same item. We use a regularization term that keeps the model flexible enough to assign different probabilities to $q_{j_1k}$, and $q_{j_2k}$ for distant users $j_1$, and $j_2$ in order to keep them away from each other. This difference is proportional to the small weight in terms of rating similarity that connects them weakly. The regularization term $w^Q_{j_1 j_2}(q_{j_1k} - q_{j_2k})^2$ best captures this property, which means it pushes the $(q_{j_1k} - q_{j_2k})$ toward zero for strong similarity between $(j_1, j_2)$ pair with similar rating. Using local assumption of nearby users have similar ratings, the discrete $k = 10$-nn graph estimates the global manifold for the underlying users. The observed ratings can propagate through this estimated graph to impute the missing values. Consequently, in order to spread the information among users, Laplacian of the constructed graph plays the central role in predicting the missing data from the observed values. Laplacian denoted by $L^Q = D^Q - W^Q$, is considered in this study where $D^Q$ is a diagonal matrix whose entries are $d^Q_{jj} = \sum_t w^Q_{jt}$. $L^Q$ is a symmetric and positive semi-definite matrix. This representation allows the information of ratings propagates smoothly between users with probability that is proportional to the weight between them. For the sake of calculating the stochastic element-wise gradient, we expand the $w^Q_{j_1 j_2} \sum_{k=1}^{K} (q_{j_1 k} - q_{j_2 k})^2$ term within the likelihood cost function as the sum element-wise notation expressed in Eq. 4. The model defined in 3 minimizes the variance term for the true estimator $\theta$. The class of unconstrained models are usually prone to overfitting. At the same time simple models suffer from underfitting due to increase in bias. We suggest an adaptive regularization technique based on locality preservation to compromise both issues. Hence, we consider a specific assumption regarding the connection between the marginal and the conditional distributions of $P$ and $Q$. Let us assume that if two items $p_1, p_2 \in P$ share close ratings in the form of intrinsic geometry of $Pr(P)$, then this implies the conditional distributions of users $Pr(Q|p_1)$ and $Pr(Q|p_2)$ on the constructed manifold with respect to items are correspondingly alike. In other words, the conditional probability distribution $Pr(Q|p_i)$ varies smoothly along the geodesics in the true geometrical shape of $Pr(P)$. Therefore, we have two regularization terms, one controls user similarities in the intrinsic space of their underlying rated items. The other tunes rating of items using the geometry of users distribution. In order to compute the maximum likelihood of our model, we take the partial derivative of Eq. 4 with respect to $P$ and $Q$:

$$\frac{\partial \mathcal{L}(P,Q;Z)}{\partial q_{j^*k^*}} = \sum_i z_{ij^*k^*} \frac{p_{ik^*}}{q_{j^*k^*}} - \sum_{ij} \frac{p_{ik^*}}{\sum_l p_{il} q_{jl}} + \lambda_1 (d^Q_{j^*j^*} q_{j^*k^*} - \sum_{t=1}^{m} q_{tk^*} w^Q_{tj^*})$$

$$\frac{\partial \mathcal{L}(P,Q;Z)}{\partial p_{i^*k^*}} = \sum_j z_{i^*jk^*} \frac{q_{jk^*}}{p_{i^*k^*}} - \sum_{ij} \frac{q_{jk^*}}{\sum_l p_{il} q_{jl}}$$
$$+ (1-\lambda_1)(d^P_{i^*i^*} p_{i^*k^*} - \sum_{t=1}^{n} p_{tk^*} w^P_{ti^*})$$

And then we set them to zero which leads to the following equations:

$$q_{j^*k^*} = \frac{\sum_i z_{ij^*k^*} p_{ik^*}}{\sum_i \frac{p_{ik^*}}{\sum_l p_{il} q_{j^*l}} z_{ij^*k^*} - \lambda_1 (d^Q_{j^*j^*} q_{j^*k^*} - \sum_{t=1}^{m} q_{tk^*} w^Q_{tj^*})} \tag{5}$$

$$p_{i^*k^*} = \frac{\sum_j z_{i^*jk^*} q_{jk^*}}{\sum_j \frac{q_{jk^*}}{\sum_l p_{i^*l} q_{jl}} z_{i^*jk^*} - (1-\lambda_1)(d^P_{i^*i^*} p_{i^*k^*} - \sum_{t=1}^{n} p_{tk^*} w^P_{ti^*})} \tag{6}$$

Since we would like to optimize both $p_{ik}$ and $q_{jk}$, our model is bi-convex that is prone to get trapped in one of the local optimums. One possible solution is to fix one of the unknown parameters, and solve the optimization problem for the other. We use the average alternating projections method [2] with different initial values to provide a set of estimators. For this purpose we first fix $p_{ik}$ and solve the optimization problem for $q_{jk}$, and then fix $q_{jk}$ and solve the problem for $p_{ik}$, and continue until convergence. We assume the model converges when the following holds for Q (and a similar term for P):

$$\frac{1}{m} \sum_{1 \le j \le m} \sum_k (q_{jk} - q_{j*k*})^2 < \epsilon \tag{7}$$

In 7 $\epsilon$ is a small number ($10^{-3}$ in our case). In the two real-world data sets that we examined, Chiron converges in at most 5 steps.

# 5 Experimental Setting

## 5.1 Datasets

We use the MoveLens100K, and Netflix3m1k databases for our experiments. The first data set is gathered by GroupLens Research Project [1] at the University of Minnesota. The last one is provided by Netflix in the Netflix prize [4]. Prea software [10] gathered all these data sets in its toolkit. In each dataset each user rated at least 20 movies from 1 (defined as did not like) to 5 (liked very much). We perform a cross-validation by splitting each of dataset into a training set (80%), a validation set (10%), and a test set (10%), and compare the predicted ratings with actual ratings of the test set. We repeat our experiments 10 times and use the averaged results.

## 5.2 Attack Design

We are only concerned with shilling attacks in which attackers try to manipulate a recommendation system by introducing fake users, and subsequently fake ratings. We only focus on push attacks since they are usually more successful than nuke attacks [8]. The effect of an attack is measured by the deviation in predicted ratings before and after adding the attack profiles. The most common metric for evaluating recommendation systems is Mean Absolute Error(MAE) which is used to measure accuracy in predicting ratings. Two important metrics that are used for evaluation of different shilling attacks are the attack size and the filler size [12]. The attack size is the ratio of added shilling profiles to the original data set. For example, a 10% attack size indicates that the number of shilling profiles added to the system is equal to 10% of the users in the original data set. Another metric that is used for evaluation of different shilling attacks is the filler size. The filler size is the set of items which are voted for in the attacker profile. We target a set of 20 items for the push attack. We repeat each experiment 10 times, and consider the mean value across these 10 times for each item in order to make sure our results are statistically significant. The most effective attack models are derived by reverse engineering the recommendation algorithms to maximize their impact. According to Burke et al. [5] the most common recommendation systems attack methods are random, average, and bandwagon. In a random attack, the assigned ratings made by attackers are around the overall mean rating with standard deviation 1.1. In average attacks, the assigned ratings made by attackers are around the mean rating of every item and standard deviation 1.1. Bandwagon attack is similar to the random attack, and some popular items are rated with the maximum rate. Random and Bandwagon attacks do not require much knowledge about the set of items they are attacking. On the other hand, average attacks require more information and are shown to be near optimal in impact [11]. They are also very challenging to detect [17], and are stronger than random or bandwagon attacks [12]. Therefore, we are only concerned with the average attacks.

### 5.3 Experiments

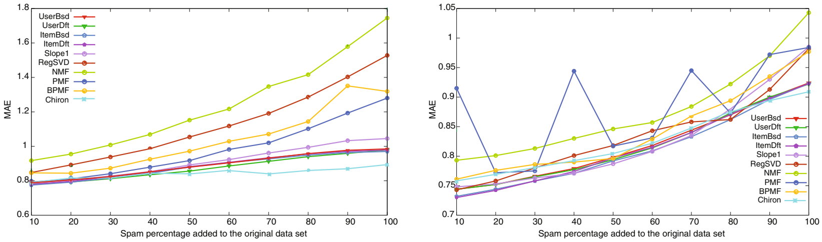
We use Prea [9] to compare our proposed model with different recommendation systems. The different algorithms we select to compare with Chiron fall into these two categories: memory-based neighborhood methods (User-based Collaborative Filtering, User-based Collaborative Filtering (Default Voting), Item-based Collaborative Filtering, Item-based Collaborative Filtering (Inverse User Frequency), and Slope One), and matrix factorization methods (Regularized SVD, Non-negative Matrix Factorization (NMF), Probabilistic Matrix Factorization (PMF), and Bayesian Probabilistic Matrix Factorization (BPMF)).

## 6 Results and Discussion

The results of running different recommendation systems on the Netflix3M1K, and MovieLens100K, are shown in Table 1. We compared the prediction accuracy growth among different collaborative system methods after an attack size 100% on the first two data sets. A 100% attack is a shill, in which the number of fake users is equal to the number of genuine users. The mission of attackers is to promote a certain list of items (20 items in our experiment) and give an average rating to another set of random items to remain undetected. As expected, Chiron

**Table 1.** Changes in prediction accuracy after an attack size of 100% in the data sets. The statistically significant result is shown in bold in each column.

| Method | Data set | | | | | |
|---|---|---|---|---|---|---|
| | Netflix3M1K | | | MovieLens100k | | |
| | MAE | | | | | |
| | Before | After | Growth | Before | After | Growth |
| User-based CF | 0.772 | 0.985 | 27% | 0.734 | 0.924 | 25% |
| User-based DF | 0.765 | 0.979 | 27% | 0.735 | 0.922 | 25% |
| Item-based CF | **0.756** | 0.970 | 28% | 0.722 | 0.923 | 27% |
| Item-based DF | 0.760 | 0.979 | 28% | **0.718** | 0.923 | 28% |
| Slope One | 0.775 | 1.045 | 34% | 0.744 | 0.985 | 32% |
| Regular SVD | 0.819 | 1.528 | 86% | 0.729 | 0.982 | 34% |
| Non negative MF | 0.868 | 1.745 | 101% | 0.780 | 1.043 | 33% |
| Probabilistic MF | 0.786 | 1.280 | 62% | 0.775 | 0.984 | 26% |
| Bayesian PMF | 0.793 | 1.319 | 66% | 0.745 | 0.977 | 31% |
| Chiron | 0.775 | **0.943** | **21**% | 0.737 | **0.908** | **23**% |



(a) Accuracy changes: Netflix3M1K (b) Accuracy changes: MovieLens100K

**Fig. 2.** Changes in prediction accuracy in two different data sets.

is the most robust recommendation system and has the least amount of change in accuracy. In another similar experiment, we attacked both the Netflix3M1K and MovieLens100K data sets with a 10% attack and gradually increased the attack size until it reached 100%. We increase the size till 100% to demonstrate how different models react to the increase in attack size. We pick 100% because most models could be distinguished from each other at that point. The changes in prediction accuracy are illustrated in Fig. 2a and b. In both of them the performance of Chiron almost remains unchanged with increases in the attack size while the prediction accuracy of other methods drops.

## 7 Conclusion

Various methods exist for protecting recommendation systems against attacks by malicious users. More research has been done in the detection of attackers rather than proposing a robust recommendation system. Besides, the level of spam in real world data is often high, and simple spam detection methods are often reverse engineered. We proposed a new model which is empirically robust to attacks, explored its characteristics, and provided compelling evidence of its robustness. Chiron can not easily be manipulated since it relies on modeling and other users ratings. Furthermore, we have illustrated its improved robustness in comparison with other state-of-the-art methods, and concluded that Chiron does not lose accuracy in the presence of shill attacks.

## References

1. MovieLens. http://grouplens.org/datasets/movielens/
2. Agarwal, A., Anandkumar, A., Jain, P., Netrapalli, P., Tandon, R.: Learning sparsely used overcomplete dictionaries via alternating minimization. CoRR abs/1310.7991 (2013). http://arxiv.org/abs/1310.7991
3. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J. Mach. Learn. Res. **7**, 2399–2434 (2006). http://dl.acm.org/citation.cfm?id=1248547.1248632

4. Bennett, J., Lanning, S., Netflix, N.: The netflix prize. In: KDD Cup and Workshop in Conjunction with KDD (2007)
5. Burke, R., Mobasher, B., Williams, C., Bhaumik, R.: Classification features for attack detection in collaborative recommender systems. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2006 (2006)
6. Cai, D., He, X., Han, J., Huang, T.S.: Graph regularized non-negative matrix factorization for data representation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(8), 1548–1560 (2011)
7. Cosley, D., Lam, S.K., Albert, I., Konstan, J.A., Riedl, J.: Is seeing believing? how recommender system interfaces affect users' opinions. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2003, NY, USA, pp. 585–592. ACM, New York (2003)
8. Lam, S.K., Riedl, J.: Shilling recommender systems for fun and profit. In: Proceedings of the 13th International Conference on World Wide Web, WWW 2004, NY, USA, pp. 393–402. ACM, New York (2004)
9. Lee, J., Sun, M., Lebanon, G.: A comparative study of collaborative filtering algorithms. CoRR abs/1205.3193 (2012). http://arxiv.org/abs/1205.3193
10. Lee, J., Sun, M., Lebanon, G.: Prea: personalized recommendation algorithms toolkit. J. Mach. Learn. Res. **13**(1), 2699–2703 (2012)
11. Mehta, B.: Unsupervised shilling detection for collaborative filtering. In: Proceedings of the 22nd National Conference on Artificial Intelligence, AAAI 2007, vol. 2, pp. 1402–1407. AAAI Press (2007)
12. Mehta, B., Hofmann, T.: A survey of attack-resistant collaborative filtering algorithms. IEEE Data Eng. Bull. **31**(2), 14–22 (2008)
13. Mehta, B., Nejdl, W.: Attack resistant collaborative filtering. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, NY, USA, pp. 75–82. ACM, New York (2008)
14. Mobasher, B., Burke, R., Bhaumik, R., Williams, C.: Toward trustworthy recommender systems: an analysis of attack models and algorithm robustness. ACM Trans. Internet Technol. **7**(4), 23 (2007)
15. O'Mahony, M., Hurley, N., Silvestre, G.: Promoting recommendations: an attack on collaborative filtering. In: Database and Expert Systems Applications. Lecture Notes in Computer Science, vol. 2453, pp. 494–503. Springer, Heidelberg (2002)
16. Resnick, P., Sami, R.: Manipulation-resistant recommender systems through influence limits. SIGecom Exchanges **7**(3) (2008). http://dblp.uni-trier.de/db/journals/sigecom/sigecom7.html#ResnickS08
17. Zhang, S., Ouyang, Y., Ford, J., Makedon, F.: Analysis of a low-dimensional linear model under recommendation attacks. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2006 (2006)
18. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Thrun, S., Saul, L.K., Schölkopf, B. (eds.) Advances in Neural Information Processing Systems 16, pp. 321–328. MIT Press (2004). http://papers.nips.cc/paper/2506-learning-with-local-and-global-consistency.pdf

# A 2D/3D Convolutional Neural Network for Brain White Matter Lesion Detection in Multimodal MRI

Leire Roa-Barco[1], Oscar Serradilla-Casado[1], Mikel de Velasco-Vázquez[1],
Asier López-Zorrilla[1], Manuel Graña[1,2(✉)], Darya Chyzhyk[3],
and Catherine Price[3]

[1] Computational Intelligence Group, UPV/EHU, San Sebastian, Spain
[2] ACPySS, San Sebastian, Spain
manuel.grana@ehu.es
[3] McKnight Brain Institute, University of Florida, Gainesville, USA

**Abstract.** White matter hyperintensities (WHM) are characteristics of various brain diseases, so automated detection tools have a broad clinical spectrum. Deep learning architectures have been recently very successful for the segmentation of brain lesions, such as ictus or tumour lesions. We propose a Convolutional Neural Network composed of four parallel data paths whose input is a mixture of 2D/3D windows extracted from multimodal magnetic resonance imaging of the brain. The architecture is lighter than others proposed in the literature for lesion detection so its training is faster. We carry out computational experiments on a dataset of multimodal imaging from 18 subjects, achieving competitive results with state of the art approaches.

## 1 Introduction

White matter hyperintensities (WMH) can be caused by a variety of factors including ischemia, micro-hemorrhages, gliosis, damage to small blood vessel walls. Many patients showing WMH are idiopathic, however WMH have a strong relationship with age, arterial hypertension, demographic parameters such as gender, and some disease, such as diabetes, and biomarkers such as cholesterol [15]. It has been found associated with progressive cognitive impairment [5]. WMH are small size lesions compared with tumours and stroke lesions, lacking their structure of necrotic and inflamed tissues. They are mostly periventricular lesions, which primarily appear at the top of the horns of the lateral ventricles progressing around the ventricles. They may also appear as subcortical lesions [9,25]. Several magnetic resonance image (MRI) modalities may be used used for WMH detection and segmentation. They appear as hypointense in T1-weighted and as hyperintense in T2-weighted images [23]. The best modality is the fluid attenuated inversion recovery (FLAIR) imaging, where the lesions appear as hyperintense and with greater contrast, allowing to differentiate between

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_39

periventricular and subcortical lesions. Recent studies [17,21] also consider diffusion tensor imaging (DTI), specifically the scalar coefficients such as fractional anisotropy (FA), radial diffusivity (RD), and mean diffusivity (MD), which give the information about privileged directions of water diffusion, so they are sensitive to microstructural changes in white matter. In the last years, the interest in brain lesion image segmentation has increased, for example, public challenges have been carried out BRATS http://braintumorsegmentation.org/ and ISLES http://www.isles-challenge.org/ to advance the field. Most research on small lesion detection has been carried out for multiple sclerosis (MS) patients. Early approaches consisted in semiautomatic labellings in structural images [16] and FLAIR [11]. Early multimodal approaches applied voxelwise fuzzy expert systems [1] and Markov random fields (MRF) [20]. Machine learning supervised approaches have been also applied, such as Random Forest [8] and MRF regularized versions [22]. Unsupervised approaches have made advantage of the brain symmetry for big lesion detection [6]. Recently, Deep Learning approaches report great success in the segmentation of brain tumours, specifically Convolutional Neural Networks (CNN) [18,26] which is the approach that we are following in our own proposal. Processing 3D medical images by the CNNs can be done in 3 ways: (a) Considering each 2D slice of the 3D volume in some direction (sagittal, coronal or axial) as an independent input image that is feed to the CNN [18,26]. (b) Considering 3D windows of the volumetric image as input. (c) Considering hybrid 2D/3D inputs, i.e. feeding 2D slices and 3D windows of the volumetric image. This decision carries some implications in the CNN design, because a 3D input forces that hidden layers resulting of the filters have 3D structure [3,24]. This additional structural complexity has been found cumbersome to deal with large datasets, because the number of operations scalate cubically instead of quadratically. So the intended advantage of preserving 3D spatial relation information, is countered by convergence issues and computational time, so that the 3D windows are small, loosing information of long distance spatial relations. Finally, the use of hybrid 2D and 3D input information [2,7] allows a good balance between the preservation of 3D spatial relations and the long distance relations that can be analysed in 2D data. In our architecture, we have used an hybrid 2D/3D NN where we use a small 3D cube and three different 2D windows, one for each of the 3 dimensional axis. The paper contents is as follows: first we present the dataset used for the experiments. Secondly, we discuss our architecture and the others used for comparison. Then we present our experimental results and, finally, some conclusions and future work.

## 2 Materials

The experimental evaluation of the proposed CNN architecture has been carried out in a set of 18 subjects MRI images corresponding to a previous study [19] where WHM was performed manually, thus providing the ground truth segmentation for the present work as 3D lesion masks. Each subject image includes a 3D T1-weighted, FLAIR image, and diffusion weighted images from which DTI images, and subsequent FA coefficients, were computed using FSL software. T1-weighted volumes have been registered to 1 mm MNI template. The FLAIR and FA images have been corregistered to the MNI space by affine registration to normalized T1-weighted images. The lesion masks are also corregistered to MNI space. All the image intensities are normalized to the [0,1] interval.

## 3 Tested CNN Architectures

Throughout the last years, Convolutional Neural Networks (CNNs) [13] have achieved excellent performance in many computer vision tasks. Several advances have solved convergence issues, and the advent of easy to exploit powerful Graphics Processing Units (GPUs) has speed up the training times by several orders of magnitude [4]. A CNN is a shared-weight neural network: all the neurons in a hidden layer share the same weights and bias. In fact, each layer implements a linear convolution filter whose kernel is learnt by gradient descent. Therefore, the output of the successive layers is a series of filtered/subsampled images which are interpreted as progressively higher level abstract features. Most CNN are applied to 2D signals, i.e. images, however in the medical image domain they are increasingly applied to 3D signals, i.e. volumetric imaging information. Specifically, two recent instances of CNNs have been successfully applied to brain lesion segmentation [10,12] achieving remarkable success in the BraTs competition. Another recent segmentation example using a 2D/3D input data is [7], where authors trained two separate CNNs for each input dimensionality, performing a combination of their outputs by averaging.

### 3.1 Our Proposal: MPCNN

Our proposal is a Mixed Parallel CNN (MPCNN), which takes four inputs: three orthogonal big 2D windows on 3D image slices (one per spatial dimension) centered at the same voxel of the brain, and a 3D window, a cube whose sides are smaller than that of the 2D windows. Therefore, 2D data carry farther away spatial relations, while the 3D window carries 3D spatial relations. The MPCNN architecture consists of four parallel CNN, three dedicated to process the 2D window, and the fourth processing the 3D window. Furthermore, we use multimodal MRI data, specifically T1, FLAIR and FA volumes, so that each voxel is in fact a three dimensional vector, much like an RGB image. In this sense, independent CNN filters at each layer are learnt for each image modality. The output is a couple of binary units that provide an estimation of the probability that the central pixel of the 2D and 3D windows is a WMH lesion voxel. Figure 1 shows a diagram of the MPCNN architecture. Each parallel subnetwork is a CNN, composed of a sequence of convolutional layers and max-pooling layers which reduce the reduce the dimensionality of the feature space after each convolution. In the version of the network tested in this paper the dimension of each of the input 2D windows is $35 \times 35$, whereas the dimension of the input 3D cube is $11 \times 11 \times 11$. The activation function used to compute the output of each neuron of the CNN is the Rectified Linear Unit (ReLU) [13,14] due to both its efficient computation and the fact that it solves the vanishing gradient problem. The architectures of the three 2D CNNs are identical, they are composed of three convolutions with kernels of size 3 × 3. The number of convolutions increases along the layers, increasing the number of features accordingly. Moreover, a dimensionality reduction max-pooling layer with pool size of 2 × 2 is applied to the output of the second and third layers. The dimensions of the output of each layer are shown in Fig. 1. Thus, each 2D subnetwork's output layer has $6 \times 6 \times 55 = 1980$ neurons. The 3D CNN is composed only of two 3D convolutions (with kernel size of $3 \times 3 \times 3$), and one 3D max-pooling (with pool size of $2 \times 2 \times 2$) after the second convolution. Finally, all the subnetworks are merged (this results in $1980 \times 3 + 1485 = 7425$ nodes) and fully connected to the next layer, composed of 128 neurons. Finally, these 128 outputs are used to compute the final output of the network via the Softmax function. Hence, the two outputs will always be bounded between 0 and 1, and they will sum 1. This facilitates a probabilistic interpretation of the network output as a probability of lesion at the central voxel.

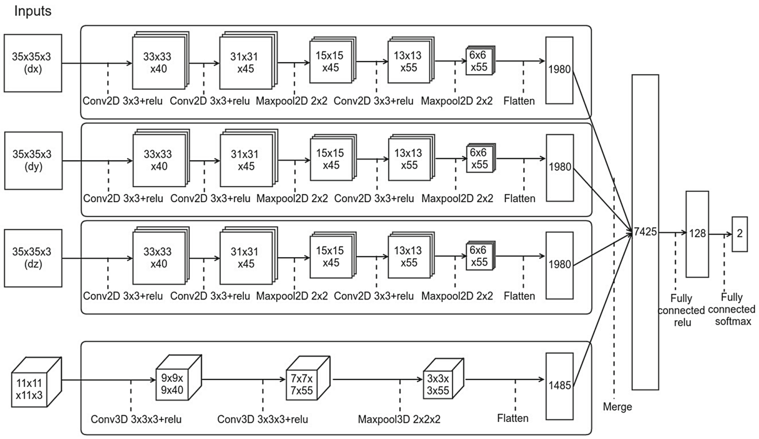


**Fig. 1.** The structure of the proposed for WMH lesion detection

### 3.2 ICCNN

For comparison, we have implemented a version of the Input Cascade CNN (ICCNN) architecture [10]. This network has two inputs: one for global context, and a smaller one for specific context. A convolution to the global context input is concatenated with the small input. Then, this data is divided into two parallel networks, each one analysing local and global features, by applying smaller and bigger kernels respectively. These networks are merged applying a final convolution, which ends up in a softmax layer. In our implementation of the network we have reduced the dimension of the last layer to two neurons, which indicate whether the input represents a damaged voxel or not, and we have changed the training process, which has been done in one step with unbalanced data (10 negative cases per each positive damaged voxel). Moreover, we have changed the activation function to the ReLU, removed dropouts and used binary crossentropy loss function for training. The main difference relative to MPCNN is that ICCNN only uses 2D slices as input.

### 3.3 DeepMedic

The other architecture tested for comparison is the DeepMedic [12], whose architecture has two main components; a 3D CNN and a fully connected 3D Conditional Random Field (CRF), which performs a postprocessing of the CNN output removing false positives. The CNN consists of four layers with $5 \times 5 \times 5$ kernels for feature extraction, and the classification layer is implemented as a convolutional layer with kernel of size $1 \times 1 \times 1$, allowing efficient dense-inference. The 3D CNN network has two pathways; one processes local information and the other processes larger contextual information, hence carrying out multi-scale processing of the data. Moreover, BN (Batch Normalization) is also applied to all the hidden layers, so that all Feature Maps obtained after each layer are normalized, preserving the signal, and avoiding sourious weight convergence. After that, there are two hidden layers for combining the multi-scale parallel pathways. The full network is trained patch-by-patch and the size of the batches is selected automatically according to the neighborhood of the voxel in the input. The batches are built by extracting segments from the training images with 50% probability of being centered on a foreground or background voxel, which corrects the class-imbalance. The DeepMedic network training implementation downloaded from github was originally prepared for the ISLES and BraTS challenges, reporting state-of-the-art results on both performance on brain tumor and stroke lesion. However, since in our problem we only have 2 outputs not 5 as in the segmentation problems, in order to work with this network the last layer output has been reduced from 5 to 2 outputs.

**Table 1.** Results of the networks using holdout: TPR (True Positive Rate) and FPR (False Positive Rate)

| #id | MPCNN | | ICCNN | | DeepMedic | |
|---|---|---|---|---|---|---|
| | TPR | FPR | TPR | FPR | TPR | FPR |
| #7 | 0.572 | 0.037 | 0.106 | 0.018 | 0.599 | 0.019 |
| #15 | 0.503 | 0.024 | 0.259 | 0.063 | 0.650 | 0.013 |
| #18 | 0.622 | 0.012 | 0.490 | 0.065 | 0.613 | 0.017 |
| #21 | 0.245 | 0.011 | 0.280 | 0.034 | 0.463 | 0.012 |

## 4 Results

The MPCNN and ICCNN architectures have been implemented in Python using Keras with Tensorflow as backend. The DeepMedic implementation has been downloaded from github (https://github.com/Kamnitsask/deepmedic). The training and validation scripts have been executed in a desktop computer with RAM of 16 GB, and GPU NVIDIA GTX 1070 which has been used to speed up training. For validation, we apply holdout over the 18 available subject datasets: 14 have been used for training, and 4 for testing. To carry out the training in a limited reasonable time, we have subsampled the brain images as shown in Fig. 2 to obtain the training dataset. The brain image is decomposed in regular non-overlapping windows and a random voxel is picked from this window as

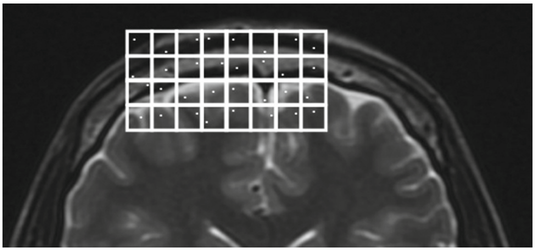


**Fig. 2.** Brain image subsampling to obtain the training dataset

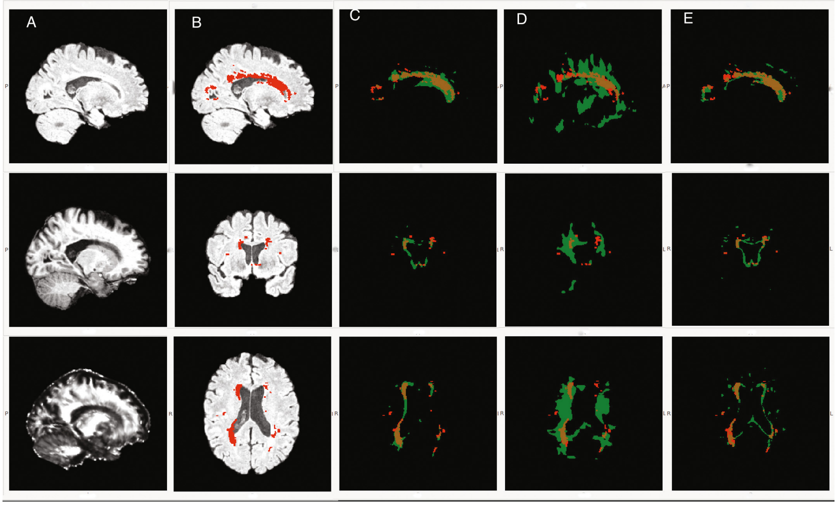


**Fig. 3.** Data and results of subject #18. A - Sample sagittal slides of T1, FA and FLAIR volumes. B - WMH ground truth lesion manually labeled overlaid on FLAIR slices C,D,E - prediction (green) and ground truth lesion (red), C for MPCNN, D for ICCNN, and E for DeepMedic

the center for the 2D/3D windows that conform the inputs. This process ensures a rather regular sampling interval and that the whole brain volume is sampled. Testing is carried out evaluating all the brain voxels in the test datasets. The problem is naturally imbalanced, i.e. there are many more healthy than lesion voxels, therefore we need to respect this imbalance in the training dataset. After some experimentation with a small CNN carrying crossvalidation on a reduced dataset, we set the imbalance ratio to 10 in the training dataset. In other words, we ensure that there is a ratio 10:1 of healthy to lesion voxels. We report True Positive Ratio (TPR) and False Positive Ratio (FPR) values, measuring how well the lesion is detected and the false alarms raised. Accuracy results for each test images are presented in Table 1. Overall, DeepMedic neural network has the best and most stable results, while ICCNN performs poorly. Our proposed MPCNN is faster to train than DeepMedic (a ratio 7:1) and has comparable results in two subjects (#7, #18), and slightly worse in another (#15). If we consider the maximum TPR achieved (0.65), seems that the architectures need to be improved, and that the success in tumour segmentation does not ensure success in WMH lesion detection. Figure 3 presents visual results of the experiment. From left to right, the first column shows images of the three modalities as an illustration of the dataset. The second column shows the lesion detected manually in three slices of brain #18 overlaid on the FLAIR image. Next columns illustrate the detection by MPCNN, ICCNN, and DeepMedic. It can be appreciated that all of them leave some lesion clusters undetected, and overestimate others. DeepMedic seems to create spurious lesion detection clusters, while our proposal MPCNN false alarms are more of the kind of cluster extensions, or conections between clusters. So, some qualitative differences of the response of the architectures can be appreciated which deserver further analysis and experimentation.

## 5 Conclusions and Future Work

We have proposed and tested a new 2D/3D CNN architecture for the detection of WMH lesions, which are smaller than other brain lesions (tumours and stroke lesions), lacking the necrotic and inflammation structures. We compare results with two other architectures published in the literature achieving competitive results. Qualitative assessment of the results, shows some advantage of our approach, which is closer to the manual segmentation in the sense that follows more closely the delineated voxel clusters, and creates less spurious detection clusters. The combination of 2D and 3D input windows allows to process the long distance spatial relations, while reducing the computational burden. Ongoing work improves the validation process computing a more complete cross-validation procedure, and more datasets will be included in the experiment. Our proposal may be also subject to changes in kernel parameters and other features of the CNN. Notice that no postprocessing to remove false alarms is done, contrary to DeepMedic, so additional work in postprocessing MPCNN results may provide enhanced results. In order to go ahead in this research area, we made the code available in github so that everyone can contribute to it.

## References

1. Admiraal-Behloul, F., van den Heuvel, D., Olofsen, H., van Osch, M., van der Grond, J., van Buchem, M., Reiber, J.: Fully automatic segmentation of white matter hyperintensities in MR images of the elderly. NeuroImage **28**(3), 607–617 (2005). http://www.sciencedirect.com/science/article/pii/S105381190500460X
2. de Brébisson, A., Montana, G.: Deep neural networks for anatomical brain segmentation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 20–28, June 2015
3. Brosch, T., Yoo, Y., Tang, L.Y.W., Li, D.K.B., Traboulsee, A., Tam, R.: Deep convolutional encoder networks for multiple sclerosis lesion segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015, Part III. LNCS, vol. 9351, pp. 3–11. Springer, Cham (2015). doi:10.1007/978-3-319-24574-4_1
4. Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: Advances in Neural Information Processing Systems, pp. 2843–2851 (2012)
5. Debette, S., Markus, H.S.: The clinical importance of white matter hyperintensities on brain magnetic resonance imaging: systematic review and meta-analysis. BMJ **341**, c3666 (2010). http://www.bmj.com/content/341/bmj.c3666
6. Erihov, M., Alpert, S., Kisilev, P., Hashoul, S.: A cross saliency approach to asymmetry-based tumor detection. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds.) MICCAI 2015, Part III. LNCS, vol. 9351, pp. 636–643. Springer, Cham (2015). doi:10.1007/978-3-319-24574-4_76
7. Gao, X.W., Hui, R., Tian, Z.: Classification of ct brain images based on deep learning networks. Comput. Methods Programs Biomed. **138**, 49–56 (2017)
8. Geremia, E., Clatz, O., Menze, B.H., Konukoglu, E., Criminisi, A., Ayache, N.: Spatial decision forests for MS lesion segmentation in multichannel magnetic resonance images. NeuroImage **57**(2), 378–390 (2011). http://www.sciencedirect.com/science/article/pii/S1053811911003740
9. Grueter, B.E.: S.U.G.: age-related cerebral white matter disease (leukoaraiosis): a review. Postgrad. Med. J. **88**, 79–87 (2012)
10. Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.M., Larochelle, H.: Brain tumor segmentation with deep neural networks. Med. Image Anal. **35**, 18–31 (2017). http://www.sciencedirect.com/science/article/pii/S1361841516300330
11. Iorio, M., Spalletta, G., Chiapponi, C., Luccichenti, G., Cacciari, C., Orfei, M.D., Caltagirone, C., Piras, F.: White matter hyperintensities segmentation: a new semi-automated method. Front. Aging Neurosci. **5**(76) (2013). http://www.frontiersin.org/aging_neuroscience/10.3389/fnagi.2013.00076/abstract
12. Kamnitsas, K., Ledig, C., Newcombe, V.F., Simpson, J.P., Kane, A.D., Menon, D.K., Rueckert, D., Glocker, B.: Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Med. Image Anal. **36**, 61–78 (2017)
13. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
14. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
15. Murray, A., Staff, R., Shenkin, S., Deary, I., Starr, J., Whalley, L.: Brain white matter hyperintensities: relative importance of vascular risk factors in nondemented elderly people. Radiology **237**, 251–257 (2005)

16. Payne, M.E., et al.: Development of a semi-automated method for quantification of MRI gray and white matter lesions in geriatric subjects. Psychiatry Res. Neuroimaging **115**(1), 63–77 (2002)
17. Pelletier, A., Periot, O., Dilharreguy, B., Hiba, B., Bordessoules, M., Chanraud, S., Pérés, K., Amieva, H., Dartigues, J., Allard, M., Catheline, G.: Age-related modifications of diffusion tensor imaging parameters and white matter hyperintensities as inter-dependent processes. Front. Aging Neurosci. **7**(255) (2016). http://www.frontiersin.org/aging_neuroscience/10.3389/fnagi.2015.00255/abstract
18. Pereira, S., Pinto, A., Alves, V., Silva, C.A.: Deep convolutional neural networks for the segmentation of gliomas in multi-sequence MRI. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Handels, H. (eds.) BrainLes 2015. LNCS, vol. 9556, pp. 131–143. Springer, Cham (2016). doi:10.1007/978-3-319-30858-6_12
19. Price, C., Mitchell, S., Brumback, B., Tanner, J., Lamar, I.S.M., Giovannetti, T., Heilman, K., Libon, D.: MRI-leukoaraiosis thresholds and the phenotypic expression of dementia. Neurology **79**(8), 734–740 (2012)
20. Schwarz, C., Fletcher, E., DeCarli, C., Carmichael, O.: Fully-automated white matter hyperintensity detection with anatomical prior knowledge and without FLAIR. Inf. Process. Med. Imaging **21**, 239–251 (2009). Proceedings of the Conference
21. Tuladhar, A.M., van Dijk, E., Zwiers, M.P., van Norden, A.G., de Laat, K.F., Shumskaya, E., Norris, D.G., de Leeuw, F.E.: Structural network connectivity and cognition in cerebral small vessel disease. Hum. Brain Mapp. **37**(1), 300–310 (2016). http://dx.doi.org/10.1002/hbm.23032
22. Tustison, N., Wintermark, M., Durst, C., Avants, B.: Ants and árboles. In: MICCAI BraTS Workshop. Miccai Society, Nagoya (2013)
23. Uchiyama, Y., Kunieda, T., Hara, T., Fujita, H., Ando, H., Yamakawa, H., Asano, T., Kato, H., Iwama, T., Kanematsu, M., Hoshi, H.: Automatic segmentation of different-sized leukoaraiosis regions in brain MR images. In: Proceedings of SPIE, vol. 6915, pp. 69151S-1–69151S-8 (2008). http://dx.doi.org/10.1117/12.770045
24. Urban, G., Bendszus, M., Hamprecht, F., Kleesiek, J.: Multi-modal brain tumor segmentation using deep convolutional neural networks. In: MICCAI BraTS (Brain Tumor Segmentation) Challenge. Proceedings, Winning Contribution, pp. 31–35 (2014)
25. Yoshita, M., Fletcher, E., Harvey, D., Ortega, M., Martinez, O., Mungas, D.M., Reed, B.R., DeCarli, C.S.: Extent and distribution of white matter hyperintensities in normal aging, MCI, and AD. Neurology **67**(12), 2192–2198 (2006). http://www.neurology.org/content/67/12/2192.abstract
26. Zikic, D., Ioannou, Y., Brown, M., Criminisi, A.: Segmentation of brain tumor tissues with convolutional neural networks. In: Proceedings MICCAI-BRATS, pp. 36–39 (2014)

# Automated Segmentation of Visceral Adiposity in MRI in Obese Children

Manuel Graña[1,2](✉), Oier Echaniz[1,2], Beatriz Rodriguez-Vigil[2,3], and Idoia Labayen[2,3]

[1] Computational Intelligence Group, University of the Basque Country (UPV/EHU), San Sebastián, Spain
manuel.grana@ehu.es

[2] Department of Magnetic Resonance Imaging, Osatek, University Hospital of Alava (HUA), Vitoria-Gasteiz, Spain

[3] Nutrition, Exercise and Health Research Group, ELIKOS Group, University of the Basque Country, UPV/EHU, Vitoria-Gasteiz, Spain

**Abstract.** Children obesity is a growing concern in the healthcare system, because a life of dependence and cronic health problems results in the adult phase of life. Non-alcoholic liver fat and visceral adiposity are two biomarkers of the health status of the child. Some studies try to measure the impact of exercise and improved habits in the reduction of these biomarkers. The studies use a fat enhancing magnetic resonance imaging sequence, but visceral fat is difficult to segment manually. In this paper we describe the automated process that we have devised, providing some visual results. The process has three main phases: (a) image inhomogeneity correction, (b) removal of spurious features such as arms and navel, (c) identification of the peripheral and visceral volumes, (d) localization of the intervertebral disks, which need to be removed. As there is no manual tracing to use as gold standard, we can not provide quantitative performance measures.

## 1 Introduction

Overweight and childhood obesity in developed countries has become epidemic and constitute a huge problem in the public health system [8]. Children with obesity are 5 times more likely to develop insulin resistance and type 2 diabetes mellitus than non-obese ones. Moreover, the majority of children with overweight has at least one cardiovascular (CV) risk factor [6]. Although most CV problems commonly associated with obesity are manifested in adulthood, its origins appear in early stages of life [5]. Thus, CV risk factors measured in childhood predict the development of coronary lesions in adulthood, suggesting that some of these lesions causing CV risk factors may already be occurring in childhood [4]. The accumulation of liver fat is closely linked to visceral adipose tissue (VAT) and insulin resistance. Since exercise improves insulin sensitivity and is able to

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_40

reduce VAT in overweight children, a clinical trial [1,7] has been proposed aiming to measure the effects of controlled exercise sessions in several biomarkers, among then the volume of VAT. The study protocol has been approved by the Ethic Committee of Clinical Investigation of Euskadi (PI2014045). The study hypothesis is that exercise of moderate to high intensity (between ventilatory thresholds) will reduce liver fat, VAT and improve body composition and cardiovascular health in overweight children. To test this hypothesis a comparison between the quantity of adipose tissue before and after the exercise periods is needed. Since manual segmentation is hard and time consuming task, and given the high volume of data that will be generated by the project, an automatic segmentation method is required. This method should provide a quantification of the total VAT and Subcutaneous Adipose Tissue (SAT) volume in order to compare both sequences. The main objective of this work was to develop an automated method to distinguish different type of adipose tissue in a large number of subjects, before and after they are in treatment, so that we can improve our understanding of the relation between the treatments and the presence of adipose tissue. Image based volume quantification will allow to test the study hypothesis. For the method to be successful the result of the newly developed segmentation system have to be accurate, reproducible and, above all, comparable to manual segmentation, which we will demonstrate with visual data presented in the following sections. Section 2 describes the methods we have developed of automatic segmentation. Section 3 provides some visual results on different datasets, highlighting the issues that are solved robustly by the proposed algorithm. Section 4 gives some conclusions and future work to be done.

## 2 Materials and Methods

### 2.1 The Fat Signal Image

The MRI acquisition for this study has been conducted in a Magnetom Avanto equipment, Siemens Healthcare, 1.5 Tesla, of 33mt m maximum gradient amplitude, minimum rise time of 264 µs, high sink rate of 125 T/m/s, version syngo MR B17, Numaris/4 software. The equipment is located at the magnetic resonance unit of Osatek in Hospital Santiago in Vitoria (Álava University Hospital). The sequences were performed in supine position, in apnea without intravenous contrast injection, using phased array matrix body antennas and spine matrix. The fat signal (proton density fat fraction) is acquired by a recently developed MRI sequence [9] that has been made available by Siemens in a beta version. The method uses multi-echo 3D gradient echo acquisition. A Dixon decomposition provides initial guesses of the separation between fat and water using two echos. The estimation is refined in a multistep adaptive fitting process. Figure 1 shows a sample volume, where the navel and the arms can be appreciated. The SAT is a solid region, however the image contains strong inhomogeneities, and the VAT is connected in many points to the SAT. The intervertebral discs, which are fat rich can be appreciated in the leftmost cut.

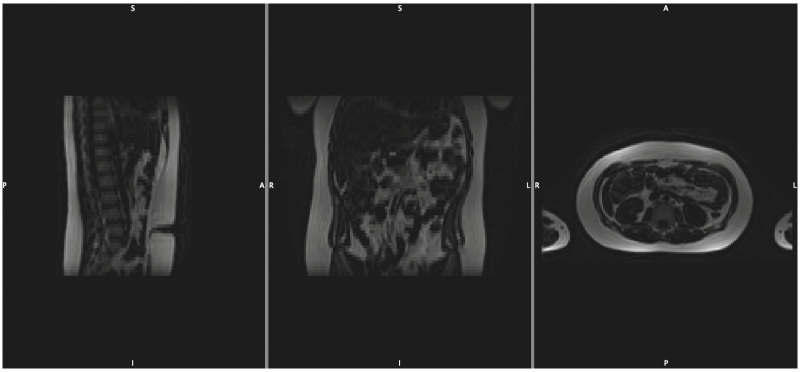


**Fig. 1.** A sample volume of fat signal of a child.

## 2.2 VAT Segmentation Algorithm

The VAT segmentation algorithm has the following main steps:

1. Image intensity normalization by inhomogeneity correction.
2. Removal of anatomical irregularities: the navel and the arms.
3. Identification of the peripheral and visceral regions.
4. Extraction of the VAT applying the mask.
5. Detection of the vertebrae and the intervertebral disks.
6. Computation of the VAT removing intervertebral disks.

### 2.2.1 Image Intensity Normalization

There are several approaches to MRI inhomogeneity correction, based on the homomorphic filtering approach. They need the estimation of the changing illumination field in the volume. Some approaches use parametric models [2,3] where the illumination filed is some linear combination of a family of parametric functions. Here we use a simplistic approach consisting in computing a strong smoothing of the volume with a large Gaussian kernel. The result is a volume where each voxel contains roughly the average value of a large neighborhood, thus it is proportional to the value of the illumination field. We divide the original fat image by this smooth image obtaining intensity values around 1, so that we can safely use a threshold of a value near 1 to produce a mask that selects significant regions of fat. Values near 0.8 of the threshold provide good results.

### 2.2.2 Removal of Anatomical Irregularities

The peripheral subcutaneous fat is rather smooth and free of irregularities, but for two: the navel and the arms, which can be problematic. The navel can effectively produce a connection of the visceral region with the outside air region due to the partial volume effects that effectively make the boundary very light in some cases. We remove the navel from the fat mask computed above by considering the external boundary. The navel introduces big gradients in the external body boundary, which is quite smooth all around the skin, hence we proceed as follows:

- We compute the 3D boundary of the body of one voxel of width by substracting from the fat mask an erosion with an structural element that is a square of $3 \times 3$ filled with ones.
- We compute the derivatives of the boundary considered as a line in each slice.
- When we find two large derivatives coming from the two sides of the frontal boundary, we have found the navel boundaries,
- We link these points with a line in the external boundary. Filling the corrected external boundary, we can fill the navel.

To remove the arms, we start from the middle of the body where the arms are clearly separated from the body moving upwards. We consider at each slide the external boundary of the central fat region in the image, the separated external boundaries correspond to the arms, and can be removed. When we reach the axiles, there is a fusion of the outer arm region with the body, we proceed by assuming that the external boundary of the body remains the same, because the gradients between the arm and the body are not significative enough to be followed in the separation of body and arm.

### 2.2.3 Identification of the VAT and SAT Masks

After navel removal, we can safely compute again the fat mask boundary by subtraction of the erosion with the square unit structural element as before.

Now we are sure that external and internal boundaries are separated, so that we can use the internal boundary to delineate the boundary of the visceral region. In order to break some links between peripheral fat and the visceral fat, we carry out a strong opening of the image. This operation does not shift the internal boundary, but breaks the links effectively. Filling the internal boundary we have the internal mask, and the difference between the whole filled and the internal mask is the mask of the subcutaneous fat.

### 2.2.4 Extraction of the VAT Image

This process is a simple multiplication of the fat signal image by the visceral mask. We retain the fat signal, because its intensity is proportional to the fat in the voxel. Therefore the volumetry of the VAT must be carried out on this segmented fat signal.

### 2.2.5 Detection of the Vertebrae

It can be appreciated in Fig. 1 that the vertebral column has a regular structure, with a regular succession of local maxima and minima. We detect the location of the vertebrae by computing the correlation in the saggital and coronal planes, where this structure is more salient, of a sliding window with a pattern that has a strip of minimum values in the center and maxima in the sides. The dimensions of the pattern are different for the saggital and the coronal planes, to fit the vertebrae dimensions in these planes. The intervertebral disk are detected by the same procedure, but with a pattern that is roughly the inverse of the pattern used for the vertebrae, but not exactly. We profit from the knowledge that the vertebrae signal is zero or very close to that, and that the intervertebral disks have a definite range of values in the fat signal, allowing to remove many confounders that provided false positive correlations. For the final volumetric measurement, the intervertebral disks may be removed.

## 3 Results

In this section we showing visual results of the segmentation algorithm. Figure 2 shows the segmentation masks achieved for the fat signal volume of Fig. 1. The SAT mask is red, and the VAT blue. It can be appreciated that the navel is correctly segmented, and that the arms are separated from the SAT mask.

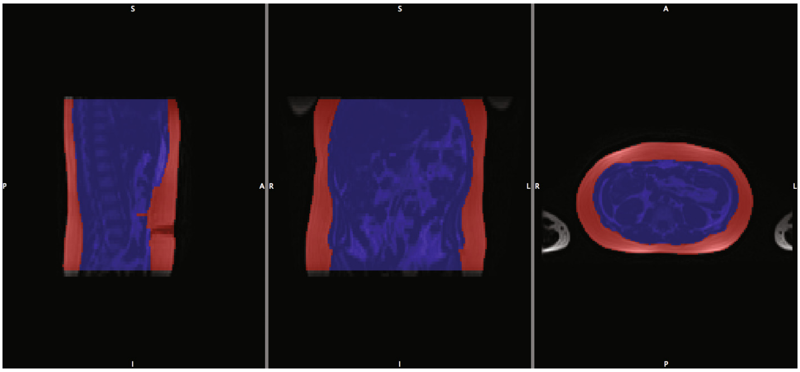


**Fig. 2.** Segmentation masks of the peripherical subcutaneous fat SAT (red) and the visceral fat VAT (blue)

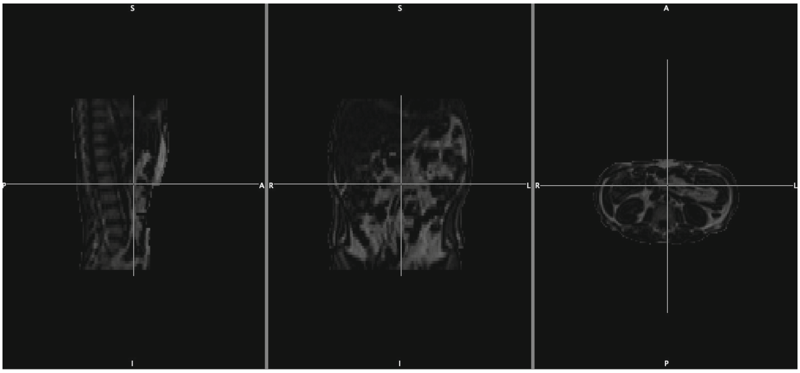


**Fig. 3.** Extracted VAT volume

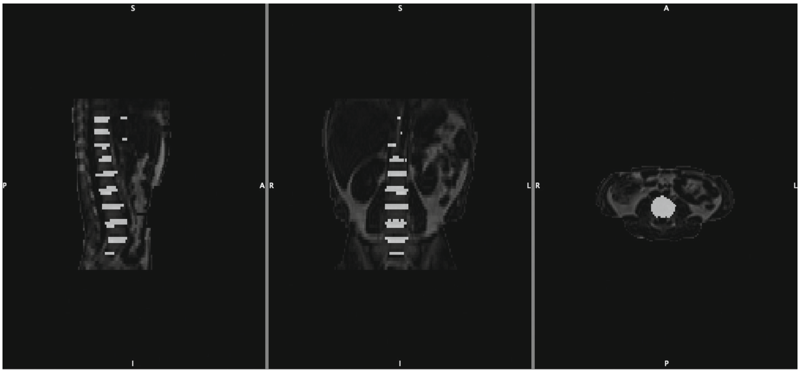


**Fig. 4.** Vertebral discs localized in the VAT volume

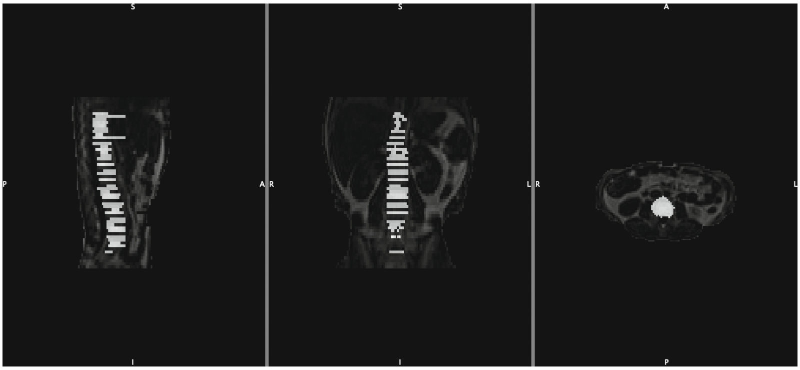


**Fig. 5.** The vertebral and intervertebral disks located

The inner boundary follows closely the visceral space. Figure 3 shows the extracted visceral region, so that the VAT volume can be easily computed. Figure 4 the localization of the vertebral disks. The radiologist may be interesting in specifying the region delimitated by two vertebrae, in order to have some kind of anatomical normalization of the measures, this image facilitates this selection. Finally, Fig. 5 provides the localization of the intervertebral disks as well, which can be removed from the automated fat volume measurement.

## 4 Conclusions

We have developed a robust VAT and SAT automatic segmentation algorithm, that is currently applied to the date produced by a study on the effect of exercise in the VAT volume of obese children, among other biomarkers of obesity which are processed concurrently. The software is operational, but from the point of view of academic validation we need a ground truth for each volume. Obtaining that is rather difficult because it requires an excessive amount of time. Our strategy is to ask the radiologist to point errors in the produced segmentation, thus giving us the actual error without going into the process of entire manual segmentation. These error measures would be used as validation proof in further academic publications.

**Acknowledgments.** This research has been partially funded the Basque Government grant IT874-13 for the GIC research group.

## References

1. Cadenas-Sanchez, C., Mora-Gonzalez, J., Migueles, J.H., Martin-Matillas, M., Gomez-Vida, J., Escolano-Margarit, M.V., Maldonado, J., Enriquez, G.M., Pastor-Villaescusa, B., de Teresa, C., Navarrete, S., Lozano, R.M., de Dios Beas-Jiménez, J., Estevez-Lopez, F., Mena-Molina, A., Heras, M.J., Chillón, P., Campoy, C., Muñoz-Hernández, V., Martínez-Ávila, W.D., Merchan, M.E., Perales, J.C., Gil, Á., Verdejo-García, A., Aguilera, C.M., Ruiz, J.R., Labayen, I., Catena, A., Ortega, F.B.: An exercise-based randomized controlled trial on brain, cognition, physical health and mental health in overweight/obese children (activebrains project): rationale, design and methods. Contemp. Clin. Trials **47**, 315–324 (2016). http://www.sciencedirect.com/science/article/pii/S1551714416300210
2. Fernandez, E., Graña, M., Ruiz-Cabello, J.: On a gradient based evolution strategy for parametric illumination correction. Electron. Lett. **40**(9), 531–532 (2004)
3. Garcia-Sebastián, M., Fernandez, E., Graña, M., Torrealdea, F.J.: A parametric gradient descent MRI intensity inhomogeneity correction algorithm. Pattern Recogn. Lett. **28**(13), 1657–1666 (2007)
4. Labayen, I., Ruiz, J.R., Ortega, F.B., Harro, J., Merenakk, L., Oja, L., Veidebaum, T., Sjostrom, M.: Insulin sensitivity at childhood predicts changes in total and central adiposity over a 6-year period. Int. J. Obes. **35**(10), 1284–1288 (2011)
5. Labayen, I., Ortega, F.B., Sjostrom, M., Ruiz, J.R.: Early life origins of low-grade inflammation and atherosclerosis risk in children and adolescents. J. Pediatr. **155**(5), 673–677 (2009)
6. May, A.L., Kuklina, E.V., Yoon, P.W.: Prevalence of cardiovascular disease risk factors among US adolescents, 1999–2008. Pediatrics **129**(6), 1035–1041 (2012)
7. Medrano, M., Maiz, E., Maldonado-Martín, S., Arenaza, L., Rodríguez-Vigil, B., Ortega, F., Ruiz, J., Larrarte, E., Diez-López, I., Sarasúa-Miranda, A., Tobalina, I., Barrenechea, L., Pérez-Asenjo, J., Kannengiesser, S., Manhães-Savio, A., Echaniz, O., Labayen, I.: The effect of a multidisciplinary intervention program on hepatic adiposity in overweight-obese children: protocol of the EFIGRO study. Contemp. Clin. Trials **45**, Part B, 346–355 (2015). http://www.sciencedirect.com/science/article/pii/S1551714415300926
8. Ogden, C.L., Carroll, M.D., Flegal, K.M.: High body mass index for age among US children and adolescents, 2003–2006. JAMA **299**(20), 2401–2405 (2008)
9. Zhong, X., Nickel, M.D., Kannengiesser, S.A., Dale, B.M., Kiefer, B., Bashir, M.R.: Liver fat quantification using a multi-step adaptive fitting approach with multi-echo GRE imaging. Magn. Reson. Med. **72**(5), 1353–1365 (2014). http://dx.doi.org/10.1002/mrm.25054

# EEG Classification for MI-BCI with Independent Component Analysis

Izabela Rejer(✉) and Paweł Górski

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

**Abstract.** Independent Component Analysis (ICA) is often used in EEG signal processing but only when the matrix of channels is big enough. If the matrix is small, the artifacts usually cannot be extracted as single components and the decision which components should be removed from the components' set is impeded. Therefore, in our opinion, in order to apply ICA successfully for a low-dimensional EEG, the strategy for dealing with the components' set should be reversed.

In the paper we propose a strategy of searching for components correlating with the desired brain activity, instead of looking for artifact-components. Obviously, since the brain activity depends on the task at hand, different tasks would require adaptation of the proposed approach but the overall scheme is independent from the task. In the paper we describe the strategy and illustrate it via the experiment with a simple 2-states Motor Imagery Brain Computer Interface. Our results show that even if we added only one spare channel to the core set of two channels (C3 and C4) essential for hand movement recognition, we obtained 18% increase in the recognition accuracy after applying ICA and our strategy.

**Keywords:** BCI · EEG · Preprocessing · ICA · Independent Component Analysis

## 1 Introduction

A Brain Computer Interface (BCI) is a system that is used for controlling the computer applications or output devices directly by the signals recorded from a human brain. The brain activity can be recorded with many different devices, but because of a high availability, noninvasive recording technique, and low costs, EEG (Electroencephalograph) is usually used to deal with this task. The brain activity, recorded via a matrix of electrodes connected to EEG device has to undergo a series of transformations before it can be used in the control process. First, it has to be denoised with some preprocessing techniques like spatial filters, frequency filters, Independent Component Analysis etc. [1–4]. Next, the preprocessed EEG signals have to be described by a set of features [5,6].

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_41

Finally, the features have to be passed to the classifier that decides which command should be executed as a result of the recorded brain activity [7,8].

Although, all of the mentioned steps have to be carefully designed in order to built a successful BCI system, the preprocessing stage seems to play the major role. Even if we use the most sophisticated classifiers or methods for feature selection or extraction, if the signal quality is not satisfactory, the whole system will have a very low ITR (Information Transfer Rate – the quality measure used for evaluating BCI performance). Hence, a very important part of a BCI designing process is to ensure a high SNR (Signal-to-Noise Ratio) of the EEG signal recorded from the scalp electrodes.

ICA (Independent Component Analysis) is a method often used in EEG signal analysis to enhance signal-to-noise ratio. It is a method that transforms one set of signals into another set of signals. The optimization criterion used in the transformation process is the maximization of independence of the signals from the output set. In other words, ICA algorithms start with a set of mixed signals (observed signals) and try to decompose them into independent components (source signals). When ICA is used in EEG signal analysis, the set of EEG signals recorded from a matrix of electrodes located over different parts of a subject brain constitutes ICA input set and the set of independent components returned by ICA is considered as the set of unknown sources of recorded EEG signals.

A set of independent components returned by ICA reflects all components that contribute in the signals recorded from scalp electrodes. That means that these components can have their origin not only in cortical sources but can be as well artifacts mixing with the true brain activity. Actually, the artifacts usually dominate over the cortical sources in most EEG channels and so there is a high probability that if only there is enough channels, the artifacts will be retrieved as individual independent components. That is why most research applying ICA for EEG signal analysis focuses on detecting specific artifacts in originally recorded signals (ocular [9,10], muscle, electrocardiographic, and power lines artifacts [11–13]).

When the artifacts are discovered among the components returned by ICA, they can be removed from the set of components and the EEG channels can be reconstructed by using only non-artifacts components to enhance the signal-to-noise ratio (SNR). This strategy provides a considerable increase of the signal quality in the case of multichannel ICA. This fact has been confirmed in many research for different numbers of channels, from 16 [7], through 19–20 [10,11] up to 71 [14] and even many more.

The problem is that this strategy works only if there are enough EEG channels to extract artifacts as individual components. If there are only a few channels, the independent components obtained at ICA output do not present individual sources but are still their mixtures. The components should be of course more independent than the raw signals recorded from the scalp, as ICA optimization criterion is to enhance the statistical independency, but they usually will not represent individual sources (nor cortical or external). Hence, much better solution in a low-channel EEG seems to be picking out the components that enhance the brain activity correlated with the task at hand than to remove the components that might be artifacts. The problem is, however, how to choose the right components in an automatic way.

In the paper we would like to introduce a possible solution to the mentioned problem – a method that allows for deciding which components from the set of independent components returned by ICA should be used in the process of classification of the user mental states to one of the predefined motor imagery classes. To demonstrate the possible benefits of the method, the comparison of the classification accuracy obtained with a matrix of EEG signals before and after ICA transformation is presented in the paper. The EEG data was acquired during an on-line session with a simple 2-states Motor Imagery BCI (MI-BCI).

## 2 Methods

### 2.1 Experiment Setup

The experiment was carried out with a male subject, age 32. The subject was right-handed, had normal vision and did not report any mental disorders. The subject had previous experience with BCI, but only with a BCI designed according to SSVEP (Steady State Visually Evoked Potentials) paradigm. It was the first time when he attended the session with MI-BCI. The experiment was conducted according to the Helsinki declaration on proper treatments of human subjects. Written consent was obtained from the subject before the experiment.

The detailed scheme of the experiment was as follows. The subject was placed in a comfortable chair and EEG electrodes were applied on his head. In order to limit the number of artifacts, the participant was instructed to stay relaxed and not move. The start of the experiment was announced by a short sound signal. The main part of the experiment was divided into 200 trials. During each trial an arrow pointing to the left or right was presented to the subject. The subject task was to imagine the wrist rotation of the hand indicated by the arrow (arrow to the left – left hand; arrow to the right – right hand). The arrow directions for the succeeding trials were chosen randomly. There were no breaks between trials. The experiment was divided into 4 session, 50 trials each. The trial length was fixed and was equal to 10 s. There were 3 min breaks between sessions. Hence, the whole experiment was about 45 min long.

EEG data was recorded from six monopolar channels at a sampling frequency of 256 Hz. Eight passive electrodes were used in the experiments. Six of them were attached to the subject's scalp at O1, O2, Pz, C3, Cz, and C4 positions according to the International 10–20 system [15]. The reference and ground electrodes were located at Fpz and the right mastoid, respectively. The impedance of the electrodes was kept below 5 kΩ. The EEG signal was acquired with Discovery 20 amplifier (BrainMaster) and recorded with OpenVibe Software [16].

### 2.2 Data Preprocessing

At the preprocessing step, the EEG data from all 6 channels was filtered with a Butterworth band-pass filter of the 4th order in the band 4–35 Hz. Next, 16 signal sets, containing different combinations of channels were prepared. Two channels, the critical for a hand motor imagery recognition – C3 and C4, were the core of the analysis and so, were present in each set. The core set was called S2 ("2" stands for two inputs, "S" stands for "signal"). The four remaining channels were added to the core set in all possible combinations to create remaining 15 sets. Hence, we created: 4 three-input sets (S3_Cz S3_Pz; S3_O1; S3_O2); 6 four-input sets (S4_Cz_Pz; S4_Cz_O1; S4_Cz_O2; S4_Pz_O1; S4_Pz_O2; S4_O1_O2); 4 five-input sets (S5_Cz_Pz_O1; S5_Cz_Pz_O2; S5_Cz_O1_O2; S5_Pz_O1_O2); and 1 six-input set (S6).

Each data set used in the analysis was 3-dimensional. The size of the first dimension was variable and depended on the number of channels used in the channel combination (from 2 to 6). The size of the two other dimensions were the same for each set – 200 was the size of the trials dimension, and 2560 was the size of time window dimension.

When the sets were ready, all of them (apart from the core set that was left unchanged as the reference set), were submitted one by one to FastICA algorithm. The algorithm was used separately on each trial. As a result 15 signal sets were transformed to 15 sets of independent components returned by FastICA. The size of each component set was the same as the size of the corresponding signal set. The new sets were named according to the same scheme as signal sets, with the prefix "IC" ("IC" stands for "Independent Component").

The combinations of channels were prepared in order to present that the pairing process described in a next subsection works properly regardless of channels added to the core set. Therefore, in further analysis 16 data sets would be used: the core raw signal sets and 15 components sets (i.e. sets transformed with FastICA).

### 2.3 Components Pairing

The next step was the crucial point of our research. Since we knew that the classification should be performed on the basis of channels located over the hand area of the motor cortex (C3, and C4), we had to choose these components from the set of components returned by ICA that revealed the same brain activity as the raw EEG signals recorded from both channels. We tried two approaches to deal with this task in our previous research but none of them provided the satisfactory output. First, we tried to assign the components to channels according to their similarity, measured by the correlation coefficients. Then, we analyzed the demixing matrix, returned by ICA, and matched the components to channels searching for the channels of the highest contribution to each component. We started from the maximal coefficient found in the matrix, paired the independent component from the row where the coefficient was found to the channel from the corresponding column, then we moved to the second maximum coefficient, did the same pairing, and so on until all components were paired to the channels. That second solution seemed to be quite justified because this is just the way how the head maps of ICs are created, however, it did not increase the classification accuracy. We started to analyze this approach deeper and we finally found the proper solution.

The components returned by ICA are the most independent components contributed to the mixed signals. In EEG data, the strongest contribution usually comes from the artifacts. Hence, when we started the pairing process from the ICs that had the highest demixing coefficients, we looked for the channels where the unknown sources (probably artifacts) were the most visible. In this way we first paired the probably-artifact-components with the channels where they contributed the most, and after that we tried to do the pairing between the remaining components and channels. As a result the whole process started to be random.

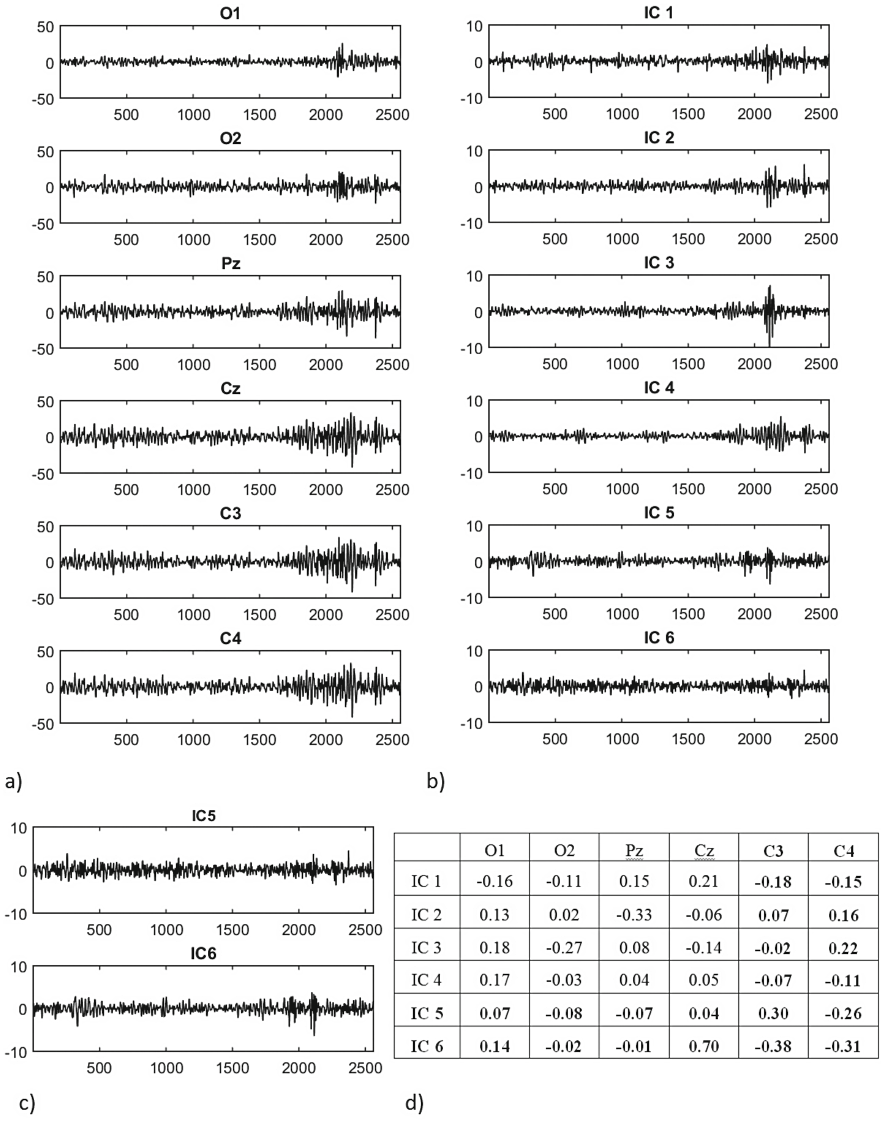
Therefore, we reversed the pairing process. Instead of trying to match components to channels, we matched the channels to the components. It was a simple computational change but it made a big difference. Moreover, we also did not try to match all channels to all components. We knew that we needed only the components that contributed the most to C3 and C4 channels. The remaining channels should not carry useful information for hand movement recognition, and hence we assumed that also the components that were not paired to C3 and C4 channels could be removed from further analysis. We started the procedure from channel C3. We looked for the highest demixing coefficient in a column corresponding to this channel and paired the channel with the independent component from the corresponding row. Next, we did the same pairing for channel C4. As a result we obtained a pair of components, one corresponded to C3, and one corresponded to C4. We did not removed the component matched to C3 channel from the set of components before searching for the component corresponded to C4 channel. Hence, theoretically it was possible that the same component would be returned for both channels. If such situation really occurred, it could lower the classification precision but only for the given trial (it did not occur in the reported survey).

An example of the procedure described above is presented in Fig. 1 via one randomly chosen trial from the six-input channel set. The subfigures presents: Fig. 1a - the set of raw signals from the succeeding channels, Fig. 1b – the set of independent components returned by FastICA, Fig. 1c – the demixing matrix (the columns corresponding to channels C3 and C4 and the rows corresponding to the chosen components are marked), and Fig. 1d – the chosen components.

### 2.4 Feature Extraction, Feature Selection, and Classification

The classification of the motor imagery EEG data was performed with power band features. Twelve frequency bands were used to extract the features: alpha band (8–13 Hz), beta band (13–30 Hz), five sub-bands of alpha band (8–9 Hz; 9–10 Hz; 10–11 Hz; 11–12 Hz; 12–13 Hz), five sub-bands of beta band (13–17 Hz; 17–20 Hz; 20–23 Hz; 23–26 Hz; 26–30 Hz). The features were calculated separately per each second of the recording, hence there were 120 features per each channel, and 240 features per 2-channel/2-component set.

When the paring process ended all sets were reduced to the same size: 2 channels (C3 and C4, or components corresponding to C3 and C4); 200 trials, and 2560 samples. Each of the final data set was individually subjected to feature extraction, feature selection, and classification. Because of the high number of features, the feature selection process was performed. Taking into account the Raudys and



| | O1 | O2 | Pz | Cz | C3 | C4 |
|---|---|---|---|---|---|---|
| IC 1 | -0.16 | -0.11 | 0.15 | 0.21 | -0.18 | -0.15 |
| IC 2 | 0.13 | 0.02 | -0.33 | -0.06 | 0.07 | 0.16 |
| IC 3 | 0.18 | -0.27 | 0.08 | -0.14 | -0.02 | 0.22 |
| IC 4 | 0.17 | -0.03 | 0.04 | 0.05 | -0.07 | -0.11 |
| IC 5 | 0.07 | -0.08 | -0.07 | 0.04 | 0.30 | -0.26 |
| IC 6 | 0.14 | -0.02 | -0.01 | 0.70 | -0.38 | -0.31 |

**Fig. 1.** An example of the approach for pairing channels to components; a – EEG signals from the succeeding channels; b – independent components returned by FastICA; c – the chosen components; d – the demixing matrix.

Jain recommendation that at least 10 times more training examples per class than the features should be gathered to train the classifier correctly [17], we limited the number of features to 8. For feature selection we applied LASSO algorithm, proposed by Tibshirani [18]. The LASSO optimization problem assumes linear dependency between input features and output values and is given as minimization of:

$$\frac{1}{2N}\sum_{n=1}^{N}(y_i - \beta_0 - x_i^T\beta)^2 + \lambda\sum_{j=1}^{P}|\beta_j| \tag{1}$$

with respect to $\beta_0$ and $\beta$, where: $N$ is the number of observations, $y_i$ is the response for observation $i$, $x_i$ is the p-dimensional input vector for observation $i$, $x_i$ is a non-negative regularization parameter, $\beta_0$ and $\beta$ are regression parameters ($\beta_0$ is a scalar, $\beta$ is a p-dimensional vector). In the research the matlab implementation for LASSO was used with the following parameters: $\lambda$ - 1, $k$-fold cross validation - 10, maximum number of non-zero coefficients in the model - 8.

The classification was performed together with the feature selection in a double 10-fold cross validation loop. Linear SVM classifiers were used in the classification process. The classifiers built for different IC-sets and the reference set were compared according to the validation error, calculated as the mean over 10 validation errors of the classifiers from the outer cross validation loop. To obtain even more unbias results, the whole procedure, starting from independent components calculation and ending with evaluating classifiers accuracy, was repeated 5 times per each of 15 IC-sets. Also the core set was submitted five times to the feature selection, feature extraction and classification processes.

## 3 Results and Discussion

Figure 2 presents the mean classification accuracy obtained per each of the channel combinations in all five rounds. The average classification accuracy evaluated on the core set (composed of channels C3 and C4) is marked at Fig. 2 as a straight dashed line. As it can be noticed, regardless of the channel combination, the average accuracy calculated over the set of raw signals, equal to 64% was always smaller than the classification accuracy obtained after applying ICA and our channels-to-components pairing method. The highest mean accuracy, equal to 79% was obtained for the 8th combination (IC4_O1_Cz) and the smallest for the 11th combination (IC5_O1_O2_Pz – 70%). The grand average classification accuracy calculated over all combinations of channels was equal to 75%, and hence was 17% higher than the accuracy calculated over the set composed of the unprocessed EEG data. Table 1 presents the classification accuracy obtained in each of 5 testing rounds. As it can be noticed the results obtained for succeeding rounds differed slightly, but still each one was higher than in the case of the unprocessed data. The highest gain in the classification accuracy (more than 20%) was observed when the channel set contained Cz channel. On the contrary, the smallest improvement was characteristic for sets that contained O1 and O2 channels, but not Cz channel.

Figure 2 and Table 1 shows that in each single case the application of ICA improved the classification precision. However, the increase was not the same for different combinations of channels. At the beginning of our research we believed that the more channels would be added to the electrode matrix, the higher accuracy increase would be obtained (because of the higher SNR of the output C3 and

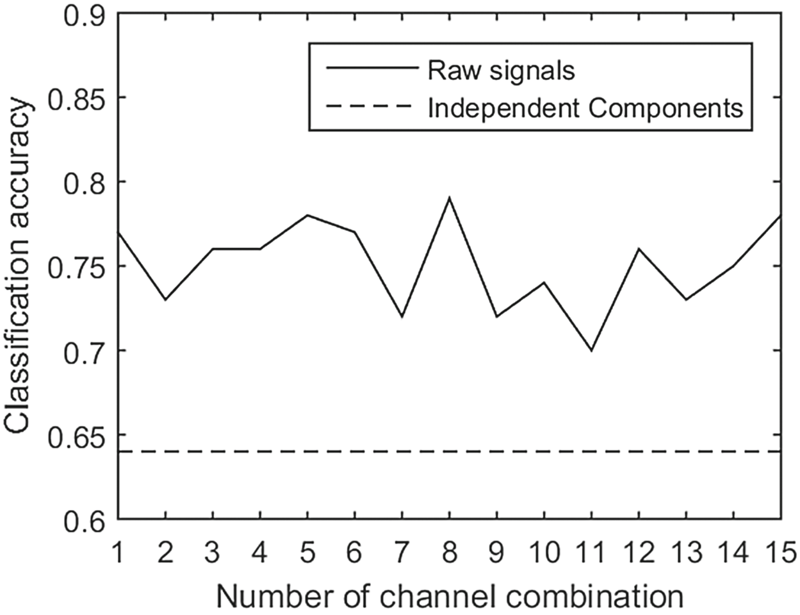


**Fig. 2.** The comparison of the mean classification accuracy of all IC-sets. The accuracy of the reference set is marked as a straight dashed line

**Table 1.** The classification accuracy obtained for the core set (first row) and all 15 sets of components (remaining rows) in succeeding test rounds; two last columns presents the accuracy averaged over 5 test rounds and the accuracy increase after applying ICA

| Combination number | Channels/ components set | Classification accuracy [%] | | | | | | Accuracy increase [%] |
|---|---|---|---|---|---|---|---|---|
| | | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Mean | |
| - | S2_(core set) | 65 | 63 | 64 | 65 | 63 | 64 | - |
| 1 | IC3_Cz | 77 | 81 | 76 | 77 | 75 | 77 | 21 |
| 2 | IC3_Pz | 73 | 73 | 75 | 72 | 73 | 73 | 14 |
| 3 | IC3_O2 | 77 | 75 | 75 | 77 | 77 | 76 | 19 |
| 4 | IC3_O1 | 72 | 76 | 74 | 79 | 79 | 76 | 18 |
| 5 | IC4_Cz_Pz | 78 | 77 | 75 | 80 | 77 | 78 | 21 |
| 6 | IC4_Cz_O2 | 78 | 75 | 75 | 76 | 79 | 77 | 20 |
| 7 | IC4_Pz_O2 | 71 | 74 | 72 | 74 | 71 | 72 | 13 |
| 8 | IC4_Cz_O1 | 78 | 80 | 81 | 74 | 82 | 79 | 23 |
| 9 | IC4_Pz_O1 | 70 | 72 | 73 | 69 | 74 | 72 | 12 |
| 10 | IC4_O1_O2 | 71 | 75 | 76 | 74 | 73 | 74 | 15 |
| 11 | IC5_Pz_O1_O2 | 72 | 69 | 67 | 67 | 74 | 70 | 9 |
| 12 | IC5_Cz_O1_O2 | 79 | 79 | 70 | 77 | 74 | 76 | 18 |
| 13 | IC5_Cz_Pz_O1 | 75 | 68 | 75 | 73 | 74 | 73 | 14 |
| 14 | IC5_Cz_Pz_O2 | 72 | 76 | 77 | 77 | 75 | 75 | 18 |
| 15 | IC6_Cz_Pz_O1_O2 | 80 | 78 | 75 | 77 | 81 | 78 | 22 |
| | Mean | 75 | 75 | 75 | 75 | 76 | 75 | 17 |

C4 signals). Meanwhile, it occurred that the number of channels is not relevant for the accuracy increase. As it can be noticed in Table 2, presenting the classification accuracy averaged over the combination of channels containing the same number of channels, the highest increase was in fact observed for a six IC-set (22%), but the smallest - for five IC-sets (15%).

**Table 2.** The classification accuracy averaged across the number of channels used in the channels combinations.

| Number of components | Mean accuracy [%] | Accuracy increase [%] |
|---|---|---|
| 6 | 78 | 22 |
| 3 | 76 | 18 |
| 4 | 75 | 17 |
| 5 | 74 | 15 |

## 4 Conclusion

The channels-to-components pairing method proposed in the paper provided the better classification accuracy than the direct classification performed on the channels from the core set. This result was stable, regardless of the number of channels and their localization. This allows to believe that first, the right components were paired to C3 and C4 channels, and second by adding additional channels to the core set, composed of channels most suitable for recognizing the imagery hand movement, improved the signal quality. Hence, we conclude that in our survey the components returned by ICA reflected the same brain activity as raw C3 and C4 channels but had higher SNR. Moreover, analyzing the results from succeeding channels combinations, we found that the highest improvement in the classification accuracy was observed when Cz channel was added to the core set. Because of the localization of Cz channel (very close to the channels from the core set) this observation might mean that Cz channel, contaminated with the same artifacts as C3, and C4 channels, seized the artifacts and as a result more clean components corresponding to C3 and C4 channels were obtained.

## References

1. Wang, Y., Shangkai, G., Xiaorong, G.: Common spatial pattern method for channel selection in motor imagery based brain-computer interface. In: Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference (2005)
2. Oja, E., Yuan, Z.: The FastICA algorithm revisited: convergence analysis. IEEE Trans. Neural Netw. **17**(6), 1370–1381 (2006)
3. Augustyniak, P.: Adaptive wavelet discrimination of muscular noise in the ECG. In: Proceedings of Computers in Cardiology (IEEE-EMB), vol. 33, pp. 481–484 (2006)

4. Rejer, I., Górski, P.: Independent component analysis for EEG data preprocessing - algorithms comparison. In: Computer Information Systems and Industrial Management. LNCS, vol. 8104, pp. 108–119. Springer, Heidelberg (2013)
5. Yang, R., Song, A., Xu, B.: Feature extraction of motor imagery EEG based on wavelet transform and higher-order statistics. Int. J. Wavelets Multiresolut. Inf. Process. **8**(3), 373–384 (2010)
6. Müller-Putz, G.R., Kaiser, V., Solis-Escalante, T., Pfurtscheller, G.: Fast set-up asynchronous brain-switch based on detection of foot motor imagery in 1-channel EEG. Med. Biol. Eng. Comput. **48**(3), 229–233 (2010)
7. Pfurtscheller, G., Neuper, Ch., Schlögl, A., Lugger, K.: Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters. IEEE Trans. Rehabil. Eng. **6**(3), 316–325 (1998)
8. Burduk, R.: Imprecise information in Bayes classifier. Pattern Anal. Appl. **15**(2), 147–153 (2012)
9. Vigario, R.N.: Extraction of ocular artefacts from EEG using independent component analysis. Electroencephalogr. Clin. Neurophysiology **103**(3), 395–404 (1997)
10. Wallstrom, G.L., Kass, R.E., Miller, A., Cohn, J.F., Fox, N.A.: Automatic correction of ocular artifacts in the EEG: a comparison of regression-based and component-based methods. Int. J. Psychophysiol. **53**(2), 105–119 (2004)
11. Jung, T.P., Humphries, C., Lee, T.W., Makeig, S., McKeown, M.J., Iragui, V., Sejnowski, T.J.: Extended ICA removes artifacts from electroencephalographic recordings. Advances in Neural Information Processing Systems, pp. 894–900 (1998)
12. Zhou, W., Gotman, J.: Removal of EMG and ECG artifacts from EEG based on wavelet transform and ICA. In: 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEMBS 2004, vol. 1, pp. 392–395 (2004)
13. Xue, Z., Li, J., Li, S., Wan, B.: Using ICA to remove eye blink and power line artifacts in EEG. In: Innovative Computing, Information and Control, vol. 3, pp. 107–110 (2006)
14. Delorme, A., Palmer, J., Onton, J., Oostenveld, R., Makeig, S.: Independent EEG sources are dipolar. PloS One **7**(2), e30135 (2012)
15. Jasper, H.H.: The ten-twenty electrode system of the international federation in electroencephalography and clinical neurophysiology. EEG J. **10**, 371–375 (1958)
16. Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., Bertrand, O., Lécuyer, A.: OpenViBE: an open-source software platform to design, test and use brain-computer interfaces in real and virtual environments. Presence Teleoperators Virtual Environ. **19**(1) (2010)
17. Raudys, S.J., Jain, A.K.: Small sample size effects in statistical pattern recognition: recommendations for practitioners. IEEE Trans. Pattern Anal. Mach. Intell. **13**(3), 252–264 (1991)
18. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Roy. Stat. Soc. Ser. B (Methodological) **58**(1), 267–288 (1996)

# Objective Description of Choral Singers Voice Quality Using Glottal-to-Noise Excitation Ratio

Edward Półrolniczak(✉)

Faculty of Computer Science and Information Technology,
West Pomeranian University of Technology, Szczecin,
Żołnierska Street 49, 71-210 Szczecin, Poland
epolrolniczak@wi.zut.edu.pl
http://www.zut.edu.pl/eng/home/news/current-news.html

**Abstract.** The article presents the results of the research joining signal analysis and medical aspects. It focuses on the analysis of the voice signals of the singers using GNE parameter. The GNE (Glottal-to-Noise Excitation Ratio) parameter has been used for the analysis of singing voice quality. The GNE values were calculated basing on the recorded samples. The results show that GNE may be useful in assessing the overall singing quality of the choir singers. The results may be useful for the development of the tools for computer analysis of singer's or speaker's voices.

**Keywords:** Voice · Singing voice · Singing quality · GNE

## 1 Introduction

The research presented in the article was driven by need to support the process of voice production training in the choral ensembles. Choirs are continuously working on developing their sound and artistic quality. The quality of the choir sound depends on the characteristics and quality of individual singers voices. The measures of voice quality can be useful in supervision of voice production. The properly selected evaluation criteria allow for corrections of particular voice characteristics in the right way. Not only singing quality but also health of the singers may benefit from such corrections. One of the various parameters (attributes) characterizing singing voice is a phenomenon connected to a noise component. It gives the impression of humming, hissing or rustling voice hereinafter named as the noisy voice as all of the above listed characteristics are connected with noise component. It may influence resonance parameter of the human voice [1]. From the medical point of view this problem is connected with disorder of vibration of the vocal folds [2]. This can result from some vocal apparatus disease or this could be just an individual voice characteristic of a singer. Sometimes a loss of high frequency components in the signal is associated with this problem. The presence of intense noise component in a singing voice can be a warning signal, inducing the singer to perform a phoniatric diagnostic. In [3] an example study on the noise in choir singers voices has been presented.

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_42

## 2 Literature Review

The degree of noise in voice signal belongs to the important group of features applied for the assessment of voice quality. The methods are primarily limited to the assessment of the accuracy of the harmonic structure hence are often referred to as $HNR$ (Harmonic-to-Noise Ratio) methods. In the literature various indicators can be found. An important indicator is $HNRYumoto$ which is estimated in the time domain. First the average glottal period is calculated and then the length of each period is normalized. Next each segment is compared with the average (segment). The variance of deviations obtained in this way is the measure of noise [4]. In the modified version, HNRQi, the dynamic time warping is used to normalize segments in the time domain [5]. $NNE$ (Normalized Noise Energy) is another measure which presents the ratio of non-harmonic signal energy to the total energy of signal. The location of the minima between the harmonics in the range 1–5 kHz, carrying the information about the level of noise, is used for its estimation [6]. The parameters presented in [7] are based on the spectral comb filtering, separating the harmonic and non-harmonic energy of the signal. These are the $HNR$, $SPI$ and $TNI$ measures. Each measure is the ratio of energy value in specific frequency ranges. Cepstral HNR ($CHNR$) [8] adapts the cepstrum to determine the baseline of spectrum. The ratio of total energy to the base energy of spectrum is then determined. Further modifications consist in better fitting the baseline to the spectrum [9]. To this end the algorithm presented in [9] is often used to calculate the measure. The discrimination between harmonic and noise energy in the magnitude spectrum by means of a comb filtering operation in the cepstrum domain is used in this method. According to the authors the method is a valid technique for determining the amount of spectral noise and may be a useful parameter in the analysis of voice quality. The $HNR$ measure is calculated in this case in the following way: a variable window length equal to 5 pitch periods is determined, then the pitch component of the cepstrum is filtered and the energy of the harmonics is compared with the noise floor. Some submeasures such as $HNR5$ which measures the $HNR$ between 0–500 Hz, $HNR15$ which measures the $HNR$ between 0–1500 Hz and $HNR25$ which measures the parameter in the range 0–2500 Hz can be found in literature. The effectiveness of the $F0$ detection algorithm is the main factor affecting reliability of all above mentioned measures. $HNRQi$ and $HNRYumoto$ measures are particularly sensitive to this due to the need of precise detection of glottal segments in the time domain. The fact that they do not allow to estimate a $HNR$ parameter for the required frequency is another disadvantage of these measures. The other measures require correct estimation of the mean $F0$ in the test signal frame. NNE determination involves significant problem with detection of harmonics and determination of their range. The calculation of $CHNR$ instead entails the problem with precise estimation of the baseline of the spectrum which is used to separate non-harmonic and harmonic energies. As a parameter competitive to ratio of noise a $GNE$ (Glottal-to-Noise Excitation Ratio) has been proposed [10]. The authors show the advantage of their method over $CHNH$ and $NNE$, as the algorithm does not require the information estimated on $F0$ trajectory, what should result in increased reliability of the measure. The disadvantage of the method is its high complexity. In this study $GNE$ parameter is used to measure the singing voice quality of choral singers in the context of noise in the analysed signal. While comparing $GNE$ to a competitive $CHNR$ parameter one should keep in mind that $CHNR$ is reported to be $F0$ dependent, therefore the results can be compared when calculated for the same pitches. The $GNE$ used in the study represents an approach quantifying the amount of voice excitation by vocal-fold oscillations versus excitation by turbulent noise. In this sense it is closely related to breathiness. The latter term however is a multidimensional perceptual phenomenon which we do not intend to model directly. The earlier methods defined parameters for breathiness or additive noise either in frequency domain or in time domain [11] and depended on the regularity of several glottal oscillations. The advantage of $GNE$ is that it is applicable even for highly irregular glottal oscillations.

## 3 Material for the Study

In this study a database consisting of representative samples reflecting the abilities of choir singers was used. The database has been created in the frame of a research project of West Pomeranian University of Technology: “Computerized methods of supporting the process of training choir voices” [12]. The decision about using own sound material resulted from the necessity of having a database which would represent the abilities of choir voices as a specific class of voices. It was assumed that there are certain singing parameters which may be measured irrespectively of what is being sung and that these parameters are influenced by what is being sung. Therefore, the creation of such a database was of significant importance. The database consists of recorded samples of five vocal exercises used as material for assessing the quality of singing voice. The exercises were selected from a set usually used during vocal trainings of the choir. The most important features for the exercises are: the text, sound scheme in reference to the key of C-major and the approximate tempo of the performance of the exercise. The selection of these exercises took into account the possibility of detecting different problems: intonation, timbre, vibrato, noise and others. The development of the database is still ongoing. Most of the recorded singers sing in the choir of the same university. The recently developed part of the database consists of the set of recordings of amateurs (non-singers). The study was carried out in twenty singers of the choir. Each recorded person was in good health. The singers represented different levels of advancement (Table 1) according to the assignments made by experts (beginner, intermediate and advanced). In the male group three singers denoted in the database as s05m, s14m and s16m were assigned to the beginners group and another three singers: s02m, s04m and s05m to the intermediate group. The last two male singers (s07m and s10m) belong to the group of advanced singers. In the female group three singers denoted as s01f, s15f and s20f were assigned to the group of beginners. The intermediate group consisted of seven singers. One female singer (s13f) was assigned to the

**Table 1.** Description of the audio database content

| Symbol of singer (DB v2.0) | Gender | Voice | Advancement | Years in the choir |
|---|---|---|---|---|
| s01f | Female | Soprano | Beginner | 5 |
| s02m | Male | Bass | Intermediate | 16 |
| s03f | Female | Soprano | Intermediate | 15 |
| s04m | Male | Bass | Intermediate | 3 |
| s05m | Male | Bass | Beginner | 1 |
| s06f | Female | Soprano | Intermediate | 5 |
| s07m | Male | Bass | Advanced | 18 |
| s08f | Female | Alto | Intermediate | 7 |
| s09f | Female | Alto | Intermediate | 3 |
| s10m | Male | Bass | Advanced | 6 |
| s11f | Female | Soprano | Intermediate | 3 |
| s12m | Male | Bass | Intermediate | 3 |
| s13f | Female | Soprano | Advanced | 15 |
| s14m | Male | Bass | Beginner | 2 |
| s15f | Female | Alto | Beginner | 2 |
| s16m | Male | Bass | Beginner | 2 |
| s17f | Female | Soprano | Intermediate | 4 |
| s18m | Male | Tenor | Intermediate | 6 |
| s19f | Female | Alto | Intermediate | 2 |
| s20f | Female | Soprano | Beginner | 1 |

advanced singers group. The investigated singers were singing the sequence of vowels ‘a-e-i-o-u’. The exercise used for the study closely reproduces the real singing conditions.

## 4 Research Background

The basic assumption in using $GNE$ coefficient is that during the closing of the vocal folds frequencies in different bands are synchronously aroused. Irregular air flow is manifested as uncorrelated frequencies in different bands. In this study the $GNE$ parameter is estimated for the band 3000 Hz in the following way [10]:

1. down-sampling of the signal to 10 kHz and the inverse linear-predictive filtering of the signal,
2. band-pass filtering with Hanning window (3000 Hz wide) with differences in median frequencies of at least 1500 Hz,
3. calculation of the Hilbert envelopes (the absolute value of the complex analytic signal) for different frequency bands with fixed bandwidth and different centre frequencies,

4. translation to the time domain,
5. estimation of the correlation parameter: every pair of envelopes for which the difference of their centre frequencies is equal or greater than half the bandwidth is considered and the cross correlation function between such envelopes is calculated,
6. the maximum of each correlation function is saved in a maxima vector, the maximum from the maxima vector obtained in the previous stage is the resulting value.

The whole process is illustrated in Fig. 1. Due to the correlation nature the maximum of $GNE$ equals 1. This is the case when two different envelopes (for different bands) are exactly the same. $GNE$ decreases when contribution of noise in voice increases and, at the same time, the frequency coefficients are reduced. This factor may be associated with the so-called humming voice in an extreme case.

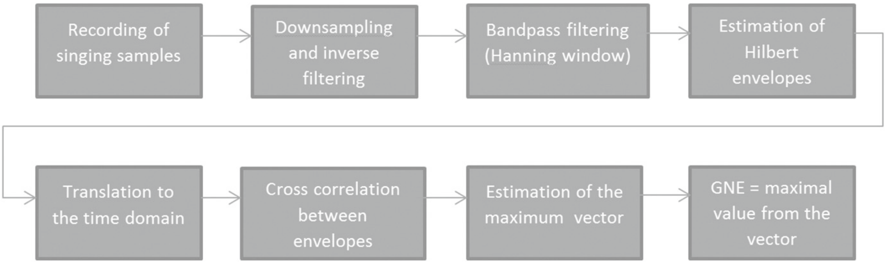

Recording of singing samples
Downsampling and inverse filtering
Bandpass filtering (Hanning window)
Estimation of Hilbert envelopes
Translation to the time domain
Cross correlation between envelopes
Estimation of the maximum vector
GNE = maximal value from the vector

**Fig. 1.** Estimation of GNE in the singing signal

## 5 The Analysis and the Results

After the singing samples were acquired the $GNE$ spectrum has been estimated for each recorded sample. The analysis of the data should give the answer to the posed questions. The study was focused on measuring the $GNE$ parameter in the choral singers. The example values obtained for the female singer s01f are presented in Fig. 2. It should be noted that singers were singing starting from the highest pitches to the lowest possible to achieve. While interpreting the Fig. 2 it can be observed that the values for the higher pitches are rather lower than for the sung sounds from the middle of the voice scale. Also at the end of the scale situation is similar although more stable. $GNE$ values decrease while increasing the contribution of noise in the voice at the same time. It means that the singer produces audible noise in the emitted sound at the beginning and at the end of her scale. Three voice production experts independently agreed with that conclusion. The final advice for the singer is to train singing sounds in the higher vocal range. Another solution is to change the voice the singer sings in to

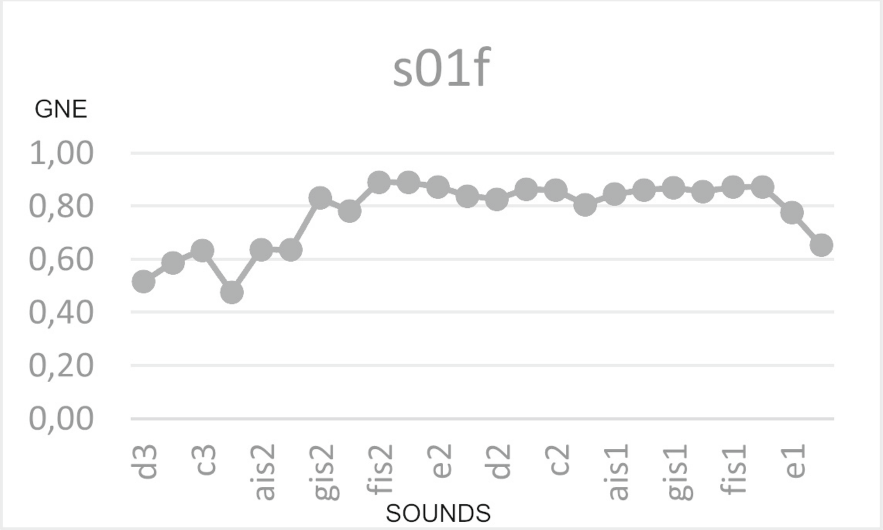


**Fig. 2.** GNE values for the singer s01f

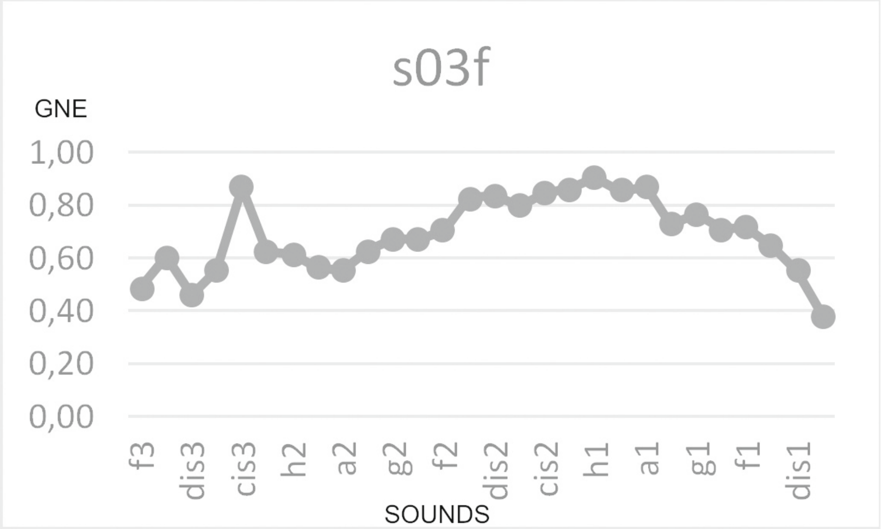


**Fig. 3.** GNE values for the singer s03f

the lower one. The mean value of *GNE* for that singer is 0,77 but, as it can be seen, without wider view the conclusion can be too general.

The singer s03f (Fig. 3) is another soprano singer, much more experienced (15 years of singing, see Table 1) but the problems are the same, or even bigger. The observed overall vocal range is a little bit wider than in previous case.

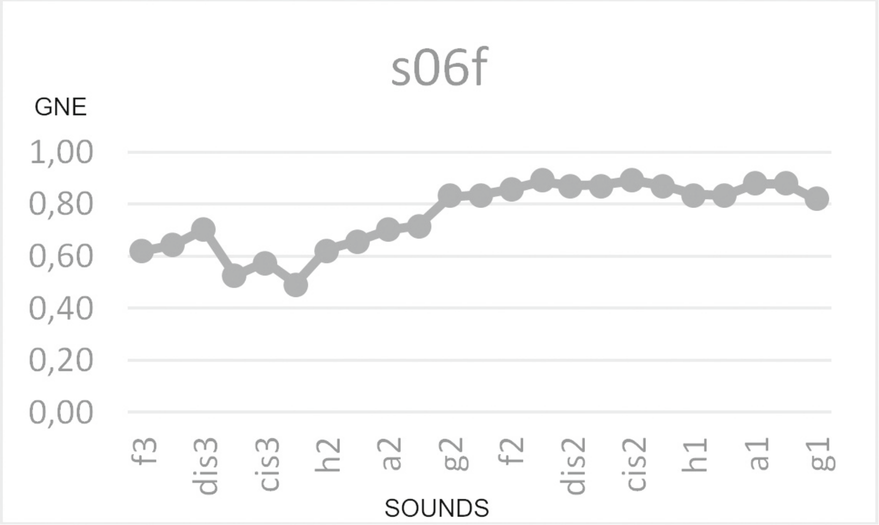


**Fig. 4.** GNE values for the singer s06f

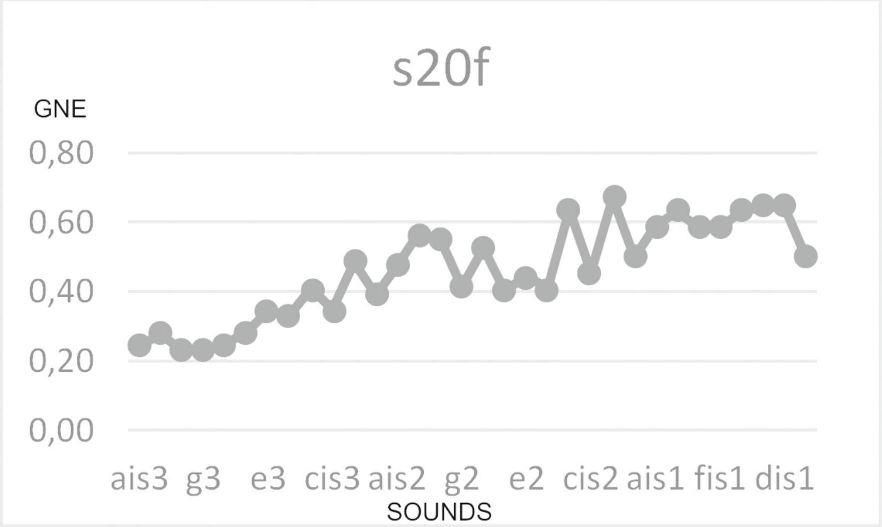


**Fig. 5.** GNE values for the singer s20f

Mean *GNE* ratio is lower than in the previously analysed case and the noise problems at the beginnings and endings of the vocal range are deeper. Proper vocal range lasts from gis1 to e2 (so is more narrow than in the previous example). The next singer, s06f, is also soprano with 5 years of singing experience (Fig. 4). Singer s06f generally sounds good (less noisy) when singing pitches

**Table 2.** Mean GNE in the context of the singing experience of the female singers

| Symbol of singer (DB v2.0) | GNE | Advancement | Years in the choir |
|---|---|---|---|
| s01f | 0,77 | beginner | 5 |
| s03f | 0,76 | intermediate | 15 |
| s06f | 0,76 | intermediate | 5 |
| s08f | 0,75 | intermediate | 7 |
| s09f | 0,68 | intermediate | 3 |
| s11f | 0,63 | intermediate | 3 |
| s13f | 0,69 | advanced | 15 |
| s15f | 0,81 | beginner | 2 |
| s17f | 0,67 | intermediate | 4 |
| s19f | 0,81 | intermediate | 2 |
| s20f | 0,48 | beginner | 1 |

below g2 but, in the contrary to the previous singers, her effective vocal range is narrow (approximately g1 to g2). Very interesting are the results obtained for the very beginner singer s20f (Fig. 5) – another soprano. The singer shows very weak *GNE* plot. It seems that the singer doesn't know the abilities of her vocal instrument. Probably she does not know how to sing properly and has no appropriate experience, no knowledge in this respect. Mean *GNE* ratio for that singer is 0,48. She needs intensive voice production training. Finally the table summarizing mean *GNE* ratios, advancement level (given by experts) of the singer and the years of singing experience is presented in the Table 2 but there is no visible direct correlation between the experience and the level of noise in the singing voice. The article omitted the results for the male voices, because the results and their interpretation are very similar.

## 6 Conclusion

The article was focused on the analysis of the voice signals of the singers with the use *GNE* parameter. The parameter may be useful to indicate the presence of noise in the singing voice. Thus the more precise question was whether the *GNE* parameter can be useful to assess quality of voice from the perspective of noise component. The noise characteristic is well recognizable by human perception and it can be used as an evaluation criterion of quality of singing voice. Referring to the literature *GNE* values decrease in case of increasing contribution of noise in the voice. The analysis has shown that noise characteristics are consistent with the assessments of experts asked to compare the results to their impressions and knowledge. The *GNE* parameter can be used to estimate the quality of the singing voice (in the context of the noise). It seems to be possible to assess properly the vocal range of the singer estimating *GNE* parameter although the examples show that it is not always possible – in particular when a singer is the very beginner. The method of signal analysis using $GNE$ is stable. The results are repeatable and gives possibility of valuable interpretation. The final conclusion is that the use of the $GNE$ analysis can be used in the systems for automatic assessment of singing quality and assessment of the level of advancement.

## References

1. Półrolniczak, E., Kramarczyk, M.: Vocal tract resonance analysis using LTAS in the context of the singer's level of advancement. In: Kobayashi, S., Piegat, A., Pejaś, J., El Fray, I., Kacprzyk, J. (eds.) ACS 2016. AISC, vol. 534, pp. 249–257. Springer, Cham (2017). doi:10.1007/978-3-319-48429-7_23
2. Hammarberg, B., Fritzell, B., Gaufin, J., Sundberg, J., Wedin, L.: Perceptual and acoustic correlates of abnormal voice qualities. Acta Otolaryngol. **90**(1–6), 441–451 (1980)
3. Półrolniczak, E., Kramarczyk, M.: Computer analysis of the noise component in the singing voice for assessing the quality of singing. Przegl. Elektrotechniczny **91**, 79–83 (2015)
4. Yumoto, E., Gould, W.J., Baer, T.: Harmonics-to-noise ratio as an index of the degree of hoarseness. J. Acoust. Soc. Am. **71**(6), 1544–1550 (1982)
5. Qi, Y.: Time normalization in voice analysis. J. Acoust. Soc. Am. **92**(5), 2569–2576 (1992)
6. Kasuya, H., Ogawa, S., Mashima, K., Ebihara, S.: Normalized noise energy as an acoustic measure to evaluate pathologic voice. J. Acoust. Soc. Am. **80**(5), 1329–1334 (1986)
7. Deliyski, D.: Acoustic model and evaluation of pathological voice production. In: Eurospeech 1993, pp. 1969–1972 (1993)
8. Qi, Y., Hillman, R.E.: Temporal and spectral estimations of harmonics-to-noise ratio in human voice signals. J. Acoust. Soc. Am. **102**(1), 537–543 (1997)
9. de Krom, G.: A cepstrum-based technique for determining a harmonics-to-noise ratio in speech signals. J. Speech Lang. Hear. Res. **36**(2), 254–266 (1993)
10. Michaelis, D., Gramss, T., Strube, H.W.: Glottal-to-noise excitation ratio-a new measure for describing pathological voices. Acta Acustica United Acustica **83**(4), 700–706 (1997)
11. de Krom, G.: Some spectral correlates of pathological breathy and rough voice quality for different types of vowel fragments. J. Speech Lang. Hear. Res. **38**(4), 794–811 (1995)
12. Łazoryszczak, M., Półrolniczak, E.: Audio database for the assessment of singing voice quality of choir members. Elektron. Konstrukcje Technol. Zastosowania **54**(3), 92–96 (2013)

# Permutation-Based Diversity Measure for Classifier-Chain Approach

Pawel Trajdos(✉) and Marek Kurzynski

Department of Systems and Computer Networks,
Wroclaw University of Technology,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
pawel.trajdos@pwr.wroc.pl

**Abstract.** In this paper, the problem of multilabel classification using the classifier chain scheme is addressed. We deal with the problem of building a diverse ensemble of the classifier-chain-based ensemble. For this purpose, we propose a permutation-based criterion of chain diversity. The final ensemble is build using a multi-objective genetic algorithm, which is used to optimise classification quality and chain diversity simultaneously. The proposed methods were evaluated using 29 benchmark datasets. The comparison was performed using four different multi-label evaluation measures. The experimental study reveals that the proposed approach provides a better classification quality than response-based diversity criteria.

**Keywords:** Multi-label classification · Classifier-chain · Diversity

## 1 Introduction

In many real-world recognition task, there appears a situation when an object is simultaneously assigned to multiple categories. This is an example of so called multi-label data [8]. Unfortunately, traditional single-label classification methods cannot directly be employed to solve this problem. A solution to this issue is a generalization of classical classification task called multi-label classification which assumes that object is described by a set of tags. Nowadays, multi-label learning becomes a useful tool for solving many data-mining tasks, since the amount of multi-labelled data increases rapidly [8]. As a consequence, multi-label learning was employed in a wide range of practical applications including text classification [13], multimedia classification [23] and bioinformatics [12,28,30,31]. In this paper, we address an issue of building a diverse committee of multi-label classifiers. Our focus is put on building a diverse ensemble of multi-label *classifier chains* (CC) algorithms [22]. To deal with this problem, we propose a model-based criterion of chain diversity. More precisely, the criterion assesses chain similarity using permutation distance [4]. To form a final ensemble, we employed a multi-objective genetic algorithm that simultaneously optimises classification

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_43

quality and ensemble diversity [15]. This paper is organized as follows. The next section (Sect. 2) shows the work related to the issue which is considered throughout this paper. The subsequent section (Sect. 3) provide a formal notation used throughout this article, and introduces the proposed algorithm. Section 4 contains a description of experimental setup. In Sect. 5 the experimental results are presented and discussed. Finally, Sect. 6 concludes the paper.

## 2 Related Work

One of the most widespread methods of dealing with multi-label tasks are decomposition methods. Methods from the this group transform an original multi-label problem into a set of simpler, single-label classification sub-problems, and then apply a combination procedure to build a single multi-label prediction from multiple single-label outcomes [8,26]. A great advantage of this approach is that it allows to use any state-of-the-art single-label classifier to solve a multi-label task. The most intuitive approach in this group is the *binary relevance* (BR) approach that decomposes a multi-label classification task into a set of *one-vs-rest* binary classification problems [26]. It means that one classifier is built to recognize one label. This approach assumes that labels are conditionally independent [26]. However the assumption does not hold in most of real-life recognition problems, the BR framework is one of the most widespread multi-label classification methods [8]. To preserve scalability of the BR classifiers and incorporate into it a model of inter-label relations, Read et al. [22] proposed the *classifier chain* model (CC) which establish a linked chain of modified BR classifiers. The modification builds a label sequence, for which the feature space of each classifier along the chain is extended with binary variables corresponding to the labels that precede the given one. The described approach passes along the chain, information allowing CC to take into account inter-label relations at the cost of allowing the label-prediction-errors to propagate along the chain [22]. Another major drawback of the CC system is that the performance of a chain classifier strongly depends on chain configuration. To overcome these effects, the authors suggested to generate an *ensemble of chain classifiers* (ECC). The ensemble consists of classifiers trained using randomly chosen label sequences. An alternative solution to building a classifier chain ensemble is to find chain structures that allows the ensemble to improve the classification quality [19].

To achieve this goal Read et al. extends the original mechanism of generating chain sequences. They proposed a strategy which uses Monte Carlo sampling to explore the label sequence space [19]. Chain sequence can be also found in a heuristic way. That is, Goncalves et al. developed a strategy that utilises a *genetic algorithm* (GA) to find an ensemble of classifier chains [9]. The other way is to build an explicit model of inter-label dependencies. In [17] authors proposed to compute label-dependence using information gain. Then they applied Page Rank Algorithm to build dependency graph. A *Bayesian Network* (BN) model can also be employed to model inter-label dependencies [24]. Since the computational complexity of BN inference is high, Read et al. proposed a simplified method based on pre-defined structures (trellis) which describe inter-label dependencies [20]. The studies on multi-label ensembles show that building ensemble classifiers focused mainly on optimisation of classification quality is often insufficient. In order to build an efficient and robust classifier ensemble, we should take into account the diversity of classifiers that constitutes the ensemble [1,3]. However, the body of literature describing this idea in the context of ML classification is not so wide as for the traditional classification [3]. For example Chekina et al. provided a ensemble generating strategy based on generating label partitions and combine their outputs [3]. Approaches which presents model-based diversity measures dedicated to CC classifier are also described [1].

## 3 Proposed Method

### 3.1 Preliminaries

Under the *multi-label* (ML) formalism a $d$-dimensional object $\boldsymbol{x} = [x_1, x_2, \ldots, x_d] \in \mathcal{X}$ is assigned to a set of labels indicated by a binary vector of length $L$: $\boldsymbol{y} = [y_1, y_2, \ldots, y_L] \in \mathcal{Y} = \{0,1\}^L$, where $L$ denotes the number of labels. Each element of the vector is related to a single label and $y_i = 1$ denotes that $i$-th label is relevant to the object $\boldsymbol{x}$. In this study we suppose that multi-label classifier $H$, which maps feature space $\mathcal{X}$ to the set $\mathcal{Y}$, is built in a supervised learning procedure using the training set $\mathcal{T}$ containing $N$ pairs of feature vectors $\boldsymbol{x}$ and corresponding class labels $\boldsymbol{y}$:

$$\mathcal{T} = \left\{(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}), (\boldsymbol{x}^{(2)}, \boldsymbol{y}^{(2)}), \ldots, (\boldsymbol{x}^{(N)}, \boldsymbol{y}^{(N)})\right\}. \tag{1}$$

When classifier validation is needed, we also use a validation set $\mathcal{V}$: $\mathcal{V} \cap \mathcal{T} = \emptyset$. The classifier produces a soft response vector:

$$\boldsymbol{r}(\boldsymbol{x}) = [d_1(\boldsymbol{x}), \ldots, d_L(\boldsymbol{x})] \in [0,1]^L \tag{2}$$

where $d_i(\boldsymbol{x}) \in [0,1]$ is label-specific support. The soft response vector is converted into multilabel output using thresholding procedure:

$$\boldsymbol{h}(\boldsymbol{x}) = [[\![d_1(\boldsymbol{x}) > 0.5]\!], \ldots, [\![d_L(\boldsymbol{x}) > 0.5]\!]] \in \mathcal{Y}, \tag{3}$$

where $[\![.]\!]$ is the Iverson bracket. In this paper, we consider ML classifiers build according to the chain rule. That is, the classifier $H$ is an ensemble of $L$ single-label classifiers $\psi_i$ that constitutes a linked chain which is build according to a permutation of label sequence $\Pi$. During the learning stage the input space of a base classifier $\psi_i$ is build according to the following rule:

$$A_i(\boldsymbol{x}^{(n)}) = \left\{\boldsymbol{x}^{(n)}, y^{(n)}_{\Pi(1)}, \ldots, y^{(n)}_{\Pi(i-1)}\right\}. \tag{4}$$

During the inference phase, due to lack of the ground-truth labels, we use predictions of previously queried single-label classifiers:

$$B_i(\boldsymbol{x}) = \left\{\boldsymbol{x}, h_{\Pi(1)}(\boldsymbol{x}), \ldots, h_{\Pi(i-1)}(\boldsymbol{x})\right\}. \tag{5}$$

Now, let us define a ML $K$-element classifier ensemble: $eH = \{H_1, \ldots, H_K\}$ whose soft prediction vector is a simple average of response vectors corresponding to base classifiers: $\widetilde{d}(\boldsymbol{x})_i = K^{-1}\sum_{k=1}^{K} d_i^k(\boldsymbol{x})$. Building of a binary response vector is performed using (3).

### 3.2 Diversity Measures

In this section, we propose a model based diversity criterion for classifiers built according CC approach. Namely, we employ similarity measures of permutations which described classifier chains that constitute the ensemble. The diversity approach is defined using pairwise comparison of ensemble members:

$$\mathrm{DivM}(eH) = \frac{2}{K * (K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} p(\varPi_i, \varPi_j), \tag{6}$$

where $p(\varPi_i, \varPi_j) \in [0, 1]$ is a normalised measure of permutation similarity [4]. During the experimental studies, we consider following permutation similarity measures (a detailed description of the measures is provided in [4]): Cayley similarity, Precedence similarity, Swap similarity, Ulam similarity Additionally, as reference methods, we provide a macro-averaged version of pairwise diversity measures which are previously proposed in [14]:

$$\mathrm{DivR}(eH, \mathcal{V}) = \frac{2}{LK(K-1)} \sum_{l=1}^{L} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} q(\psi_l^i, \psi_l^j, \mathcal{V}), \tag{7}$$

where $q(\psi_l^i, \psi_l^j, \mathcal{V})$ is one of pairwise diversity measures proposed by Kuncheva [14]: Correlation based, Disagreement based, Q measure based.

### 3.3 Ensemble Selection

In order to find the best classifier ensemble, we conducted the following procedure. First, an initial ensemble $eH$, containing $K$ multi-label classifiers, is generated. The chain orders for those ensembles are generated in a random way. Then we use NSGA-II algorithm [15] to prune the initial ensemble. As a fitness functions for the multi-objective GA optimisation we use well-known macro-averaged $F_1$ measure and one of the diversity measures defined above. Chromosomes for the GA algorithm are defined as binary vectors $c \in \{0, 1\}^K$ which selects base classifiers that are used to build a pruned ensemble.

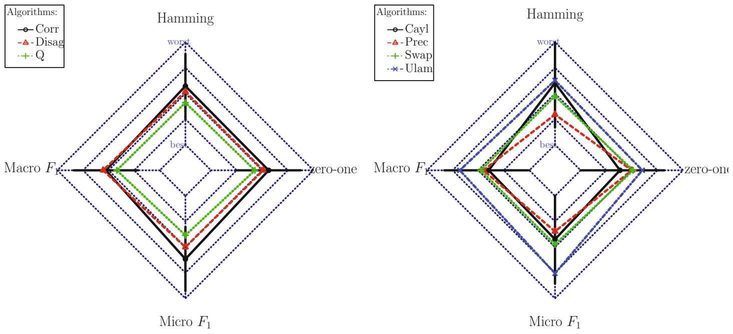
## 4 Experimental Setup

All base single-label classifiers were implemented using the J48 algorithm which is the C4.5 [18] algorithm implemented in WEKA framework [11]. The classifier parameters were set to its defaults. All multi-label algorithms were implemented using

**Table 1.** Summarised properties of the datasets employed in the experimental study. Sr denotes the source of dataset, No. is the ordinal number of a set, $N$ is the number of instances, $d$ is the dimensionality of input space, $L$ denotes the number of labels. LD, avIR, AVs are label density, average imbalance ratio and label scumble respectively [2,16,26].

| Name | Src | No. | N | d | L | LD | avIR | AVs | Name | Src | No. | N | d | L | LD | avIR | AVs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Azotobacter | [33] | 1 | 407 | 20 | 13 | .113 | 2.225 | .010 | Haloarcula | [33] | 16 | 304 | 20 | 13 | .123 | 2.419 | .016 |
| Birds | [27] | 2 | 645 | 260 | 19 | .053 | 5.407 | .033 | Human | [29] | 17 | 3106 | 440 | 14 | .085 | 15.29 | .020 |
| Caenorh. | [33] | 3 | 2512 | 20 | 21 | .115 | 2.347 | .010 | IMDB_0 | [27] | 18 | 3042 | 1001 | 28 | .071 | 24.61 | .109 |
| CAL500 | [27] | 4 | 502 | 68 | 174 | .150 | 2.578 | .337 | IMDB_1 | [21] | 19 | 3044 | 1001 | 28 | .071 | 24.56 | .106 |
| Cbmi_0 | [27] | 5 | 4381 | 100 | 101 | .044 | 307.7 | .355 | LLOG | [21] | 20 | 1460 | 1004 | 75 | .016 | 39.27 | .051 |
| Drosophila | [33] | 6 | 2605 | 20 | 22 | .121 | 1.744 | .004 | Mediamill | [27] | 21 | 4391 | 120 | 101 | .043 | 251.9 | .356 |
| Eclipse-bug | [5] | 7 | 997 | 5 | 5 | .053 | 28.92 | .018 | Medical | [27] | 22 | 978 | 1449 | 45 | .028 | 89.50 | .047 |
| Eclipse-ch. | [5] | 8 | 997 | 15 | 5 | .053 | 28.92 | .018 | MimlImg | [32] | 23 | 2000 | 135 | 5 | .247 | 1.193 | .001 |
| Emotions | [27] | 9 | 593 | 72 | 6 | .311 | 1.478 | .011 | Mylyn-bug | [5] | 24 | 1862 | 5 | 5 | .047 | 21.01 | .011 |
| Enron | [27] | 10 | 1702 | 1001 | 53 | .064 | 73.95 | .303 | Scene | [27] | 25 | 2407 | 294 | 6 | .179 | 1.254 | .000 |
| Flags | [27] | 11 | 194 | 43 | 7 | .485 | 2.255 | .061 | SimpleHC | [25] | 26 | 3000 | 30 | 10 | .190 | 1.138 | .001 |
| Flare1 | [27] | 12 | 323 | 25 | 3 | .077 | 2.423 | .005 | Tmc2007 | [27] | 27 | 2857 | 500 | 22 | .101 | 17.15 | .195 |
| Flare2 | [27] | 13 | 1066 | 27 | 3 | .070 | 14.15 | .006 | Water-qual | [27] | 28 | 1060 | 16 | 14 | .362 | 1.767 | .037 |
| Genbase | [27] | 14 | 662 | 1186 | 27 | .046 | 37.32 | .029 | Yeast | [27] | 29 | 2417 | 103 | 14 | .303 | 7.197 | .104 |
| Geobacter | [33] | 15 | 379 | 20 | 11 | .115 | 2.750 | .014 | | | | | | | | | |

MULAN [27] framework. We employed NSGA-II algorithm implemented in the MOEA framework [10]. The number of iterations and the size of population was set to 1000 and 300 respectively The sourcecode of the described approach is available online at[1]. The experiments were conducted using 29 multi-label benchmark sets. The main characteristics of the datasets are summarized in Table 1. The extraction of initial-training datasets was performed using ten-fold crossvalidation. Some of the employed sets needed preprocessing. That is, multi label regression sets (No.: 7, 8, 12, 13, 24, 28) were binarised using thresholding procedure. To be more accurate, when the value of output variable, for given object, is greater than zero, the corresponding label is set to be relevant to the object. We also used multi-label multi-instance [32] sets (No.: 1, 3, 6, 15, 16, 23) which were transformed to single-instance multi-label datasets according to the suggestion made by Zhou et al. [33]. To reduce the computational burden we use only subsets for each of IMDB, Tmc2007, Mediamill and Cbmi sets. The algorithms were compared in terms of 4 different quality criteria coming from three groups: instance-based, micro-averaged and



(a) Preliminary results for response-based criteria.

(b) Preliminary results for permutation-based criteria.

**Fig. 1.** Results of preliminary experiments. Each radar plot visualises average ranks obtained by algorithms (smaller is better). Black bars shows the critical distances for the Nemenyi post-hoc procedure.

**Table 2.** Wilcoxon test – p-values for paired comparisons of investigated algorithms Algorithms are numbered according to Sect. 4. The last row of the table presents average ranks achieved over the test sets.

| | Hamming | | | Zero-one | | | Macro $F_1$ | | | Micro $F_1$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| 1 | | 0.595 | 0.798 | | 0.698 | 0.759 | | 0.417 | 0.040 | | 0.155 | 0.269 |
| 2 | | | 0.595 | | | 0.698 | | | 0.094 | | | 0.155 |
| Rnk | 1.879 | 2.259 | 1.862 | 1.983 | 2.155 | 1.862 | 2.155 | 2.259 | 1.586 | 1.914 | 2.431 | 1.655 |

[1] https://github.com/ptrajdos/mlWorkHorse.

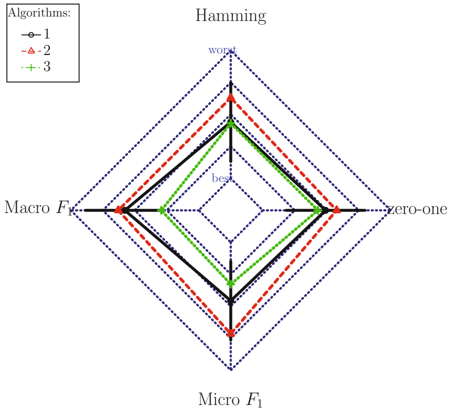


**Fig. 2.** Results of the main experiment.

**Table 3.** Outcome of the Friedman test

| Criterion | p-Value |
|---|---|
| Hamming | .448 |
| Macro $F_1$ | .063 |
| Micro $F_1$ | .040 |
| Zero-one | .511 |

macro-averaged [16]. We applied three instance-based criterion, namely zero-one loss and Hamming loss [16]. In addition we harness micro/macro-averaged $F_1$-measure [16]. Statistical evaluation of the results was performed using the Wilcoxon signed-rank test [6] and the family-wise error rates were controlled using the Bergmann-Hommel's procedure [7]. Additionally, we employed the Friedman test followed by the Nemenyi [6] post-hoc procedure. For all statistical tests, the significance level was set to $\alpha = 0.1$. The experiments were conducted using a two-step procedure. A preliminary experiment which is conducted to choose the best model based and response based diversity criteria. However, due to space limitation, we do not present full results of preliminary experiments. The second stage compares diversity ensembles build using Q-measure based, and Precedence based criteria (See Sect. 5). The main experiment compares ensembles build using previously selected diversity measures and quality criterion with an ensemble build using only quality criterion. In the following section the algorithms are numbered as follows:

1. An ensemble build using only macro-averaged $F_1$ quality measure.
2. An ensemble build using the quality criterion and Q-measure based diversity criterion.
3. An ensemble build using the quality criterion and permutation based precedence diversity distance.

The maximal and minimal size of the initial committee, that is pruned using genetic algorithm, is set to 40 and 2 respectively.

**Table 4.** Wilcoxon test – p-values for paired comparisons of investigated algorithms Algorithms are numbered according to Sect. 4.

| | Hamming | | | Zero-one | | | Macro $F_1$ | | | Micro $F_1$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | .113 | .112 | .112 | .563 | .556 | .546 | .887 | .866 | .877 | .848 | .828 | .834 |
| 2 | .049 | .048 | .050 | .501 | .496 | .498 | .613 | .605 | .610 | .562 | .556 | .571 |
| 3 | .097 | .096 | .097 | .412 | .411 | .411 | .686 | .685 | .686 | .630 | .632 | .631 |
| 4 | .139 | .140 | .139 | 1.00 | 1.00 | 1.00 | .810 | .810 | .809 | .660 | .662 | .659 |
| 5 | .033 | .033 | .033 | .911 | .913 | .912 | .764 | .764 | .763 | .484 | .485 | .485 |
| 6 | .111 | .112 | .112 | .440 | .441 | .442 | .762 | .768 | .761 | .727 | .733 | .729 |
| 7 | .044 | .045 | .044 | .190 | .192 | .190 | .523 | .523 | .523 | .536 | .538 | .535 |
| 8 | .043 | .045 | .045 | .194 | .200 | .197 | .519 | .527 | .526 | .527 | .532 | .526 |
| 9 | .215 | .210 | .211 | .732 | .735 | .725 | .386 | .376 | .382 | .363 | .355 | .357 |
| 10 | .049 | .049 | .048 | .846 | .843 | .850 | .678 | .681 | .678 | .431 | .432 | .428 |
| 11 | .272 | .277 | .275 | .752 | .793 | .773 | .357 | .362 | .359 | .280 | .287 | .283 |
| 12 | .082 | .083 | .081 | .193 | .196 | .189 | .770 | .783 | .770 | .950 | .967 | .950 |
| 13 | .070 | .070 | .071 | .190 | .190 | .193 | .726 | .727 | .726 | .808 | .812 | .811 |
| 14 | .001 | .001 | .001 | .030 | .029 | .029 | .046 | .043 | .043 | .013 | .012 | .012 |
| 15 | .120 | .122 | .120 | .578 | .594 | .570 | .810 | .824 | .805 | .799 | .806 | .783 |
| 16 | .116 | .118 | .116 | .625 | .628 | .625 | .796 | .802 | .798 | .691 | .707 | .699 |
| 17 | .085 | .085 | .086 | .854 | .852 | .850 | .894 | .891 | .893 | .742 | .743 | .742 |
| 18 | .077 | .078 | .077 | .951 | .951 | .952 | .913 | .914 | .913 | .845 | .853 | .848 |
| 19 | .077 | .076 | .077 | .941 | .944 | .944 | .918 | .915 | .916 | .842 | .834 | .839 |
| 20 | .018 | .018 | .018 | .774 | .773 | .773 | .582 | .581 | .581 | .795 | .793 | .795 |
| 21 | .031 | .031 | .031 | .904 | .900 | .906 | .748 | .750 | .748 | .462 | .463 | .461 |
| 22 | .010 | .010 | .010 | .319 | .315 | .318 | .243 | .244 | .243 | .185 | .186 | .182 |
| 23 | .195 | .201 | .196 | .617 | .618 | .615 | .455 | .463 | .454 | .455 | .463 | .454 |
| 24 | .047 | .047 | .047 | .133 | .134 | .133 | .783 | .784 | .780 | .958 | .961 | .953 |
| 25 | .109 | .107 | .108 | .428 | .418 | .423 | .326 | .320 | .324 | .328 | .322 | .326 |
| 26 | .129 | .130 | .130 | .731 | .734 | .733 | .445 | .446 | .445 | .440 | .440 | .440 |
| 27 | .073 | .072 | .073 | .789 | .781 | .794 | .557 | .557 | .556 | .393 | .388 | .393 |
| 28 | .291 | .294 | .291 | .985 | .989 | .988 | .526 | .539 | .525 | .448 | .455 | .447 |
| 29 | .196 | .198 | .197 | .813 | .815 | .812 | .604 | .604 | .605 | .354 | .355 | .355 |

## 5 Results and Discussion

Due to space limitation, the results of the preliminary experiment are presented only in graphical form in Figs. 1a and b. The visualised Nemenyi post-hoc critical distances clearly shows that there are no significant differences between tested methods. However, for response based diversity criteria the algorithm based on Q-measure, the average ranks obtained by the method are the lowest. So this diversity measure is chosen to be investigated in the second phase of the experiment. The situation is not so clear in the case of permutation-based diversity measures. That is, no single algorithm achieves the lowest average rank. However, we decided to choose diversity measure based on precedence criterion. The reasons for this choice are twofold. First, precedence based criterion wins for two quality measures and it is second two times. The second competing criterion (Cayley-measure) also win in two cases but for Hamming loss, it achieves the third score. The second reason of this choice lies in the nature of the criterion. That is, the measure counts the number of times which some labels precede given label in both permutations. This scheme is the closest to the formula of building classifier chains. The full results for the main experiments are shown in Table 4. The outcome of the statistical assessment is shown in Tables 2 and 3. Additionally, the results are presented in the graphical form in Fig. 2. The results clearly show that the proposed method achieves the highest classification equality under the macro-averaged $F_1$ criterion. Statistical significance of this result is confirmed by all statistical tests. What is more, the proposed method obtained the lowest average rank under all considered quality criteria. Interestingly, utilising the response based diversity criterion tends to worsen the classification quality for all considered quality criteria, although the difference is significant only for the macro-averaged $F_1$ measure. This result shows that for highly imbalanced BR-based transformation, the diversity criterion based on the chain structure brings more useful information that allows to improve classification quality.

## 6 Conclusion

In this paper, we investigated the possibility of utilisation of structure-based diversity criterion tailored for classifier-chain based ensembles of multi-label classifiers. During the experimental evaluation, we obtained very promising results. That is, the proposed diversity criterion allows the CC-based ensemble to significantly improve the classification quality. What is more, the proposed criterion, contrary to the reference methods, does not need a validation set to calculate the diversity of the ensemble. We are willing to continue our study on model-based diversity measures of CC-based ensembles. The next step should be considering a more complicated graph model of dependencies.

**Acknowledgement.** This work was supported by the statutory funds of the Department of Systems and Computer Networks, Wroclaw University of Science and Technology. Computational resources were provided by PL-Grid Infrastructure.

## References

1. Burkhardt, S., Kramer, S.: On the spectrum between binary relevance and classifier chains in multi-label classification. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing - SAC 2015. Association for Computing Machinery (ACM) (2015)
2. Charte, F., Rivera, A., Jesus, M.J., Herrera, F.: Concurrence among imbalanced labels and its influence on multilabel resampling algorithms. In: Polycarpou, M., Carvalho, A.C.P.L.F., Pan, J.-S., Woźniak, M., Quintian, H., Corchado, E. (eds.) HAIS 2014. LNCS, vol. 8480, pp. 110–121. Springer, Cham (2014). doi:10.1007/978-3-319-07617-1_10
3. Chekina, L., Gutfreund, D., Kontorovich, A., Rokach, L., Shapira, B.: Exploiting label dependencies for improved sample complexity. Mach. Learn. **91**(1), 1–42 (2012)
4. Czogalla, J., Fink, A.: Fitness landscape analysis for the resource constrained project scheduling problem. In: Stützle, T. (ed.) LION 2009. LNCS, vol. 5851, pp. 104–118. Springer, Heidelberg (2009). doi:10.1007/978-3-642-11169-3_8
5. D'Ambros, M., Lanza, M., Robbes, R.: An extensive comparison of bug prediction approaches. In: 2010 7th IEEE Working Conference on Mining Software Repositories (MSR 2010). Institute of Electrical & Electronics Engineers (IEEE) (2010). http://dx.doi.org/10.1109/MSR.2010.5463279
6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
7. Garcia, S., Herrera, F.: An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. J. Mach. Learn. Res. **9**, 2677–2694 (2008)
8. Gibaja, E., Ventura, S.: Multi-label learning: a review of the state of the art and ongoing research. WIREs Data Min. Knowl. Discov. **4**(6), 411–444 (2014)
9. Gonçalves, E.C., Plastino, A., Freitas, A.A.: Simpler is better. In: Proceedings of the 2015 on Genetic and Evolutionary Computation Conference - GECCO 2015. Association for Computing Machinery (ACM) (2015). http://dx.doi.org/10.1145/2739480.2754650
10. Hadka, D.: http://moeaframework.org/, http://moeaframework.org/. Accessed 9 Jan 2017
11. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software. SIGKDD Explor. Newsl. **11**(1), 10 (2009)
12. Heider, D., Senge, R., Cheng, W., Hullermeier, E.: Multilabel classification for exploiting cross-resistance information in HIV-1 drug resistance prediction. Bioinformatics **29**(16), 1946–1952 (2013)
13. Jiang, J.Y., Tsai, S.C., Lee, S.J.: FSKNN: multi-label text categorization based on fuzzy similarity and k nearest neighbors. Expert Syst. Appl. **39**(3), 2813–2821 (2012)
14. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms, 1st edn. Wiley, New York (2004)
15. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, MOEA/d and NSGA-II. IEEE Trans. Evol. Comput. **13**(2), 284–302 (2009). http://dx.doi.org/10.1109/TEVC.2008.925798
16. Luaces, O., Díez, J., Barranquero, J., del Coz, J.J., Bahamonde, A.: Binary relevance efficacy for multilabel classification. Program. Artif. Intell. **1**(4), 303–313 (2012)

17. Peng, Y., Fang, M., Wang, C., Xie, J.: Entropy chain multi-label classifiers for traditional medicine diagnosing parkinson's disease. In: 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). Institute of Electrical and Electronics Engineers (IEEE), November 2015
18. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
19. Read, J., Martino, L., Luengo, D.: Efficient monte carlo methods for multidimensional learning with classifier chains. Pattern Recogn. **47**(3), 1535–1546 (2014)
20. Read, J., Martino, L., Olmos, P.M., Luengo, D.: Scalable multi-output label prediction: from classifier chains to classifier trellises. Pattern Recogn. **48**(6), 2096–2109 (2015)
21. Read, J., Peter, R.: Meka: http://meka.sourceforge.net/, http://meka.sourceforge.net/. Accessed 29 Mar 2015
22. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Mach. Learn. **85**(3), 333–359 (2011)
23. Sanden, C., Zhang, J.Z.: Enhancing multi-label music genre classification through ensemble techniques. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, NY, USA, pp. 705–714. ACM, New York (2011)
24. Sucar, L.E., Bielza, C., Morales, E.F., Hernandez-Leal, P., Zaragoza, J.H., Larrañaga, P.: Multi-label classification with bayesian network-based chain classifiers. Pattern Recogn. Lett. **41**, 14–22 (2014)
25. Tomás, J.T., Spolaôr, N., Cherman, E.A., Monard, M.C.: A framework to generate synthetic multi-label datasets. Electron. Notes Theor. Comput. Sci. **302**, 155–176 (2014). http://dx.doi.org/10.1016/j.entcs.2014.01.025
26. Tsoumakas, G., Katakis, I.: Multi-label classification: an overview. Int. J. Data Warehouse. Min. (IJDWM) **3**(3), 1–13 (2007)
27. Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: Mulan: a java library for multi-label learning. J. Mach. Learn. Res. **12**, 2411–2414 (2011). http://dl.acm.org/citation.cfm?id=1953048.2021078
28. Wu, J.S., Huang, S.J., Zhou, Z.H.: Genome-wide protein function prediction through multi-instance multi-label learning. IEEE/ACM Trans. Comput. Biol. Bioinform. **11**(5), 891–902 (2014)
29. Xu, J.: Fast multi-label core vector machine. Pattern Recogn. **46**(3), 885–898 (2013)
30. Zhang, M.L., Zhou, Z.H.: Multilabel neural networks with applications to functional genomics and text categorization. IEEE Trans. Knowl. Data Eng. **18**(10), 1338–1351 (2006)
31. Zhang, M.L., Zhou, Z.H.: ML-KNN: a lazy learning approach to multi-label learning. Pattern Recogn. **40**(7), 2038–2048 (2007)
32. Zhou, Z.H., Zhang, M.L.: Multi-instance multilabel learning with application to scene classification. In: Advances in Neural Information Processing Systems 19 (2007)
33. Zhou, Z.H., Zhang, M.L., Huang, S.J., Li, Y.F.: Multi-instance multi-label learning. Artif. Intell. **176**(1), 2291–2320 (2012)

# Static Posed Versus Genuine Smile Recognition

Krystian Radlak(✉), Natalia Radlak, and Bogdan Smolka

Insitute of Automatic Control, Silesian University of Technology,
Gliwice, Poland
{krystian.radlak,natalia.radlak,bogdan.smolka}@polsl.pl

**Abstract.** Recognition of a posed or fake smile is a vital and challenging research topic and a growing interest has been observed from the computer vision and machine learning community. The state-of-the-art algorithms related to this field focus on the facial expressions dynamics, while several psychologists suggest that the main difference between posed and spontaneous smile should be observed in different muscles contractions in the upper part of the face. Therefore, in this work we evaluate the accuracy of recognition based only on the face appearance using the High-Dimensional Local Binary Patterns. The smile authenticity is analyzed on the set of images extracted at the smile apex phase from the UvA-NEMO database. The obtained results indicate that the analyzed algorithms can spot a fake smile much better than a human, but worse than systems that incorporate the facial dynamics.

**Keywords:** Smile recognition · Smile genuineness · Face analysis · Facial expressions

## 1 Introduction

Emotions expressed on the human face are indispensable elements of non-verbal communication [24]. The emotional state of the observed person is analyzed by observation of the facial expressions and based on these visual cues other people may understand our currently feeling emotions. However, our true emotions may be masked by others. For example a husband can respond with a loving smile to his wife, when she asks for his opinion on her new clothes, even if he is not actually interested or confident.

Apparently, the vast majority of people are poor at spotting a fake smile. The possible reason for this phenomenon is the fact that it is easier for people to be unconscious what others are really feeling. Although fake smiles often look very similar to spontaneous smiles, they are slightly different, because distinct muscles are involved and they are controlled by different parts of the brain [9].

Fake smiles can be produced consciously and intentionally, because the brain signals that create them come from the conscious part of the brain and prompt the zygomaticus major muscles in the cheeks to contract. These are the muscles that pull the corners of the mouth outwards [7].

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_44

In 1862, Duchenne, a French physician, conducted the first experimental study that examined prototypical expressions of emotion, using electrical stimulation of facial muscles. He noticed that when people feel happy, the zygomaticus major muscles are contracted and pull the mouth corners upwards into a smile. When these muscles were electrically stimulated, he found that the observed facial expression of a smile (in the absence of true happiness) did not involve the contraction of the orbicularis oculi surrounding the eye. Duchenne concluded that simultaneous contractions of these two muscles are necessary to produce a genuine expression of happiness [21]. Thus, in the literature, the spontaneous smile is also called the "Duchenne smile" [6].

The Duchenne smiles, on the other hand, are elicited by the unconscious parts of brain and when people are happy, the mouth muscles move and the orbicularis oculi and the pars orbitalis muscles raise the cheeks, producing wrinkles around the eyes and raising the eyebrows. Lines around the eyes sometimes appear also during an intense fake smiles and it can look so that the eyes are contracting and the smile is genuine. Therefore, the assumption made in [16] that the smile is spontaneous if two Facial Action Units Codes (FACS) [8] - AU6 (the cheek raiser and lid compressor) and AU12 (the lip corner puller) - are performed together may be not always true. However, according to the authors of [26], we are able to find a few indicators that distinguish these smiles from real ones. For example, when a smile is genuine, the eye cover fold - the fleshy part of the eye between the eyebrow and the eyelid - moves downwards and the end of the eyebrows dips slightly.

Due to the fact that the described features that can characterize the fake smile are not clear, we investigated how accurate a machine can be in a fake smile recognition. In this work, we evaluate the accuracy of Support Vector Machine (SVM) that is utililzing the Local Binary Patters (LBP) [19] and the proposed modification - High-Dimensional Local Binary Patterns (HDLBP) presented in [2]. We also investigated the influence of different approaches for face normalisation, while the authors in [12] estimate that the human ability for a fake smile recognition is equal to 53%. Our experiment was conducted on manually chosen images with smile in the apex phase from UVA NEMO database [5] that contains 1240 videos with posed and spontaneous smiles. The exemplary fake and genuine smiles are presented in Fig. 1.

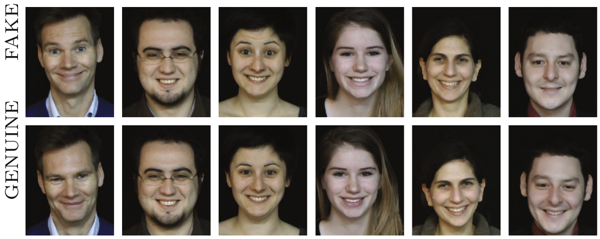


**Fig. 1.** Exemplary fake and genuine smiles extracted from UVA-NEMO database [5].

In the next Section, we discuss the state-of-the-art works focused on smile genuineness recognition. In Sect. 3, we describe the proposed framework and its variants for smile veracity recognition and present the obtained experimental results in Sect. 4. Finally, the conclusions are given in Sect. 5.

## 2 Related Work

Facial expressions recognition has been widely studied over the last 30 years and a plethora of literature has been published [1,17]. The first work in this field was published in 1978 by Suwa et al. in [23], but a significant interest in this field has been observed since the last decade of XX century and a lot of advancements emerged so far [17].

Nowadays, the research connected with the facial expressions analysis is focused on several problems: basic types of facial expressions recognition, facial action units identification and the psychology-based analysis of facial dynamics, among many others [17]. Recently, the research has expanded to include the recognition of facial expressions authenticity, due to the fact that the same facial features, which differentiate genuine and fake facial movements can be used as an indicator of deception.

Most of the state-of-the-art algorithms that were designed for recognition of spontaneous and deliberate facial expressions are focused on the analysis of the temporal changes of facial display [25]. In [3], the authors conclude that genuine smile has smaller amplitude and the ratio of duration to amplitude is bigger. Another approach is presented in [27], in which the discriminative Completed Local Binary Patterns (CLBP) are applied around five facial landmarks from three orthogonal planes and separately in three smile phases (rise, sustain and decay), which are detected using the smile detector from OpenCV library. In another study [22], the authors show that the maximum speed of the smile onset and offset phases are higher in posed samples. In [4], the authors utilized the facial tracking of the eyelid, cheek, and lip corner landmarks and extracted 25 features, such as duration of the smile phases, the amplitude or the mean speed that incorporates the dynamics of the facial landmarks movements. In the recent work [13], the authors explore the smile intensity detectors using the uniform-pattern LBP and SVM classifiers trained for different part of a face and calculate the temporal features to analyze the smile dynamics.

Previously discussed works suggest that studying dynamic properties of a smile, we are able to discriminate between falsified and sincere one. However, the research conducted by the psychologists indicate that the deliberate smile did not involve the contraction of the muscles surrounding the eyes. This finding is also confirmed in [7], what may suggest that spatial information around the eyes is also a discriminative feature. Therefore, we would like to focus on analysis of the static images to evaluate if we are able to distinguish the fake from genuine smile using only the textural information considering the temporal features.

## 3 Proposed Method

In general, the automated analysis of facial expressions can be divided into several phases. The first step is facial region detection and afterwards the facial landmarks are localized. Then, the facial display is normalized and the features are extracted and finally the facial expression is classified.

In the first step, we localize the face using the HeadHunter technique [18]. Then, the facial landmarks are localized using an ensemble of regression trees algorithm [14] that detects 68 fiducial points in the well-established configuration of Multi-PIE [10].

In the next step, the detected faces were normalized. In this work, we considered two possible normalisation approaches: affine transformation using the landmarks of the inner face (eyes, eyebrows, nose and mouth), excluding the face contour and face frontalization algorithm presented in [11], in which a single, unmodified 3D face model is used to produce a frontal view of the analyzed 2D face image. The exemplary face normalisation results obtained and the face cropped from the input images are shown in Fig. 2.

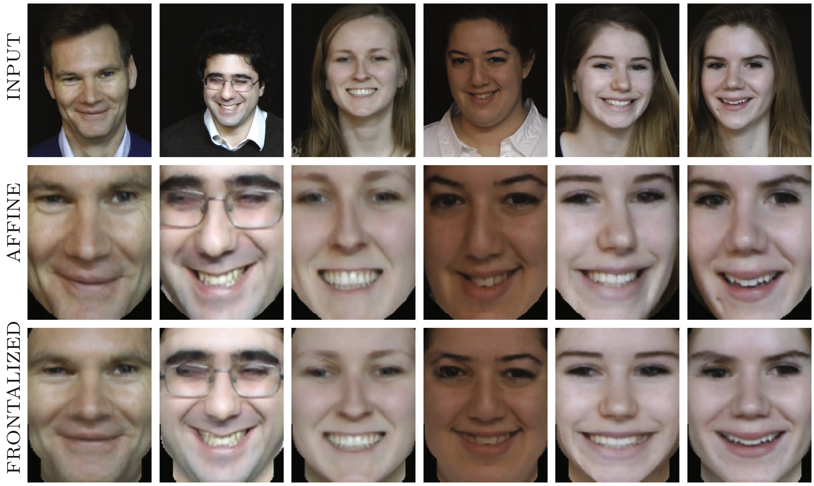
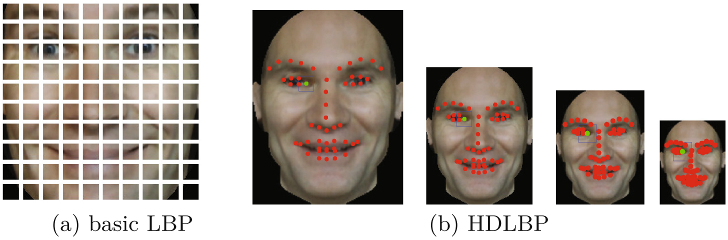


**Fig. 2.** Exemplary images exhibiting the difference between different face normalisation step. The first row depicts input images, in the second row the affine transformation has been applied and in the last row the face frontalization using 3D model is presented.

The facial features are extracted using the basic version of the LBP classification scheme in which the face image is divided into rectangular blocks and in each block the histogram of LBP features is calculated. In the second approach, the LBP histograms were calculated for each region within the patch centered around detected facial landmarks in multi-scale representation. The proposed HDLBP algorithm is a simplification of the method proposed for face recognition in [2], because in the original version the patch around each landmark was divided into 16 blocks and 16 histograms were calculated for each landmark. In our proposition only one histogram is calculated around each landmark, however we analyzed different sizes of patches $p$. Therefore, the dimensionality of our approach is much smaller and is equal to $n \times s \times b$, where $n$ denotes the number of chosen landmarks, $s$ is the number of sampling resolutions and $b$ denotes the number of bins in histograms. The comparison of the basic LBP and HDLBP scheme is presented in Fig. 3.


(a) basic LBP

(b) HDLBP

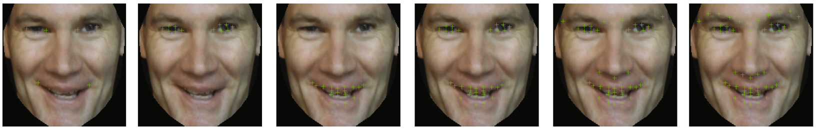
**Fig. 3.** Comparison of the basic LBP and HDLBP scheme. HDLBP is applied around each landmark.

In this work, we build an image pyramid in resolution of $400 \times 400, 283 \times 283, 200 \times 200$ and $141 \times 141$ and in both approaches we used the uniform-pattern LBP [20] with 59 bins. We also test our algorithm using different patch sizes $p = 11, 21, 31$ and we analyze different landmarks configurations (all mouth and landmarks, eyes and mouth corners and the landmarks around the nose, the eyebrows and all landmarks excluding face contours). If all landmarks excluding the face contour are analyzed, we have 51 landmarks. This gives us the maximal length of feature vector equal to $51 \times 4 \times 59 = 12016$.

Finally, in both versions of the algorithm, all LBP histograms were concatenated to form a high-dimensional feature vector. In order to reduce the dimensionality of the feature vector, the Random Frog feature selection algorithm presented in [15] was applied. In this algorithm, the selection probability of each variable is calculated and therefore, it can be used as an index for variable selection. Finally, the facial expressions are recognized using SVM classifier with the radial basis function $\mathcal{K}(a, b) = \exp(-\gamma||a - b||)$, where $a$ and $b$ are the input vectors, and $\gamma$ denotes the kernel width. We also adopted, a soft margin strategy during training to allow some vectors to be positioned on the wrong side of the decision hyperplane and this tolerance is controlled with the $C$ parameter.

## 4 Experiments

In this Section, we investigate the influence of chosen landmarks configuration and feature vector dimensionality on classification accuracy. The analyzed landmarks configuration are presented in Fig. 4.



(a) 9 points (b) 12 points (c) 20 points (d) 32 points (e) 39 points (f) 51 points

**Fig. 4.** Examined landmarks configurations in HDLBP algorithm.

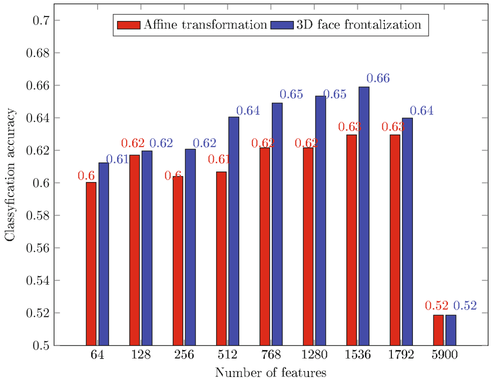


**Fig. 5.** Results obtained for different face normalisation using the basic LBP approach.

Our experiments were conducted using the static images extracted from the UvA-NEMO database. Our dataset contained 1240 images of posed and genuine smiles extracted in the apex phase of a smile (643 and 597 images, respectively). The presented results were obtained with 10-fold cross-validation using the experimental protocol presented in [5]. The values of SVM parameters ($\gamma$ and $C$) were optimized using a grid search separately for each number of the chosen features. To increase the speed of computations, we analyzed only these features for which using two-sample t-test the $p$ value $< 0.2$ and all possible features in the current landmarks configuration.

In the first experiment, we evaluated the influence of the face normalisation on the fake smile recognition using the basic LBP algorithm. The obtained results are depicted in Fig. 5. As can be observed, the face frontalization using 3D model gives much better results than the affine transformation and the accuracy is equal to 65.89%. Another interesting observation is that the Random Frog feature selection algorithm allows to obtain much better results than when considering all features.

The goal of the next experiment was to investigate the performance of the HDLBP approach, optimal landmarks configuration, face normalisation and choice of the optimal size of patch around a landmark. The experimental results are shown in Figs. 6 and 7. As can be seen, accuracies of all classifiers strongly depend on the number of chosen landmarks. The HDLBP gives slightly better results for face normalisation using affine transform. The best accuracy was

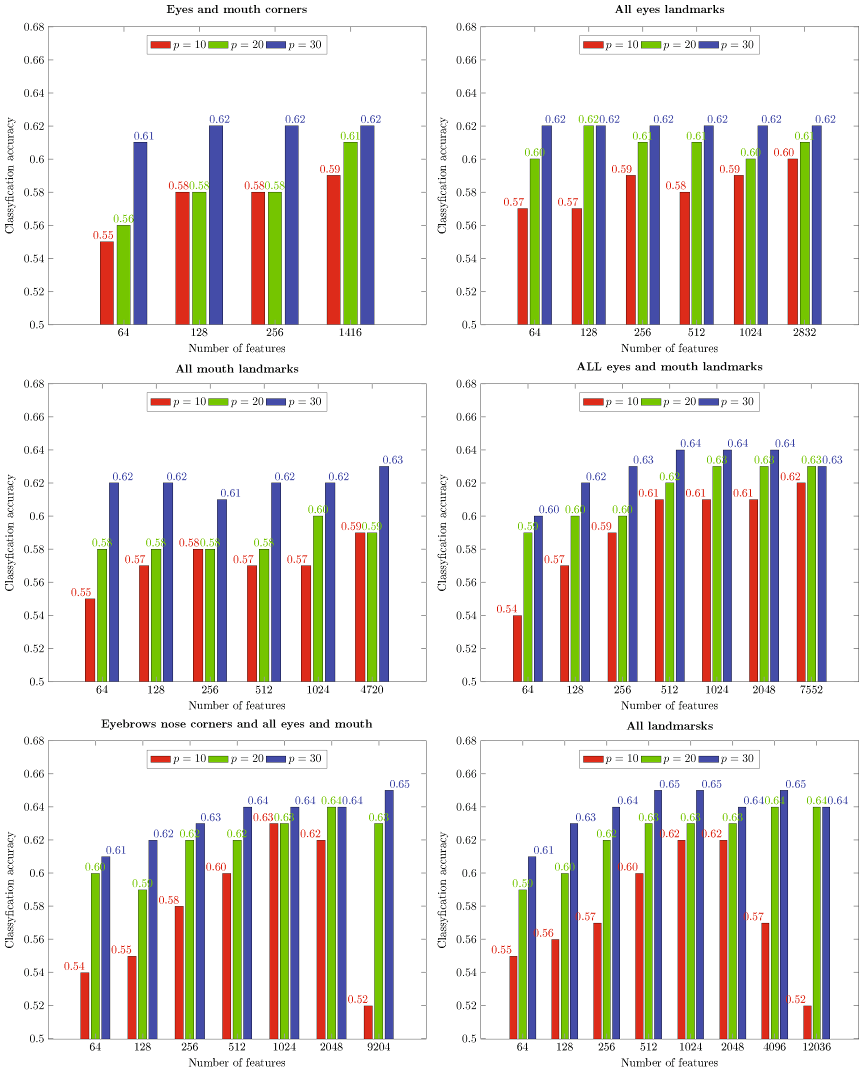


**Fig. 6.** Results for spontaneous vs. posed smile classification using HDLBP for different landmarks configurations, in which the faces were normalised using affine transformation.

obtained when considering all landmarks and is equal to 65.49%. The optimal size of the patch is $p = 31$, what may suggest that even bigger patches can produce better results. Finally, the best results were obtained for a basic LBP division scheme with 3D face frontalization and the outcomes are only slightly better than those achieved using the HDLBP. However, the accuracy results achieved on static images are much worse than those obtained for the UVA-NEMO benchmark that incorporates the face dynamics.

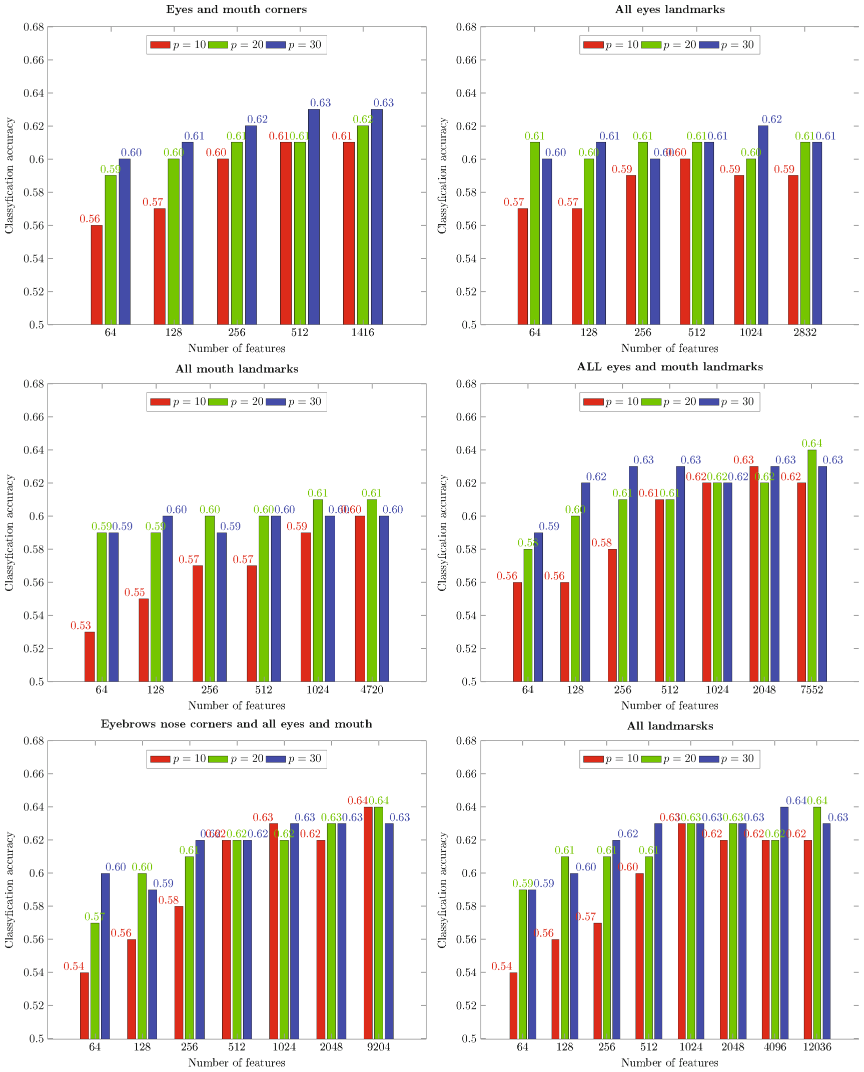


**Fig. 7.** Results for spontaneous vs. posed smile classification using HDLBP for different landmarks configurations, in which faces were normalised using 3D face frontalization.

## 5 Conclusions

In this work, we evaluated accuracy of the proposed algorithms for fake smile recognition in static images. In our approach we applied the state-of-the-art algorithms for face detection and face localization, the High-Dimensional LBP as the feature vector with the Random Frog variable selection and SVM classifier. The proposed method, which was evaluated on the static images chosen from the

UVA-NEMO database gives promising results and it significantly outperforms the ability of a typical human observer.

**Acknowledgment.** This material is based on work supported by the Polish National Science Center (NCN) under the Grant: DEC-2012/07/B/ST6/0122. This work has received funding from statutory funds (BK/213/RAU1/2017) of the Institute of Automatic Control, Silesian University of Technology, Poland.

## References

1. Calder, A.J., Young, A.W.: Understanding the recognition of facial identity and facial expression? Nat. Rev. Neurosci. **6**, 641–651 (2005)
2. Chen, D., Cao, X., Wen, F., Sun, J.: Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3025–3032 (2013)
3. Cohn, J.F., Schmidt, K.L.: The timing of facial motion in posed and spontaneous smiles. Int. J. Wavelets Multiresolut. Inf. Process. **02**(02), 121–132 (2004)
4. Dibeklioglu, H., Salah, A., Gevers, T.: Recognition of genuine smiles. IEEE Trans. Multimedia **17**(3), 279–294 (2015)
5. Dibeklioğlu, H., Salah, A.A., Gevers, T.: Are you really smiling at me? Spontaneous versus posed enjoyment smiles. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7574, pp. 525–538. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33712-3_38
6. Duchenne, G.B.: The mechanism of human facial expression or an electro-physiological analysis of the expression of emotions. Cambridge University Press (1990). Original work published 1862
7. Ekman, P., Davidson, R.J., Friesen, W.V.: The Duchenne smile: emotional expression and brain physiology: II. J. Person. Soc. Psychol. **58**(2), 342–353 (1990)
8. Ekman, P., Friesen, W.: Facial Action Coding System: Technique for the Measurement of Facial Movement. Consulting Psychologists Press, Palo Alto (1978)
9. Ekman, P., Friesen, W.V.: Felt, false, and miserable smiles. J. Nonverbal Behav. **6**(4), 238–252 (1982)
10. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-PIE. Image Vis. Comput. **28**(5), 807–813 (2010)
11. Hassner, T., Harel, S., Paz, E., Enbar, R.: Effective face frontalization in unconstrained images. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4295–4304 (2015)
12. Hoque, M.E., McDuff, D., Picard, R.: Exploring temporal patterns in classifying frustrated and delighted smiles (extended abstract). In: 2015 International Conference on Affective Computing and Intelligent Interaction (ACII), pp. 505–511 (2015)
13. Kawulok, M., Nalepa, J., Nurzynska, K., Smolka, B.: In search of truth: analysis of smile intensity dynamics to detect deception. In: Montes-y-Gómez, M., Escalante, H.J., Segura, A., Murillo, J.D. (eds.) IBERAMIA 2016. LNCS, vol. 10022, pp. 325–337. Springer, Cham (2016). doi:10.1007/978-3-319-47955-2_27
14. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1867–1874 (2014)

15. Li, H.D., Xu, Q.S., Liang, Y.Z.: Random frog: an efficient reversible jump Markov chain Monte Carlo-like approach for variable selection with applications to gene selection and disease classification. Analytica Chimica Acta **740**, 20–26 (2012)
16. Liu, H., Wu, P.: Comparison of methods for smile deceit detection by training AU6 and AU12 simultaneously. In: 2012 19th IEEE International Conference on Image Processing, pp. 1805–1808 (2012)
17. Martinez, B., Valstar, M.F.: Advances, challenges, and opportunities in automatic facial expression recognition. In: Kawulok, M., Celebi, M.E., Smolka, B. (eds.) Advances in Face Detection and Facial Image Analysis, pp. 63–100. Springer, Cham (2016). doi:10.1007/978-3-319-25958-1_4
18. Mathias, M., Benenson, R., Pedersoli, M., Gool, L.: Face detection without bells and whistles. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 720–735. Springer, Cham (2014). doi:10.1007/978-3-319-10593-2_47
19. Ojala, T., Pietikainen, M., Harwood, D.: Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In: Proceedings of the 12th IAPR International Conference on Pattern Recognition, vol. 1 - Conference A: Computer Vision amp; Image Processing, vol. 1, pp. 582–585 (1994)
20. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 971–987 (2002)
21. Porter, S., Korva, N., Baker, A.: Secrets of the human face: new insights into the face and covert emotions. Psychol. Aotearoa **3**(2), 94–99 (2011)
22. Schmidt, K.L., Ambadar, Z., Cohn, J.F., Reed, L.I.: Movement differences between deliberate and spontaneous facial expressions: Zygomaticus major action in smiling. J. Nonverbal Behav. **30**(1), 37–52 (2006)
23. Suwa, M., Sugie, N., Fujimora, K.: A preliminary note on pattern recognition of human emotional expression. In: International Joint Conference on Pattern Recognition, pp. 408–410 (1978)
24. Tracy, J.L., Randles, D., Steckler, C.M.: The nonverbal communication of emotions. Curr. Opin. Behav. Sci. **3**, 25–30 (2015)
25. Valstar, M.F., Pantic, M., Ambadar, Z., Cohn, J.F.: Spontaneous vs. posed facial behavior: automatic analysis of brow actions. In: Proceedings of the 8th International Conference on Multimodal Interfaces, pp. 162–170 (2006)
26. Weker, M.M.: Smile and lie? Why we are able to distinguish false smiles from genuine ones. In: Evers, D., Fuller, M., Runehov, A., Sæther, K.-W. (eds.) Issues in Science and Theology: Do Emotions Shape the World?. ISRPESSST, pp. 59–71. Springer, Cham (2016). doi:10.1007/978-3-319-26769-2_5
27. Wu, P., Liu, H., Zhang, X.: Spontaneous versus posed smile recognition using discriminative local spatial-temporal descriptors. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1240–1244 (2014)

# Competitive Detector of Changes with a Statistical Test

Leszek J. Chmielewski(✉), Konrad Furmańczyk, and Arkadiusz Orłowski

Faculty of Applied Informatics and Mathematics – WZIM, Warsaw University
of Life Sciences – SGGW, ul. Nowoursynowska 159, 02-775 Warsaw, Poland
{leszek_chmielewski,konrad_furmanczyk,arkadiusz_orlowski}@sggw.pl
http://www.wzim.sggw.pl

**Abstract.** The detector of jumps or changes in the function value and its derivative designed with the use of the concept of competing approximators is revisited. The previously defined condition for the existence of a jump in the function value is extended by introducing a statistical test of significance. This extension makes it possible to eliminate some false positive detections which appeared in the previously obtained results. The features of the extended detector are demonstrated on some artificial and real-life data.

**Keywords:** Competitive detector · Function change · Statistical test

## 1 Introduction

The detector of changes in a one-dimensional signal which is our object of interest originated from the concept of a filter in two-dimensional images denoted as the *competitive filter* in [10,11]. The change detection ability of this filter was later noticed in [3]. The detector did not work well in two-dimensional images [4] but for one-dimensional signals it appeared useful and made it possible to detect the change in the function value as well as in its first derivative [5]. In the present paper we shall put aside the filtering effect and pay attention to the detection of the change of the function value only. One of the problems noticed in [5] was that the detector made some false positive errors. In this paper we shall complement the basic detector with a statistical test which in our opinion will reduce the number of false positives. The question of detecting changes in signals is one of the domains of intensive research. Within the domain of image processing it was surveyed in [1,2,9] and in motion detection in [7,12]. However, it seems that the concept of competitiveness understood as in the paper [3] was absent in the research reported. Because the detector of our interest has its origin in the domain of image processing, the change will be sometimes called a jump or an edge, the change in the value of the function will be called a step and the

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_45

change in the derivative of the function can be also called a roof. The concept of the method does not need any assumption on the nature of the data analyzed. The only operation on the data is the approximation with polynomial functions, one at each side of the considered point, without the condition of continuity at this point. In the present paper we use the simple approximation with a linear function, which can be treated as the first approximation of a potentially more developed approach. The assumptions are introduced in the statistical test of the significance of the detected edge. At present, we make an assumption of the Gaussian distribution of noise, but this is also only the first, simplest example, which can easily be replaced with more advanced approaches.

This paper is organized as follows. In the next Sect. 2.1 the main concept of the detector, already described in [5], is reminded in its basic form. As a complement of the heuristic criterion of edge existence, described in Sect. 2.2, the statistical criterion proposed in this paper is introduced in Sect. 2.3. In Sect. 2.4 the functioning of the detector with both criteria is explained on simple data. Finally, the ability to reject false positives in noisy data is shown in Sect. 3.1 and the detection of changes in some real-life data is presented in Sect. 3.2. The results are discussed in Sect. 4 and the paper is closed in Sect. 5.

## 2 The Method

### 2.1 General Concept

A sequence of measurements $z(x) = y(x) + n(x)$, where the independent variable $x$ is discrete and $n(x)$ is noise, will be considered. The filtering and detection is performed in the point $x_0$ called the *central point*. If $x$ is time, the past measurements are considered to be known up to the point $x_0$, and also the further measurements, up to $x_0 + D$, are considered as known. The competitive structure of the detector can be seen in that two approximators, referred to as the *$\underline{L}$eft* and the *$\underline{R}$ight* one, are used to find $y(x_0)$. The first one operates on the *past* data at the left side of $x_0$, using $z(x), x \in [x_0 - s - \Delta, x_0 - \Delta]$ to find $\hat{y}_L(x_0)$. The second one operates on the *future* data, at the right side of $x_0$, using $z(x), x \in [x_0 + \Delta, x_0 + s + \Delta]$ to find $\hat{y}_R(x_0)$. The parameter $s$ is the scale of the filter. The parameter $\Delta$ is the gap between the central point $x_0$ and the estimators. Each approximator makes its error. Errors $e_L(x_0)$ and $e_R(x_0)$, respectively, can be approximated from the differences between the data and the approximated values. The filtered value at the central point, $\hat{y}(x_0)$, is taken as the output of that filter which has a smaller error. In this the competitiveness of the filter can be seen. As the value of the output, the value at $x_0 - \Delta$ from the left approximator, or for $x_0 + \Delta$ for the right one, is used, to avoid using the extrapolated values. This gives stabler results than in the case of values extrapolated to $x_0$. As in [3], linear least square approximators are used and their mean square errors are used as the approximation errors.

### 2.2 Heuristic Criterion of Jump

When the results from the two approximators are known, they can be used to find the jump in the function value, that is, the step edge intensity value $E_0$, and jump in the first derivatives, that is, the roof edge intensity value $E_1$, as the difference of outputs from the two approximators $\hat{y}_L(x_0)$, $\hat{y}_R(x_0)$, and their derivatives

$$\begin{aligned} E_0(x_0) &= \hat{y}_R(x_0 + \mathit{\Delta}) - \hat{y}_L(x_0 - \mathit{\Delta}), \\ E_1(x_0) &= \hat{y}'_R(x_0 + \mathit{\Delta}) - \hat{y}'_L(x_0 - \mathit{\Delta}), \end{aligned} \tag{1}$$

where $\hat{y}'_{\cdot}(\cdot)$ can be found from the approximators due to that they are linear. The question remains, where is the jump. The conditions for the existence of the jump is that the graphs of the approximation errors cross in such a way that for increasing $x$ the error from the past increases and for decreasing $x$ the error for the future increases. In the present paper it is assumed that at least one of the errors increases in this case. These conditions can be expressed as

$$\begin{aligned} &e_R(x_0 - \delta) > e_L(x_0 - \delta) \ \wedge \ e_R(x_0 + \delta) < e_L(x_0 + \delta), \\ &e_R(x_0 - \delta) > e_R(x_0 + \delta) \ \vee \ e_L(x_0 - \delta) < e_L(x_0 + \delta). \end{aligned} \tag{2}$$

Because the future error should be known for $x_0 + \delta$ then the measurements for $x_0 + D = x_0 + \mathit{\Delta} + s + 2\delta$ should be known. The parameter $\delta$ can be called the neighborhood parameter. For simplicity, it is assumed $\mathit{\Delta} = \delta = 1$, so $D = s + 3$. The crossing of the graphs of errors around a jump is illustrated in Fig. 1[1]. Let us imagine the process of filtering and edge detection in such a way that the central point, with the two approximators at its left and right side, move along the data from left to right. When a step is encountered, first the right approximator moves over it. The step enters the right approximator's support. Therefore, the error of the right approximator goes up, as in Fig. 1a. As the analyzed point moves forward, the step leaves the support of the right approximator, so its error goes down, and enters that of the left one, as in Fig. 1b. Hence, the error of the left approximator increases. When both approximators leave the region of the step, both errors go down. It can be noticed that there are no separate conditions for the two edge types detected. In the case of detection, if one of the edge types is missing, then its intensity is zero (examples will be shown further in Fig. 2, where the roof edge is zero for $x = 10, 11$ or the step edge is zero for $x = 20$).

### 2.3 Statistical Criterion of Significance of a Jump

The process of crossing the graphs of errors described in the previous Section goes on precisely in the described way provided that the edges are isolated, with respect to the scale $s$. However, it is not always so; therefore, sometimes the false positive detections (as well as false negative ones) can occur. This is why we have introduced a simple mechanism of additionally testing the edge

[1] The graphs used in this paper as well as the software were developed in Matlab®.

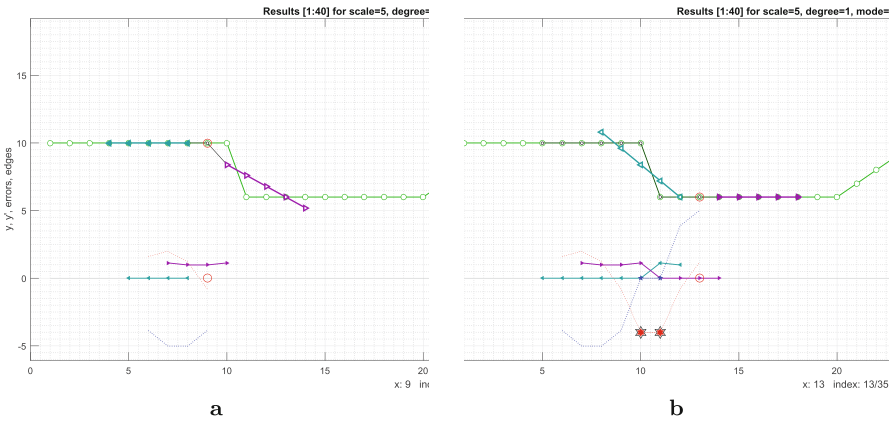


**Fig. 1.** Intermediate results for the two approximators for $x_0 = 9$, before the jump, and $x_0 = 13$, after the jump. Graphs of errors (thin magenta and cyan lines) cross between points 10 and 11 Pale green (🟩): function; dark cyan (🟦): left error, dark magenta (🟪): right error; red (🟥) star: jump of the function. Current central point $x_0$ marked with a red circle on the axis and on the graph. The left and right approximators around the central point shown with thicker cyan and magenta lines. The approximator with zero error has full triangular marks, the other one has marks filled with white. Other symbols will be explained further.

significance in a statistical way, to exclude false positive detections. In the present Subsection some notations will contain a superscript $s$ as *statistical* to underline the differences between these notations and those from the previous text. Finally the mutual relations of the relevant notations will be explained. Let us assume that the sequence of measurements form a piecewise linear signal, not necessarily continuous, with additive Gaussian noise. For an isolated point $x_0$ it is observed $y(x) = a_L + b_L x + \epsilon_x$ for $x < x_0$ and $y(x) = a_R + b_R x + \epsilon_x$ for $x \geq x_0$, where the noise $\epsilon_x$ has a zero mean normal distribution. There is a jump at $x_0$ if $\theta = a_R - a_L \neq 0$. Let us verify a hypothesis $H_0 : \theta = 0$ – the jump is absent, against the alternative hypothesis $H_1 : \theta \neq 0$ – the jump is present. To verify this, the test statistics $|\hat{y}^s_R(x_0) - \hat{y}^s_L(x_0)|$ is used, where $\hat{y}^s_L(x_0)$ is a linear regression function of $s$ points on the left of $x_0$, without this point, that is, from the set $X_L = x \in [x_0 - s, x_0 - 1]$, and $\hat{y}^s_R(x_0)$ is a linear regression function of $s$ points on the right of $x_0$, with this point, that is, from the set $X_R = x \in [x_0, x_0 + s - 1]$. An isolated jump is detected if

$$P(|\hat{y}^s_R(x_0) - \hat{y}^s_L(x_0)| > t_\alpha) = \alpha\,, \tag{3}$$

where $\alpha$ – significance level (value $\alpha = 0.05$ will be assumed throughout). Provided the hypothesis $H_0$ holds, the distribution of $\hat{y}^s_R(x_0) - \hat{y}^s_L(x_0)$ is zero-mean normal with variance $\sigma^2_L + \sigma^2_R$, where $\sigma^2_L = \mathrm{Var}(\hat{y}^s_L)$ and $\sigma^2_R = \mathrm{Var}(\hat{y}^s_R)$. Therefore,

$$t_\alpha = \sigma \Phi^{-1}(1 - \alpha/2)\,, \tag{4}$$

where $\sigma = \sqrt{\sigma_L^2 + \sigma_R^2}$. In practice, as $\sigma_.$, the standard residual error of the respective estimator $\hat{y}_.^s$ is taken. The step of the derivative has not been considered at present. The test used can be considered as a greatly simplified, basic version of tests described in [6,8]. It should be noticed that the normality of the noise distribution is assumed and this assumption will not be verified. Moreover, for some of the measurements studied with the method considered, such an assumption is merely a convenient model, but not an actual process in which the signal appears. Nevertheless, we shall use this model as a way to interpret the data in which unknown processes give rise to complex patterns and in which we seek an explanation in terms of simplified events.

### 2.4 Results for both Criteria

In the heuristic detector, as the jump in $x_0$ the relation between measurements concerning points $x_0 - 1, x_0 + 1$ are considered, while in the statistical detector this concerns $x_0 - 1, x_0$. To use the condition (3) in the common setting together with the conditions (2) the following should be noted. Due to the structure of sets $X_L, X_R$ around $x_0$, the following relations between the approximations from the heuristic notation and the regressions from the statistical notation hold

$$\begin{aligned} \hat{y}_L^s(x_0) &= \hat{y}_L(x_0)\,, \\ \hat{y}_R^s(x_0) &= \hat{y}_L(x_0 - 1)\,, \end{aligned} \tag{5}$$

and similarly the error measures from the heuristic notation can be related to the standard residual errors from the statistical notation. To come to a common meaning of a jump in $x_0$ it can be considered, in the statistical formulation, that a jump exists if there is a jump between $x_0 - 1, x_0$ or between $x_0, x_0 + 1$. If only one jump exists, its value $\theta$ is taken as the step of the function $y(x)$. If both are present, the one having a larger modulus is taken. If there is an edge according to (2) and (3), then a statistically significant edge exists. If (2) holds and (3) does not, than the edge is statistically insignificant and it is dismissed. If (2) is false, then there is no need to check (3), although in the present paper both conditions are calculated independently to show the results in a detailed way. In Fig. 2 the result is shown for data in which all the changes detectable by the heuristic algorithm are present: a step edge, a roof edge and a combined step and roof edge. The data are synthetic and clean. What is apparent is that the roof edge is detected in a single point, like this at $x = 20$, while the step edge, like this at $x = 10, 11$ is found at two points. This is correct due to that the jump of a discrete function appears between two points. It can be noted that the statistical criterion tends to detect very small changes of the value if the error measures are small, due to that in (4) the threshold depends on the variance. This gave rise to a continuous edge between $x = 20$ and 26. This edge was not accepted by the heuristic condition, though.

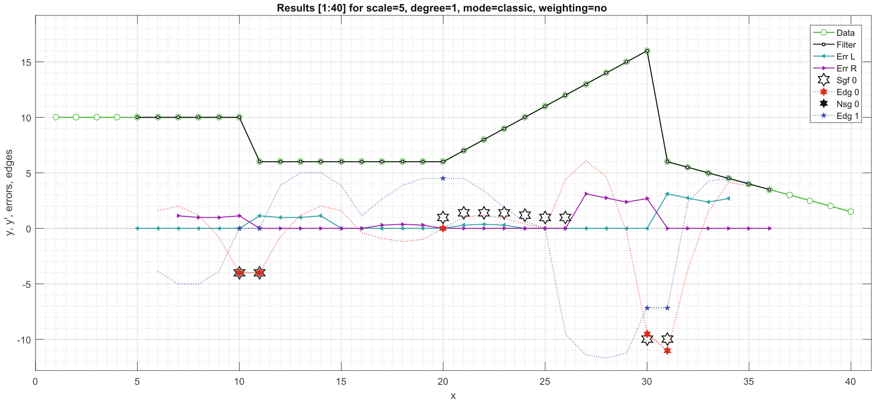


**Fig. 2.** Results for synthetic data with all the detectable changes represented: jump of the value at $x = 10, 11$, jump of the derivative at $x = 20$ and combined jump of value and derivative at $x = 30, 31$. The meaning of types and colors of lines and marks partly explained in the legend. `Err L`: left error, `Err R`: right error; `Edg 0`: jump of the function, marked with a red star; `Edg 1`: jump of its first derivative, marked with a blue star. `Sgf 0`: Statistically significant edge ((3) true) marked with an empty black star (slightly larger than other stars so that they do not obscure each other). `Nsg 0`: Statistically nonsignificant edges ((2) true and (3) false), marked with full black stars (there are no such points in this image). Angles shown in tens of degrees.

# 3 Examples

## 3.1 Rejection of False Positives in Noise

The ability to reject the less significant step changes was tested with data with noise which was actually Gaussian, zero-mean, with $\sigma = 10$. The results are shown in Fig. 3. It can be seen that indeed, some, but not all, false positive detections were successfully rejected, while the most significant, strong jump at $x = 30, 31$ was constantly maintained.

## 3.2 Real-Life Example

As an example let us consider the graph of processor load of some web server in 100 intervals of one minute each, measured on 04 July 2016 (Fig. 4). The load was averaged in each minute. The graph indicates an uneven load so there are important changes of the values. It is interesting to see that the competitive detector tends to find the steps but is not sensitive to the changes consisting in a steady increase or decrease of the value. The statistical detector, however, detects a steady increase, like in Fig. 4b at $x = 60, 65$. In this way, the criteria cooperate in forming the right decision and complement each other.

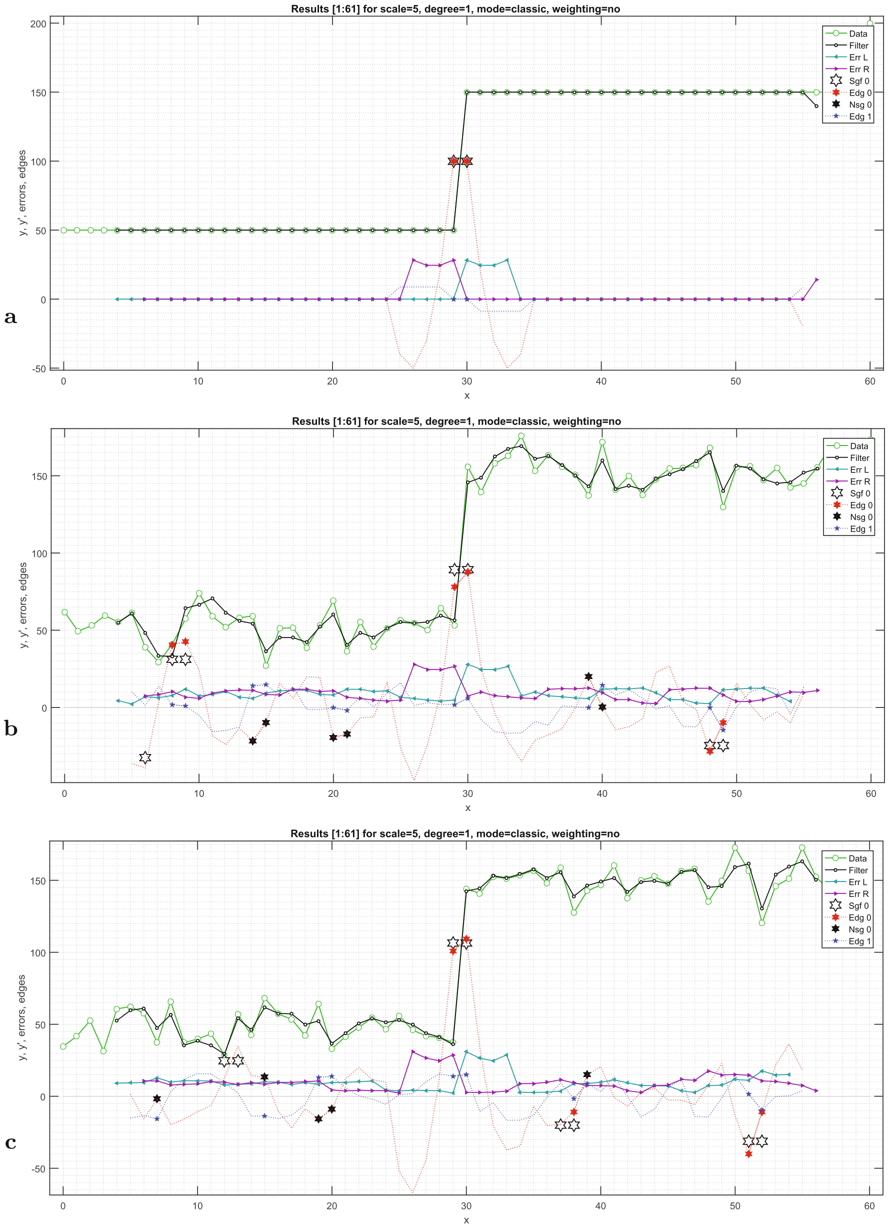


**Fig. 3.** A test step without noise (a) and with additive zero-mean noise with $\sigma = 10$ ((b) and (c), two realizations). It can be seen that some false positives were successfully rejected.

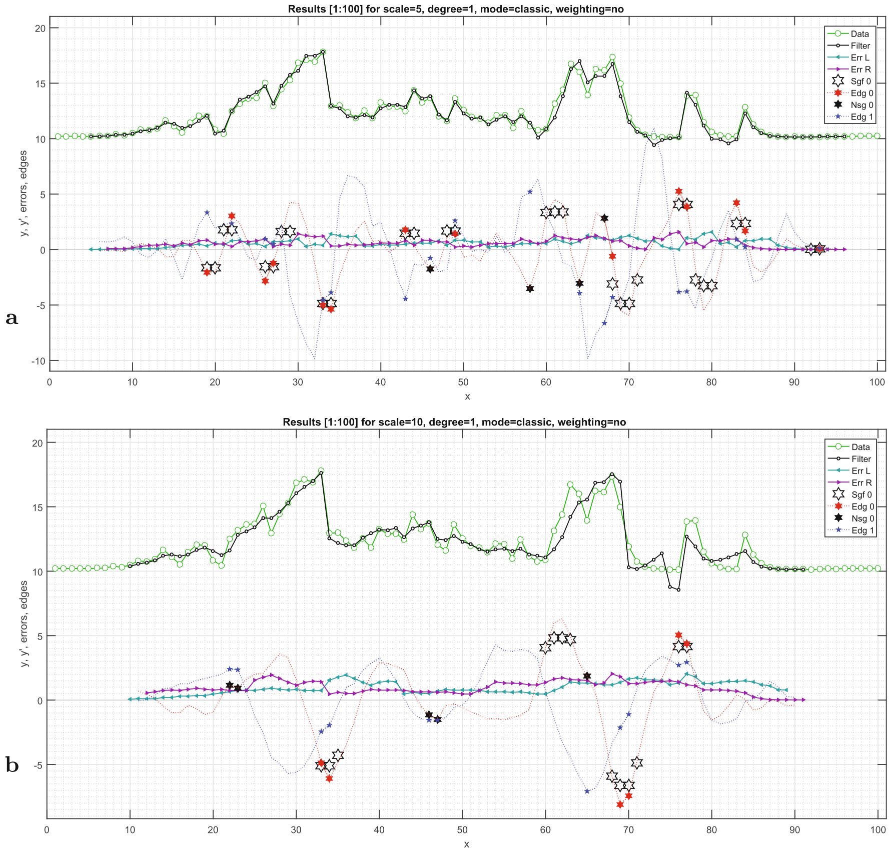


**Fig. 4.** Analysis of CPU load of a web server averaged for 1 min intervals in 100 min. Scale: (a) $s = 5$ measurements; (b) $s = 10$ measurements. At both scales it can be seen that some of the less significant jumps were dismissed by the statistical criterion. At the larger scale the minute details are neglected. Angles shown in tens of degrees. Results scaled and moved up by 10 units to make the error graphs visible.

## 4 Discussion

The use of the statistical test reduced the number of false positive detections. The number of false negatives sacrificed seems to be small, but this needs further analysis. The considered algorithm is characterized by a set of advantages and drawbacks. As the advantages, the following features can be named. Two approximators are used so the jump can be directly modelled. The complexity of the algorithm with respect to the size of the data is linear due to that only a local neighborhood of a data point of a fixed size is considered. Coming to the drawbacks, it should be said that some data concerning the future with respect to the considered data point should be known to perform the analysis, and that the method has some free parameters which should be selected, while the criteria for such selection are not self-explanatory.

## 5 Summary and Prospects

The concept of the competitive filter was extended by adding the statistical test used to check the significance of the jump of the function value. In the test, the results available from the calculations already performed are used, so the computing load is small and the complexity of the algorithm remains linear with respect to the data size. The introduction of the statistical test made the number of false positive detections smaller. The test can be used as a post-processor of the detection results. The design of the test can be extended to the derivatives of the function and its form can be improved. Also the assumptions on the distribution of noise in the data can be changed and the criterion can be reformulated accordingly. This stage of research can be treated as the proof of concept only, but the idea of combining the statistical testing with the heuristics seems to be one of the promising directions of the development of the concept of competitive filtering and detection.

## References

1. Basu, M.: Gaussian-based edge-detection methods: a survey. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **32**(3), 252–260 (2002). http://dx.doi.org/10.1109/TSMCC.2002.804448
2. Bhardwaj, S., Mittal, A.: A survey on various edge detector techniques. Procedia Technol. **4**, 220–226 (2012). http://dx.doi.org/10.1016/j.protcy.2012.05.033
3. Chmielewski, L.: The concept of a competitive step and roof edge detector. Mach. Graph. Vis. **5**(1–2), 147–156 (1996)
4. Chmielewski, L.: Failure of the 2D version of the step and roof edge detector derived from a competitive filter, Report of the Division of Optical and Computer Methods in Mechanics, IFTR PAS, December 1997
5. Chmielewski, L.J., Orłowski, A.: Detecting changes with the robust competitive detector. In: Alexandre, L.A., Sánchez, J.S., Rodrigues, J.M.F. (eds.) Proceedings of the 8th Iberian Conference on Pattern Recognition and Image Analysis IbPRIA 2017. LNCS, vol. 10255. Springer, Faro, Portugal, 20–23 Jun 2017. doi:10.1007/978-3-319-58838-4_39
6. Furmańczyk, K., Jaworski, S.: Large parametric change-point detection by a V-box control chart. Sequential Anal. **35**(2), 254–264 (2016). http://dx.doi.org/10.1080/07474946.2016.1165548
7. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **34**(3), 334–352 (2004). http://dx.doi.org/10.1109/TSMCC.2004.829274
8. Jaworski, S., Furmańczyk, K.: On the choice of parameters of change-point detection with application to stock exchange data. Quant. Methods Econ. **12**(1), 87–96 (2011)
9. Maini, R., Aggarwal, H.: Study and comparison of various image edge detection techniques. Int. J. Image Process. (IJIP) **3**(1), 1–11 (2009)

10. Niedźwiecki, M., Sethares, W.: New filtering algorithms based on the concept of competitive smoothing. In: Proceedings of the 23rd International Symposium on Stochastic Systems and their Applications, pp. 129–132. Osaka (1991)
11. Niedźwiecki, M., Suchomski, P.: On a new class of edge-preserving filters for noise rejection from images. Mach. Graph. Vis. **1**–**2**(3), 385–392 (1994)
12. Räty, T.D.: Survey on contemporary remote surveillance systems for public safety. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **40**(5), 493–515 (2010). http://dx.doi.org/10.1109/TSMCC.2010.2042446

# A Novel Simulated Annealing Based Training Algorithm for Data Stream Processing Ensemble Classifier

Konrad Jackowski(✉)

Department of Systems and Computer Networks, Wrocław University
of Science and Technology, Wyb. Wyspianskiego 27, 50-370 Wrocław, Poland
konrad.jackowski@pwr.edu.pl

**Abstract.** Training of compound ensemble classifier systems might be computationally complex and hence time consuming task. Not only elementary classifiers are to be trained, but also model of the ensemble has to be updated. Therefore, an efficiency of the training shall be considered as a compound quality which consists of not only a classification accuracy but also a running time. This gains a special importance while dealing with data streams where data arrive at high pace and the system update shall be done promptly. In this paper we present an application of Simulated Annealing based algorithm for training of data stream processing ensemble. The evaluation of our method is performed in series of experiments which show that our ensemble perform very effectively in term of accuracy and processing time.

## 1 Introduction

Ensemble classifier systems gained highly appreciated positions among wide range of machine learning methods [2]. This is due to their ability of elevating classification accuracy by fusion of a knowledge collected in a diversified pool of elementary predictors [1]. On the other hand, a training procedure of such a complex models is usually more time consuming. This is due to the fact, that an ensemble training affects not only one elementary classifier but set of them and, in many cases, fusion model has to be updated too. This aspects gains special importance when the ensemble is used for processing data streams especially such which features concept drift, i.e. changes in data characteristics [11]. Appearance of the each subsequent concept (also named context) must be followed by prompt classifier update. Any delay might lead to deterioration of classification accuracy or even the system obsolescence.

There are many methods of following the drift in data stream. Some algorithms use drift detectors which triggers update procedure when essential changes can be observed in incoming data [6]. Alternatively the classifier might me updated continuously. This approach is more commonly exploited in the ensemble systems [8]. In this case, classifier committee consists of predictors

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_46

trained in subsequent moments of time, usually on sequentially received data chunks. Therefore, each predictor in the committee represents knowledge most valid for the moment when it was trained. There are many ways how the committee can be updated.

Most popular approaches relay on assumption that committee size is fixed and usually consists of several members only [10]. The recently created classifier replaces the "oldest" one in the committee or such that shows the worst classification accuracy measured on recent data chunk. Both methods are relatively simple and intuitive, nonetheless they do not guarantee forming the best ensemble. What more, fixed and small committee size means that some knowledge about data stream concept is lost when respective classifiers are removed from the committee. Nevertheless, in most cases this algorithm works very well as it allow to maintain ensemble validity [12]. But, when the concept appears in the stream periodicity (this phenomenon is called recurring context [3]) then permanent forgetting might be costly, because recurrence of the concept requires creating new classifier from scratches.

Therefore in our previous work [7] we proposed Evolutionary Adapted Ensemble (EAE) algorithm which maintained small voting committee and large pool of all created classifiers. A new ensemble members was developed when the concept drift had been detected. Removed from the committee classifiers was not deleted but was put into the pool for further use. Ensemble training was based on selecting committee member from the pool. This procedure allowed to recall all past concepts.

EAE training was implemented as an optimisation procedure utilising genetic algorithms (GA) [5]. There are two main features which makes GA highly effective. First is their stochastic nature which allows to browse solution space avoiding falling into local minimums. The second is using population of individuals which represent possible solutions of the problem, what allow to maintain high diversity of population. Nonetheless, the last feature makes GA computationally complex. Therefore we searched for alternative which application would be less time consuming.

In this paper we propose a new of Simulated Annealing based Ensemble SAE which uses simulated annealing (SA) in its training procedure. SAE is designed to follow concept drift and preserve classifier in a pool as it was modelled in EAE. But, two important modification of EAE were implemented in SAE in order to optimize its training time. Firstly, now the size of classifier pool is also fixed, although its size is essentially larger then the size of voting committee. Because the pool is expected to preserve all concepts which can appear again in the future, the pool size shall be set large enough. Setting the size of the pool is an arbitrary decision of the system user and shall be based on they knowledge of the problem for which system is designed. Secondly, simulated annealing (SA) algorithm is used for training [9]. SA is an optimisation algorithm which search for solution in stochastic meaner, but does not processed any population what was a main reason of its selection. The goal of research presented in this work is to evaluate classification effectiveness of EAE (SA).

The rest of the paper is organized as follow. Model of ensemble classification algorithms is presented in Sect. 2. Next, some details of SA based training procedure is provided in Sect. 3. Evaluation of the EAE is presented in Sect. 4. The last section concludes the results.

## 2 SAE - Ensemble Classifier Model

We start with introducing some basic terms and symbols.

Let us assume that we deal with classification problem, i.e. assigning given object being analysed to one of possible classes. Set of classes $\mathcal{M}$ are predefined and has fixed size $M$. Decision is made by classification algorithm $\Psi$ based on observation of selected attributes $x$.

$$\Psi \rightarrow \mathcal{M}, \tag{1}$$

SAE is an ensemble system $\Psi^{SAE}$ which collect elementary classifiers $\Psi_k$ preserved in a pool $\Pi$.

$$\Pi^{SAE} = \{\Psi_1, \Psi_2, ..., \Psi_K\}. \tag{2}$$

Not all the classifiers from $\Pi^{SAE}$ contribute in decision making. Let $\Xi^{SAE}$ state for set of indexes of classifiers taken from the pool which join the committee.

$$\Xi^{SAE} = \{c_1, c_2, ..., c_E\} \tag{3}$$

A weighed fusion of committee members' discriminating functions is implemented in SAE for decision making. Weights of classifier voices are gathered in a set of weights.

$$\mathcal{W}^{SAE} = \{w_1, w_2, ..., w_E\} \tag{4}$$

Finally, the formula of ensemble decision is defined in (5)

$$\Psi^{SAE}(x) = \arg\max_{i=1}^{M} \sum_{e=1}^{E} w_e d_{c_e}(x, i), \tag{5}$$

where $d_{c_e}(x, j)$ states for discriminating function which represents support given by a $e$-th member of the committee classifier for class $j$. Note, that $e$-th member of the committee is a classifier with index $c_e$, i.e. $\Psi_{c_e}$ .

Training process aims at reducing misclassification rate and is performed based on learning set $\mathcal{LS}$, i.e. set of pairs of features $x$ and respective class labels $j$.

$$\mathcal{LS} = \{(x_1, j_1), (x_2, j_2), \ldots, (x_N, j_N)\} \tag{6}$$

The objective function, which estimate misclassification probability on LS is defined as follow

$$Q^{SAE}(\mathcal{LS}) = \frac{1}{N} \sum_{n=1}^{N} L(\Psi^{SAE}(x_n), j_n) = \frac{1}{N} \sum_{n=1}^{N} L(\arg\max_{i=1}^{M} \sum_{e=1}^{E} w_e d_{c_e}(x_n, i), j_n), \tag{7}$$

where $L$ is the zero-one loss function. There are two elements which are affected by training process:

1. $\Xi^{SAE}$ - indexes of classifiers which form the committee, and
2. $\mathcal{W}$ - values of classifiers' weights.

# 3 SAE - Training Algorithm

Simulated annealing is an optimisation algorithms which is inspired by an respective physical process. Algorithm has a stochastic nature, i.e. it search for the optimal solution in an iterative and somehow random meaner. The solution is encoded in an form of a point $\mathcal{V}^{SAE}$ consisting set of all variable being updated during optimisation. In a case of SAE $V^{SAE}$ consists of two elements described in the last paragraph of Sect. 2.

$$\mathcal{V}^{SAE} = [\Xi^{SAE}, \mathcal{W}^{SAE}] = [c_1, c_2, ...c_e, w_1, w_2, ..., w_e] \tag{8}$$

The structure of the point is of a high importance as its both constituents have a different meaner and type. While the $[\Xi^{SAE}$ is the vector of integers, the $\mathcal{W}^{SAE}$ is the vector of real numbers. That fact affects the simulated annealing procedures. Pseudo-code of the algorithm is presented below (Algorithm 1).

**Algorithm 1.** Overview of SAE training algorithm

```
Require   : $\mathcal{LS}$
            $\Pi^{SAE}$ - Pool of elementary classifiers
            $\mathcal{V}^{SAE}_{t-1}$ - The last solution vector
            $Q_{t-1}$ - The last error rate
            $SAParams$ - SAE parameter set
if not IS EMPTY $\Pi^{SAE}$ then
    $Drift$:=DetectDrift($\mathcal{LS}$)
end
if $Drift$ or is empty $\Pi^{SAE}$ then
    $\Psi$:=CreateClassifier($\mathcal{LS}$) Add $\Psi$ to $\Pi^{SAE}$
end
for $t = 1$ to $SAParams.MaxIteration$ do
    $\mathcal{V}_{temp}$ :=Annealing($\mathcal{V}^{SAE}_{t-1}$) $Q_{temp}$ := Evaluate($\mathcal{V}_{temp}, \mathcal{LS}$)
    $\mathcal{V}_t$ :=Update($\mathcal{V}_{temp}, Q_{temp}$) $Q_t$ := Evaluate($\mathcal{V}_{temp}, \mathcal{LS}$)
end
Return $\mathcal{V}^{SAE}$,$\Pi^{SAE}$,$Q_t$,
```

## 3.1 Sequential Stream Processing

SAE processes data stream, therefore data samples are extracted from the stream sequentially and are collected in chunks. The size of the chunk are arbitrarily chosen parameter. SAE training is lunched each time when the chunk is filled with requested number of samples. There are three SAE input parameters:

1. $\mathcal{LS}$ - a current data chunk extracted from the stream
2. $\Pi^{SAE}$ - a classifier pool. At the very beginning of the stream processing the pool is empty. Its constituents are added in a course of stream processing.
3. $\mathcal{V}_{t-1}^{SAE}$ - the solution vector obtained at the last run of the algorithm.
4. $Q_{t-1}$ - the most recent error rate.
5. $SAParams$ - set of parameters which control SA algorithm such as number of iterations.

### 3.2 Initialisation

If the pool $\Pi^{SAE}$ is not empty, it means that SAE has been run at least one time processing previous data chunk or chunks. That allow to check whether the drift appears in the stream (line 1-3). In SAE error base detector is used. It evaluates error $Q_t$ of the last committee $\mathcal{V}_{t-1}^{SAE}$ (i.e. such that was created in the last run) over the most recent data chunk $\mathcal{LS}$. Next, $Q_t$ is compared with $Q_{t-1}$. Significant increase in the error indicates drift. This is the first reason for creating a new classifier, the second one is running the SAE for the very first data chunk, i.e. when the pool is empty.

### 3.3 Annealing

Annealing is a process of generating some trial point $\mathcal{V}_{temp}$ at the distance from current point $\mathcal{V}$. The distance is straight proportional to a temperature (line 9). It has to be mentioned here, that this procedure must performed separately for two parts of the point. [$\Xi^{SAE}$ has to consists of integers in a range between 1 and $K$, i.e. number of classifier in the pool. $\mathcal{W}^{SAE}$ is a real number vector, therefore, it can be manipulated using standard random number generator.

The temperature controls the annealing and is decreased gradually in each iteration. Next, the trial is evaluated with a target function (7) and compared with previous one (lines 10-11). If the trial point is better than the old one, it becomes a new $\mathcal{V}_t$. Otherwise the trail point can still become a new $\mathcal{V}_t$ with some small probability which is proportional to the current temperature. The annealing process is repeated until the maximum number of iteration is reached. As the result SAE returns three objects (line 14): (1.) the $\mathcal{V}_t$ - the most current solution, (2.) classifier pool, and (3.) the current error rate.

## 4 SAE - Evaluation

A performance of SAE was checked in three series of tests which aimed at evaluation how selected parameters of the classification system and drift characteristics affected the performance. We examined the following parameters:

1. size of data chunks,
2. size of committee pool,
3. strength of drift.

**Table 1.** Details on benchmark datasets

| Dataset | Auto MPG | Pima Indians Diabetes | Glass |
|---|---|---|---|
| # instances | 398 | 768 | 214 |
| # classes | 2 | 2 | 9 |
| # attributes | 6 | 8 | 4 |
| # experiment (section) | 4.2 | 4.3 | 4.4 |

## 4.1 Experimental Set up

All the experiments were carried on in MATLAB 2014 framework using OPTIMTOOL toolbox, and PRTools toolbox [4]. Three benchmark datasets downloaded from UCI repository were used in tests (Table. 1).

Data streams were created using random generator with distributions estimated on datasets. A concept drift was also injected into the streams. Subsequent contexts were simulated by rotation of the feature space with chosen angle, which was a parameter of the drift generator and was used to control the strength of the drift, i.e. the higher the angle, the stronger the drift. For comparative analysis, four classifiers were implemented and tested, namely:

1. SAE - presented in this paper algorithm;
2. EN - Rep.Old - Ensemble updated by replacing the oldest classifier in the majority voting committee.
3. EN - Rep.Worst - Ensemble updated by replacing the classifier with the highest individual misclassification rate. Majority voting strategy was used for decision making.
4. Last - The elementary classifier which was most recently added to the pool.

Evaluation of the performance were performed using *Test-Then-Train* procedure. Samples from the stream were added to data chunks. Then the chunk was used for testing the classifier and the misclassification rate was saved. Next, SAE training procedure was launched with the chunk.

All tests were repeated 10 times, therefore presented results shows average misclassification rates.

## 4.2 Experiment 1. Misclassification Rate vs Size of Data Chunk

In the first experiment we checked how the size of data chunk affects the misclassification rate. The stream used in the test was generated base on Auto MPG dataset using the following parameters: number of concepts - 4, number of samples in each concept - 800, drift strength - 45 degree, the committee size - 5, the pool size - 10.

Results for four chunk sizes (i.e. 100, 200, 300, and 400) are presented in Table 2 and in Fig. 1.

1. Regardless the size of data chunk SAE algorithm always outperformed both competing ensemble classifiers.

**Table 2.** Misclassification rate vs size of data chunks

| Size of data chunk | SAE | En Rep. Old | En Rep. Worst | Last |
|---|---|---|---|---|
| 100 | 0.138 | 0.336 | 0.179 | 0.144 |
| 200 | 0.146 | 0.234 | 0.174 | 0.146 |
| 300 | 0.147 | 0.189 | 0.179 | 0.147 |
| 400 | 0.144 | 0.143 | 0.146 | 0.145 |

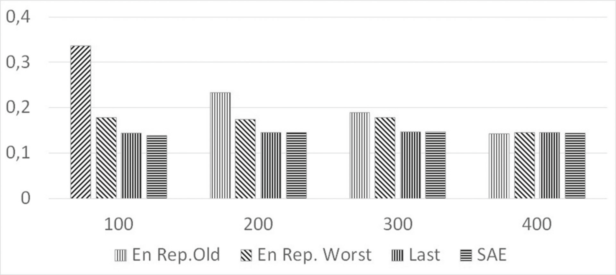


**Fig. 1.** Misclassification rate vs size of data chunks

2. In two cases SAE got the same result that Last classifier (for 200 and 300 samples in the chunk). In other two cases SAE was slightly better.
3. Increasing number of samples does not affect SAE quality. Other ensemble classifiers improved their quality.

The first observation shows that strategies of ensemble updating by replacing the oldest or the worst classifier does not produce the best committee. It does not take into consideration the knowledge of the entire ensemble. SAE selects the committee based on evaluation of the ensemble results, what proved to be better strategy.

The last observation proved that SAE is much more effective in extracting knowledge from samples comparing to other ensemble classifiers. The small misclassification rate of SAE was achieved after processing 100 samples in data chunk, while the other ensembles need much more samples to reduce their error rate.

### 4.3 Experiment 2. Misclassification Rate vs Committee Size

In the second experiment we evaluated how the ensemble performance depends on the committee size. The stream used in the test was generated base on Pima Indians Diabetes dataset using the following parameters: number of concepts - 2, number of samples in each concept - 700, drift strength - 45 degree, size of data chunks - 100, the pool size - 7.

**Table 3.** Misclassification rate vs committee size

| Committee size | SAE | En Rep. Old | En Rep. Worst |
|---|---|---|---|
| 3 | 0.225 | 0.277 | 0.254 |
| 5 | 0.221 | 0.255 | 0.240 |
| 7 | 0.212 | 0.234 | 0.232 |

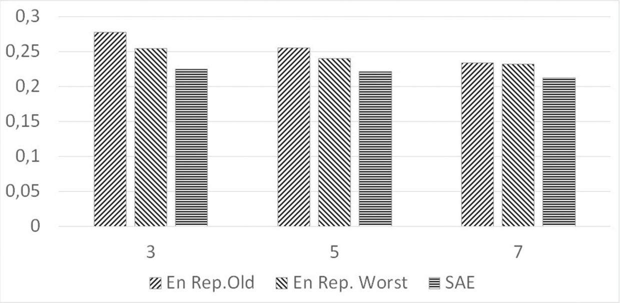


**Fig. 2.** Misclassification rate vs committee size

Results for three values of committee size (i.e. 3, 5, and 7) are presented in Table 3, and Fig. 2. The Last classifier was excluded from the test because it is simple classifier which uses no committee for decision making.

1. In all cases SAE got better result that other ensemble classifiers.
2. increasing committee size allowed to reduce error in En Rep.Old and En Rep.Worst ensemble but does not affected SAE performance.

The observations shows that strategy of preserving larger pool of available elementary classifier and selecting even small subset for decision making maintains higher flexibility of the ensemble. Removing classifier permanently from the committee limit chances for exploiting knowledge gathered in the past. Therefore SAE competitors require much larger committees to reduce the error. Worthy to note is a fact, that even when committee is extended to 7 (the size that is equal to the SAE pool size) competing ensembles did not get the same result as SAE.

### 4.4 Experiment 3. Misclassification Rate vs Strength of Drift

In the third experiment we evaluated how the strength of the drift affected error rate. The stream used in the test was generated base on Glass dataset using the following parameters: number of concepts - 2, number of samples in each concept - 300, size of data chunks - 100, the pool size - 10, the committee size - 3.

Results for four different drift strength (i.e. 1, 5, 10, 45 degree) are presented in Table 4, and Fig. 3.

**Table 4.** Misclassification rate vs strength of drift

| Drift streanth | SAE | En Rep. Old | En Rep. Worst | Last |
|---|---|---|---|---|
| 1 | 0.243 | 0.260 | 0.269 | 0.272 |
| 5 | 0.245 | 0.262 | 0.264 | 0.270 |
| 10 | 0.269 | 0.301 | 0.291 | 0.279 |
| 45 | 0.269 | 0.306 | 0.295 | 0.281 |

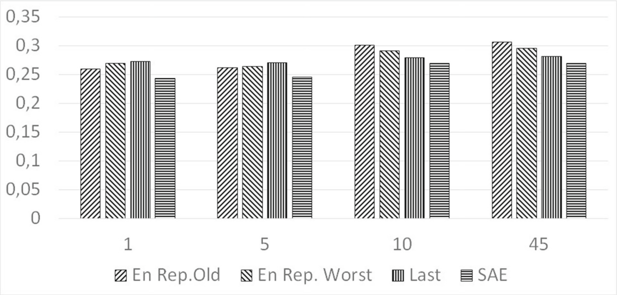


**Fig. 3.** Misclassification rate vs strength of drift

1. In all tests SAE got the best results among tested classifiers.
2. Increasing the drift strength cased elevating of classification error of all classifies. This is visible by comparison results for strength 5 and 10, where the error changes are the most profound.
3. The weakest relation between the strength and error can be observed in the Last classifier. This is due to fact, that its training means updating/replacing only one classifier not set of them, therefore such a simple system can adapted to changes much easier that the ensembles.
4. SAE are affected by a drift strength more then the last classifier but in general it got better result.

Concluding. SAE is strongly affected by the strength of drift as all other classifiers. Nonetheless, the those relation is much weaker that in case in other ensembles what shall be assessed as a positive feature.

## 5 Conclusion

In the paper we presented a Novel Simulated Annealing Based Training Algorithm for Data Stream Processing. This is an algorithm which adapt ensemble model for changes in the stream characteristic. In the implementation simulated annealing was used for training and presented tests showed, that this optimisation methods is efficient create the classifier which can outperform some other ensemble methods. This work was supported by the Polish National Science Centre under the grant no. DEC-2013/09/B/ST6/02264.

## References

1. Alpaydin, E.: Introduction to Machine Learning (Adaptive Computation and Machine Learning), vol. 5. The MIT Press (2004). http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0262012111
2. Bishop, C.M.: Pattern Recognition and Machine Learning, Information Science and Statistics, vol. 4. Springer (2006). http://www.library.wisc.edu/selectedtocs/bg0137.pdf
3. Chen, S., Wang, H., Zhou, S., Yu, P.S.: Stop chasing trends: discovering high order models in evolving data. In: 2008 IEEE 24th International Conference on Data Engineering, pp. 923–932 (2008), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4497501
4. Duin, R.P.W., Juszczak, P., Paclik, P., Pekalska, E., de Ridder, D., Tax, D.: PRTools4, A Matlab Toolbox for Pattern Recognition. Delft University of Technology (2004)
5. Eiben, A.E., Smith, J.: Introduction to Evolutionary Computing. Springer (2003)
6. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: SBIA Brazilian Symposium on Artificial Intelligence, pp. 286–295. Springer (2004)
7. Jackowski, K.: Fixed-size ensemble classifier system evolutionarily adapted to a recurring context with an unlimited pool of classifiers. Pattern Analysis and Applications, February 2013. http://link.springer.com/10.1007/s10044-013-0318-x
8. Kuncheva, L.I.: Classifier Ensembles for Changing Environments, pp. 1–15 (2004)
9. van Laarhoven, P.J.M., Aarts, E.H.L.: Introduction. In: Simulated Annealing: Theory and Applications, pp. 1–6. Springer, Netherlands (1987). http://link.springer.com/10.1007/978-94-015-7744-1_1
10. Street, W.N., Kim, Y.: A streaming ensemble algorithm (SEA) for large-scale classification. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2001, pp. 377–382 (2001). http://portal.acm.org/citation.cfm?doid=502512.502568
11. Tsymbal, A.: The problem of concept drift: definitions and related work (2004)
12. Widmer, G., Kubat, M.: Effective learning in dynamic environments by explicit context tracking. In: European Conference on Machine Learning, pp. 227–243. Springer (1993)

# Tweet Classification Framework for Detecting Events Related to Health Problems

Marcin Majak[1](✉), Andrzej Zolnierek[1], Katarzyna Wegrzyn[2], and Lamine Bougueroua[2]

[1] Wrocław University of Science and Technology, Wrocław, Poland
{marcin.majak,andrzej.zolnierek}@pwr.edu.pl

[2] AllianSTIC, Efrei Group, 33 rue Victor Hugo, 94800 Villejuif, France
{katarzyna.wegrzyn,lamine.bougueroua}@groupe-efrei.fr

**Abstract.** In this paper we present and validate the MC (Multiclassifier) system for Tweet classification related to flu and its symptoms. Proposed method consists of a preprocessing phase applying NLTK processor with converter from text corpora into feature space and as a last step ensemble of heterogenous classifiers fused at support level for Tweet classification. We have checked two methods for translating text into feature space. The first one uses standard Term Frequency times Inverse Document frequency, while the second one is enriched with hashtag analysis and word reduction after n-grams generation. Our preliminary results prove that *Twitter* can be an excellent platform for sensing real events. The most important task in proper event detection is a feature extraction technique taking into account not only text corpora, but also sentiment analysis and message intention.

**Keywords:** Short message processing · Tweet classification · Event detection

## 1 Introduction

From the past few years we are observing huge development of social media platforms. Online chats, short information and hot topic presentation are the main sources of social communication and even a way for transferring very important and critical messages [2,11]. Constant analysis of social media currently plays a crucial part because it can give an instant reflection on real-world events and dynamics. In some cases, increased user's activity on online platforms can be a signal for future event. Proper data analysis and filtering can help to prepare and correctly react on events almost in real-time. Recently, in the literature there is a term, stating that social media streams become accurate sensors of real-world events. On one hand, this gives us opportunity to react on upcoming event, but on the other, proper and effective short text message analysis is a great challenge. We should use different methods and algorithms well known

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_47

from standard document corpora processing because short social media message are noisy and can contain abbreviated words and even a slang. For some online platforms, presented posts can have specific stop words or message identification such as hashtags or related users names. Apart from aforementioned difficulties, social platform monitoring gives us really helpful view at what is happening around the world, and most importantly what people are thinking. The use of such social media content as *Twitter*, *Facebook* and *Snapchat* is broad, ranging from recommendations systems, event detection to stock market analysis, to name only a few. What is more important, it may also give us a useful insight into understanding of public health and initiate government's faster and appropriate reaction to some event [13]. In public health, prediction models may help to target preventive interventions to subjects at relatively high risk of having or developing a disease. From all available social media platforms, *Twitter* has been one of the most important data sources for data analysis. Apart from it's *API* being open source and developer friendly, it has mostly real content. In general, *Twitter* is a social network that allows the user to freely publish short messages, called Tweets via the Internet, instant messaging or SMS. The person connected to *Twitter*, can "follow" the Tweets displayed according to his own subscribers (people chosen to follow) or to publish opinions or what they are doing and thinking. According to recent statistical data, *Twitter* reported that it is now seeing over 317 million monthly active users and 500 million Tweets are generated per day, marking quite a bit of growth over the past year. Last year, the company claimed that 300 billion Tweets have been sent since March 21, 2006 and there are about 1 million sites with embedded Tweets. One thing is certain: *Twitter* is one of the main social networks for its rich and voluminous content, but also for its ease of access. In this paper, we deal with short message classification and event detection. There are many methods, but roughly they can be divided into two groups: supervised and unsupervised [4,10,14]. The first approach is used when we want to detect known events such as health problems, politics events or dangerous situation. In this case, language corpora in defined beforehand and we train classifier or a pool of classifiers on pre-processed features [9,12]. Lately, these standard classification methods have been applied in many fields for text analysis such as: sensing positive or negative emotions, politics elections and health problems or vaccination [5,7,8]. The second technique for event sensing is more general and can be used for social media monitoring. Mostly, it applies different clustering algorithms as a main tool for dividing messages into similar categories with further advanced upcoming data processing required for creating new topics or message merging to existing ones. In this study, we present a framework for Tweet filtering and identification which is based on a supervised classification. As a first step, we would like to check how proposed processing pipeline is able to detect Tweets related to flu and its symptoms. For this purpose we use Tweet database collected from September 2015 to January 2016 containing approximately 10 GB short messages. The organization of this paper is as follows. In Sect. 2 we characterize Tweet messages and method for database collection together with common preprocessing steps related to conversion from text into feature space. Proposed method for Tweet classification is described in Sect. 3 along with results from two experiments scenarios. Whole paper is concluded in Sect. 4 with proposal for further improvement and application usage.

## 2 Tweet Data

### 2.1 Short Message - Tweet

Tweet is a short message text restricted to 140 characters. Due to this limitation messages are concise, but with many oral languages and emoticons, including spelling errors, abbreviations and slang. Optionally Tweets are followed with a link to more detailed source of information. In general, collected Tweet can be described by the following properties:

- tweet language,
- time zone,
- geolocation information,
- favourited, retweeted number,
- user identification.
- hashtags - main words describing message topic and destination.

### 2.2 Data Collection

Searching the Tweets requires constant connection with the social network *Twitter* using popular REST API architecture. This method is universal and allows getting the content of Tweets by using any language with previous OAuth authentication. Rest APIs are widely used nowadays and are known for its reliability. It provides programmatic access to read and write *Twitter* data, create new Tweet, read user profile and follower data, and more. For constant message retrieval, our system used the *Twitter* Streaming API v1.1 and cached the received Tweets in a PostgreSQL database. For speeding up treatment and facilitating data linkage, the raw Tweets were then parsed into an indexed relational database schema. Filtering was done by a separate Python module and ran independently from the other steps. Final results for this study were saved as csv files for further treatment and evaluation.

### 2.3 Environment Setup

For this study we have provided an experimentation system written in Python. For data processing we have used well known scikit-learn package which is a powerful tool for document clustering, classification and text decomposition into features [1]. For further text processing NLTK processor was used [6].

### 2.4 Data Processing

Standard procedure for text analysis involves noise removal and normalization. We have seen that our collected database is quite noisy containing many commercials, article headlines and retweets. In our study, each Tweet message is subjected to the following four preprocessing steps. Each phase can be shortly described as:

- **Tokenization:** In general, Tweet posts is a combination of abbreviations, typos, or conventional word variations and any sort of punctuation and hyphenation. During preprocessing phase, we apply Tokenizer tool to extract bags of cleaner terms from the original messages by removing stop words, punctuation, compressing redundant character repetitions, and removing user's and proper's names. Apart from punctuation, we have created a stop-word list common for *Twitter* users to express various feelings.
- **Stemming:** This process aims at reducing inflected words to their root, so that related words map to the same stem. This is very important step since it reduces the number of words associated with each Tweet and as a result the final feature space is significantly decreased. In this study SnowBall stemming algorithm was used.
- **Text normalization:** After proper text filtering each Tweet is also normalized. It means that text is converted to small letters and all hashtags and user names are removed. Additionally, retweets messages are connected to one to avoid confusion in model training. Messages that are incomplete, contain errors or outliers are thrown out from the training set.
- **Language filtering:** We removed non English posts, firstly by filtering on the declared language of the *Twitter* account and secondly by using the 'langid' language detection package. This way we remove noisy messages that are declared with tag as "en", but contain Tweets in other languages.

## 3 Experiments

### 3.1 Algorithm Construction

In this section details of algorithm for Tweet identification will be presented. Our goal is to classify upcoming Tweets as related to flu or not. To successfully accomplish this task we propose an ensemble of classifiers (MC) system, which in our simulations, was constructed from the following base classifiers:

- Linear support vector machine with stochastic gradient descent (SGD) learning,
- Logistic Regression classifier,
- A decision tree classifier with Gini impurity as a function to measure the quality of a split,
- Gaussian Naive Bayes,
- 10-NN (Nearest Neighbor classifier),
- Nearest centroid classifier.

In general, MC system is a set of individual classifiers which decisions or supports are fused to produce a final decision. The greatest challenge for constructing MC system is a choice of a set of accurate and diverse base classifiers. This diversity can be obtained by applying classifiers with different design paradigm (heterogenous) or training the same base classifiers with other subset of the training set (homogenous). Another important aspect is connected with MC's final label assignment. We have used fusion at a support level. This means that weighted averaging algorithm was implemented in which instead of taking the crisp label from each classifier for majority voting we average supports and finally, the class label with the highest support is taken. For text classification, one of the important thing is a conversion from text into features easy to understand by classifiers. In this work we have implemented two methods roughly visualized in Fig. 1. In the first one, training Tweets were processed using text language technique for converting text message into scalar features. In the second method, we have added one additional feature connected with hashtags. If a Tweet contains hashtags and some of tokenized words are related to flu then we assign 1, otherwise 0. What is more, we have enriched preprocessing step, by removing words that did not appear more than a certain frequency. To accomplish this we have converted text corpora into n-grams dictionary to calculate frequencies. Later we can find words in the whole corpora which are very rare and does not transfer any valuable message, so should be removed from Tweet message. At this stage we have two parameters to be selected: n-gram size (**ng**) and threshold frequency for word removing (**f**). Each base classifier from MC system used different settings and to obtain them we have applied grid-search technique with randomly selected 500 Tweets from database. Each run checked for which **ng** and **f** classifier obtained the highest recall ratio. In this simulation **ng** ranged from $\{1, 2, 3, 4, 5\}$ while **f** was chosen from $\{1, 2, 4, 6, 8, 9, 10\}$.

At the end, few words should be written about text processing techniques from Fig. 1:

- **NLTKProcessor** - this it the main text corpora processor aiming at text normalization and noise removal. It applies the following operation: removing of urls and hashtags, word to lower conversion along with filtering punctuation and English stop words. At the end, each Tweet is tokenized with part of speed tagging using WordNetLemmatizer.
- **TfidfVectorizer** - normalized text can be transformed into feature vector. The easiest way is to count words and take the occurrence score. The main drawback with this method is connected with fact that longer message will obtain higher average score values than shorter one even if it contains the same topic. More valuable features can be extracted using Term Frequency times Inverse Document frequency approach. It controls weights assigned to words by dividing the number of occurrences of each word in a document by the total number of words in the document. Furthermore, global corpora text parameters are used to modify weights for words that occur in many documents in the corpus and are therefore less informative than those that occur only in a smaller portion of the corpus.

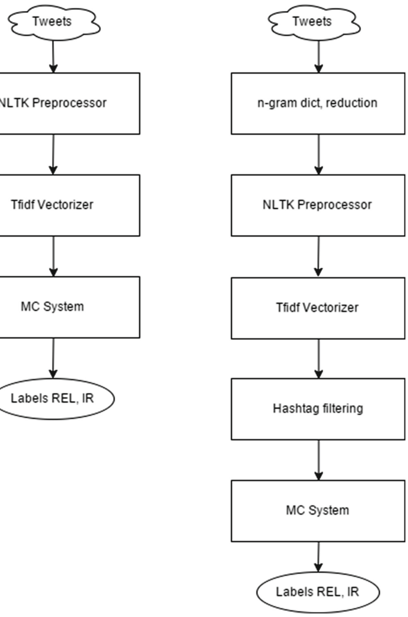


**Fig. 1.** Processing diagram for two proposed methods. Method 1 on the left, Method 2 on the right.

### 3.2 Tweet Classification Results

We have divided experiments into two stages. In the first one, to validate proposed processing pipeline we have extracted 50k Tweets from our collected database. Each Tweet from this set was labeled independently by two person as related (*REL*) or irrelevant (*IR*) to flu and its symptoms. For further classification we have taken data with an agreement between assessors. For extracting training and testing sets, we have used 5 × 2 cross-validation method [3]. Table 1 contains averaged values after 10 runs. Apart from accuracy ratio, it presents

**Table 1.** Tweet classification results averaged after 10 runs. Notation used in table is as follows: **A** - accuracy ratio, **P** - precision ratio, **R** - recall ratio, **SingleBest** - single best classifier from ensemble, **Method 1** - classification performed only with preprocessed text, **Method 2** - classification performed with both preprocessed text and hashtags analysis. Small indices in **A** columns indicate algorithms to which compared one is statistically significantly better.

| **Method 1** | | | | | | **Method 2** | | |
|---|---|---|---|---|---|---|---|---|
| **SingleBest**[1] | | | **Ensemble**[2] | | | **Ensemble**[3] | | |
| **A** | **P** | **R** | **A** | **P** | **R** | **A** | **P** | **R** |
| $0.828^{-}$ | 0.829 | 0.824 | $0.847^{\mathbf{1}}$ | 0.847 | 0.847 | $0.878^{\mathbf{1,2}}$ | 0.876 | 0.872 |

**Table 2.** Settings for the second experiment where batch classification is tested. Notation used in table is as follows: $IR_d$, $REL_d$ stand for the number of Tweets irrelevant and related to flu topic and labeled by experts; $R_d$, $R_c$ are ratio score calculated for true labels and classified by MC system Tweets, respectively. $= -$ in the last row indicate if obtained classification is the same or statistically significantly different than true labels given by experts. For this purpose we have used statistical test comparing two proportions $\frac{REL_d}{REL_d+IR_d}$, $\frac{REL}{REL+IR}$.

| Batch | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $IR_d$ | 400 | 350 | 400 | 320 | 390 | 400 | 300 | 150 | 100 | 320 | 390 |
| $REL_d$ | 100 | 150 | 100 | 180 | 110 | 100 | 200 | 350 | 400 | 180 | 110 |
| $R_d$ | 0.20 | 0.30 | 0.20 | 0.36 | 0.22 | 0.20 | 0.40 | 0.70 | 0.80 | 0.36 | 0.22 |
| $R_c$ | $0.17^=$ | $0.26^=$ | $0.18^=$ | $0.31^=$ | $0.19^=$ | $0.18^=$ | $0.36^=$ | $0.64^-$ | $0.74^-$ | $0.34^=$ | $0.18^=$ |

also precision and recall values for each algorithm and two preproposing methods related to converting text corpora into feature set. Furthermore, for pairwise algorithm comparison we have used 5 × 2 F-Test to check if one algorithm is significantly better/worse or equivalent to the other compared. The level of $p < 0.05$ was considered as statistically significant. After analyzing results in Table 1, it is clear that Ensemble system with **Method 2** gives the best classification results. Preliminary results are promising, but we should explain why proposed algorithm does not obtain the highest possible classification score. We have seen that Tweets classified by MC system as related to flu contained some specific keywords that can be associated with flu, but their assigned label is *IR* because they are commercials or only article headlines. In the future, for better topic detection, we should also take into account message intention and information if it is strictly related to the author or describes only a general knowledge or basic fact. In the second experiment we have extracted 11 batches and each set contained 500 Tweets. The number of *REL* and *IR* in each set is determined beforehand and presented in Table 2. For this experiment we have used Ensemble[3] with **Method 2** for converting text into feature space. For training phase, 1000 Tweets were randomly selected from database after removing Tweets from testing batches. In the future, proposed system can be used for detecting events in *Twitter* online platform when classification for predefined topic is rising. There are few methods for detecting events in the upcoming data. For this purpose we can calculate ratio (1) and react when it is above a predefined threshold. This approach can be illustrated in Fig. 2, where $R_c > 0.5$ can be treated as signaling threshold. What is more, this figure indicates also that even if our proposed model is not optimal it allows to follow general trends in messages and detect events related to flu. As an alternative to constant alarm threshold, we can apply the heuristic approach in which the ratio of Tweets classified from the previous batch is compared with the ratio from the current batch. In this case, statistical tests comparing two proportions can be used to check whether there is a sufficient evidence at the $\alpha = 0.05$ level, to conclude that the two populations

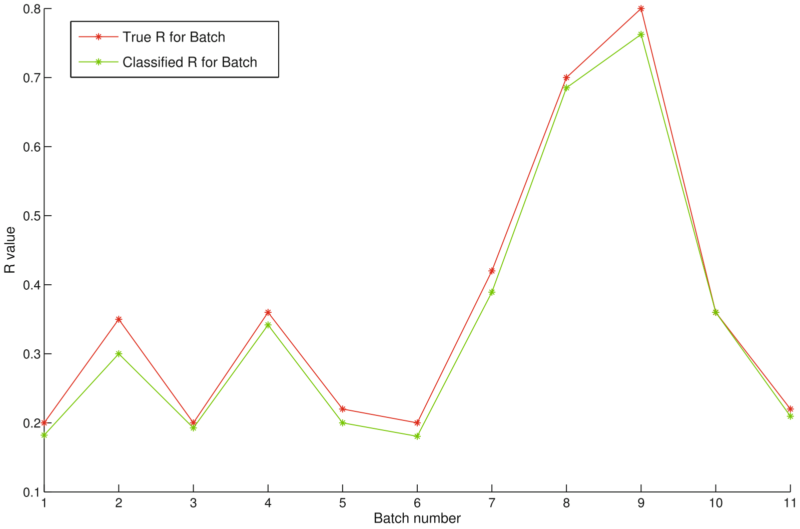


**Fig. 2.** R ratio for provided 11 batches. Red color stands for true $R_d$ calculated according to labels assigned by experts while blue one is $R_c$ score obtained from classification.

(batches from previous and current processing) differ significantly with respect to their classification as Tweet related to flu.

$$R = \frac{IR}{IR + REL} \tag{1}$$

## 4 Conclusions

In this paper, a framework for Tweet filtering was proposed which uses ensemble of classifiers for detecting event related to flu and its symptoms. For evaluation purposes 50 k messages were extracted from our database and labeled independently as related to flu or not by two experts. This approach allowed us to check classification accuracy of constructed ensemble and indicate which parameters are crucial in Tweet processing phase. For event detection and monitoring one should use knowledge from both *Twitter* text message and commonly used hashtags. On the other hand, another important aspect should be taken into account. We have noticed that for such topics as health (especially flu related) commercial Tweets are a major problem deteriorating the final classification. To combat this problem, deeper and thorough sentiment analysis should be performed. In the future we would like to improve our framework with adding a rejection mechanism for detecting if Tweet is only related to flu topic or author personally has a strong connection with flu. What is more, we would like to detect more events in upcoming Tweets so instead of classification, clustering mechanism will be applied.

**Acknowledgments.** This work was supported by EC under FP7, Coordination and Support Action, Grant Agreement Number 316097, ENGINE - European Research Centre of Network Intelligence for Innovation Enhancement (http://engine.pwr.wroc.pl/). This was also supported by the statutory funds of Department of Systems and Computer Networks, Wroclaw University of Technology. All computer experiments were carried out using computer equipment sponsored by ENGINE project.

## References

1. http://scikit-learn.org/
2. Aiello, L.M., Petkos, G., Martin, C., Corney, D., Papadopoulos, S., Skraba, R., Goker, A., Kompatsiaris, I., Jaimes, A.: Sensing trending topics in twitter. IEEE Trans. Multimedia **15**(6), 1268–1282 (2013)
3. Alpaydin, E.: Combined 5 × 2 cv F test for comparing supervised classification learning algorithms. J. Neural Comput. **11**, 1885–1892 (1999)
4. Atefeh, F., Khreich, W.: A survey of techniques for event detection in twitter. Comput. Intell. **31**(1), 132–164 (2015)
5. Batista, L.B., Ratte, S.: A multi-classifier system for sentiment analysis and opinion mining. In: Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012), pp. 96–100. IEEE Computer Society, Washington, DC (2012)
6. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O'Reilly Media (2009)
7. Cavalin, P.R., Moyano, L.G., Miranda, P.P.: A multiple classifier system for classifying life events on social media. In: ICDM Workshops, pp. 1332–1335. IEEE Computer Society (2015)
8. Celikyilmaz, A., Hakkani-Tur, D., Feng, J.: Probabilistic model-based sentiment analysis of twitter messages, pp. 79–84. IEEE (2010)
9. Joachims, T.: Text categorization with suport vector machines: learning with many relevant features. In: Proceedings of the 10th European Conference on Machine Learning, ECML 1998, pp. 137–142. Springer, London (1998)
10. Kaleel, S.B., Abhari, A.: Cluster-discovery of twitter messages for event detection and trending. J. Comput. Sci. **6**, 47–57 (2015)
11. Kwak, H., Lee, C., Park, H., Moon, S.: What is Twitter, a social network or a news media? In: WWW 2010: Proceedings of the 19th International Conference on World Wide Web, NY, USA, pp. 591–600. ACM, New York (2010)
12. Lamb, A., Paul, M.J., Dredze, M.: Separating fact from fear: tracking flu infections on twitter. In: NAACL (2013)
13. Salathe, M., Khandelwal, S.: Assessing vaccination sentiments with online social media: implications for infectious disease dynamics and control. PLOS Comput. Biol. **7**(10), 1–7 (10 2011)
14. Zubiaga, A., Spina, D.: Martínez, R., Fresno, V.: Real-time classification of twitter trends. J. Assoc. Inf. Sci. Technol. **66**(3), 462–473 (2015)

# Real-Time Image Content Assessment for Underwater Robot Manoeuvring Based on Structural Tensor Analysis

Jakub Nawała(✉) and Bogusław Cyganek

AGH University of Science and Technology,
al. Mickiewicza 30, 30-059 Krakow, Poland
jakub.tadeusz.nawala@gmail.com

**Abstract.** The paper presents an efficient method for real-time image analysis for manoeuvring of the underwater robot. Image analysis is done after computing the structural tensor components which unveil rich texture and texture-less areas. To allow a power efficient underwater operation in real-time the method is implemented on the Jetson TK1 self-standing graphics card using the CUDA compute architecture. The laboratory experimental results show that the system is capable of processing about 40 Full HD images per second while allowing orientation toward texture specific regions for obstacle avoidance.

**Keywords:** Robot manoeuvring · Structural tensor · Underwater operation · CUDA · Real-time programming

## 1 Introduction

Robot manoeuvring and more specifically, obstacle avoidance, is one of the most fundamental functions that must be implemented first for any self-standing exploratory machine to operate properly. Given the operating conditions, solution providing this functionality must perform efficiently both in terms of power- and computational throughput efficiency. Thus, no conventional central processing unit (CPU) may be used. What is more, as will be later shown, mobile counterparts of CPUs proof insufficient for more complex image processing operations like this one. This necessitates utilisation of hardware accelerated mobile platform. For this purpose, NVIDIA Jetson TK1 board was chosen as it provides sufficient computational power, at the same time, retaining low energy consumption figures. Advantageous features of Jetson come from the presence of the first mobile, CUDA capable Tegra K1 graphics processing unit (GPU) on-board. Video camera based system for underwater manoeuvring is given as a solution to the problem of growing need for autonomous inspections of maritime facilities. Amongst others, appearance of this phenomenon was described in [5]. It is important to underline that usage of standard camera, instead of sonar system, constitutes much cheaper approach that additionally allows to process data

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_48

of much higher spatial resolution. On the other hand, requiring sufficient water transparency. In order to identify rich texture and texture-less regions, structural tensor (ST) calculus is used. Procedure is based on assumption that in underwater environment any textured area represents potential obstacle. Using ST of the input image, it is possible to define highly coherent patches providing information about the presence of nearby objects. Solution based on GPU was chosen due to its remarkable performance characteristics and its potential applications for execution time reduction. Latter of those was proven to be true by many scientific publications. For example, work of Wetzl *et al.* [10] shows GPU accelerated super-resolution (SR). Utilisation of hardware acceleration allowed authors to use their solution in interactive applications like image guided surgery. Similar approach was proposed in [8], which is another solution to SR problem based on bilateral total variation filtration. Apart from super resolving low resolution images, GPU may be used to enhance object detection algorithms, like the one presented by Chikkerur [3]. In this application, it allowed to achieve up to 10 fold performance boost. Selection of NVIDIA Jetson TK1 as the main processing platform for self-standing application is natural due to its low power consumption. When measured under normal workloads, it oscillates around 4 W. For more computationally demanding operations, it reaches its maximum at 11.5 W. Combining NVIDIA Jetson TK1 properties, allowing power efficient real-time image processing, and capability of ST algorithm to extract potential obstacles in underwater environs, it was decided to design, develop and test GPU-based solution for real-time image content assessment for underwater robot manoeuvring based on structural tensor analysis. Reminder of this document consists of 4 sections. Section 2 provides more details on image content analysis based on structural tensor calculus. Section 3 includes overview of system architecture used and Sect. 4 presents experimental results achieved. Section 5 concludes the paper giving both qualitative and quantitative analysis of the work done.

## 2 Image Analysis with the Structural Tensor

Most general execution flow of an algorithm used to perform image analysis with the structural tensor may be found in Fig. 1. Input gray-scale image is first used to calculate 3 components of its structural tensor. Those are later used to find corresponding coherence image. Then, coherence image is summed in square blocks of the given size and resultant lower resolution frame becomes an input for thresholding operation. At last, binary image is generated with white pixels denoting rich texture regions that represent potential obstacles in underwater environment.

### 2.1 Computing the Structural Tensor

A structural tensor $\mathbf{T}$ can be computed in each local and compact neighbourhood $R(\mathbf{x}_0)$ around a point $\mathbf{x}_0$ of a 2D image $I$, as follows [2,4,6]

$$\mathbf{T}\left(\mathbf{x}_0\right) = B_{R(\mathbf{x}_0)}\left(\mathbf{G}\mathbf{G}^T\right), \tag{1}$$

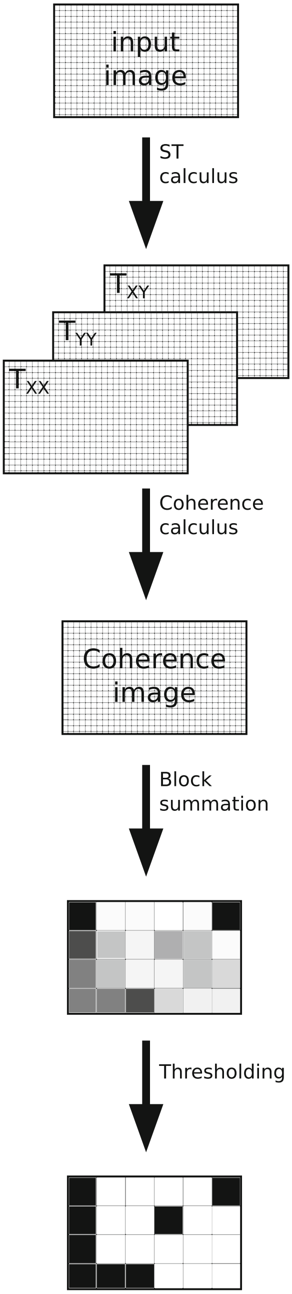


**Fig. 1.** Diagram of execution flow of image analysis algorithm

where $B_{R(\mathbf{x}_0)}$ denotes an averaging operator in a region $R$, with the centre at the point $\mathbf{x}_0$, and $\mathbf{G}$ is a gradient vector at *each* point $\mathbf{x}$ of R, that is, $\mathbf{x}\in R(\mathbf{x}_0)$. The gradient $\mathbf{G}$ at a point $\mathbf{x}$ of $I$, is defined as follows

$$\mathbf{G}\left(\mathbf{x}\right) = \begin{bmatrix} \frac{\hat{\partial}}{\hat{\partial} x} I\left(\mathbf{x}\right) \\ \frac{\hat{\partial}}{\hat{\partial} y} I\left(\mathbf{x}\right) \end{bmatrix} = \begin{bmatrix} I_x\left(\mathbf{x}\right) \\ I_y\left(\mathbf{x}\right) \end{bmatrix}, \tag{2}$$

where $I_x(\mathbf{x})$ and $I_y(\mathbf{x})$ denote discrete spatial derivatives of $I$ at the point $\mathbf{x}$, in the horizontal $(x)$ and vertical $(y)$ directions, respectively. In the presented system $G_{R(\mathbf{x}_0)}$ is a simplest discrete binomial filter [4]. It can be easily noticed that $\mathbf{T}(\mathbf{x}_0)$ constitutes a symmetric positive 2D matrix with elements describing averaged values of the gradient components in a certain neighbourhood around a point $\mathbf{x}_0$. Thus, the structural tensor $\mathbf{T}$ brings information on intensity changes not only at a point $\mathbf{x}_0$ but, more interestingly, also in its nearest neighbourhood. It is worth noticing, that when $\mathbf{T}$ is computed at each point $\mathbf{x}$ of $I$, then each $\mathbf{T}(\mathbf{x})$ conveys information on overlapping regions around the point $\mathbf{x}$. Hence, it contains information on image texture and local curvature. Inserting (2) into (1), the following is obtained

$$\mathbf{T} = B_R\left(\begin{bmatrix} I_x \\ I_y \end{bmatrix}\begin{bmatrix} I_x \; I_y \end{bmatrix}\right) = B_R\left(\begin{bmatrix} I_xI_x \; I_xI_y \\ I_yI_x \; I_yI_y \end{bmatrix}\right) = \begin{bmatrix} T_{xx} \; T_{xy} \\ T_{yx} \; T_{yy} \end{bmatrix}. \tag{3}$$

To simplify the equations, in the aforementioned derivations, the point $\mathbf{x}$ was omitted since averaging by the filter $B_R$ is over a set of points in $R$, as already described. Concluding, the structural tensor $\mathbf{T}$ is built from four components: $T_{xx}$, $T_{yy}$, $T_{xy}$ and $T_{yx}$, each of those being averaged multiplication of partial derivatives of input image's intensity signal with respect to horizontal and vertical spatial coordinate.

### 2.2 Texture Analysis for Robot Manoeuvring

In order to practically verify the concept of robot manoeuvring based on texture analysis, consideration done in this subsection will be accompanied by the presentation of execution steps outcomes for real underwater image acquired (see Fig. 2). As it was already shown, first step is calculation of structural tensor of the input image. Result of performing this operation may be found in Fig. 3. It should be noted that rich texture regions can be roughly estimated this early in the processing procedure, using simple visual inspection. 3 components of ST are now used to calculate *coherence* parameter for each pixel of the input image. Formula used to do it is given in Eq. 4.

$$coherence = \frac{(T_{xx} - T_{yy})^2 + (2T_{xy})^2}{(T_{xx} + T_{yy})^2} \tag{4}$$

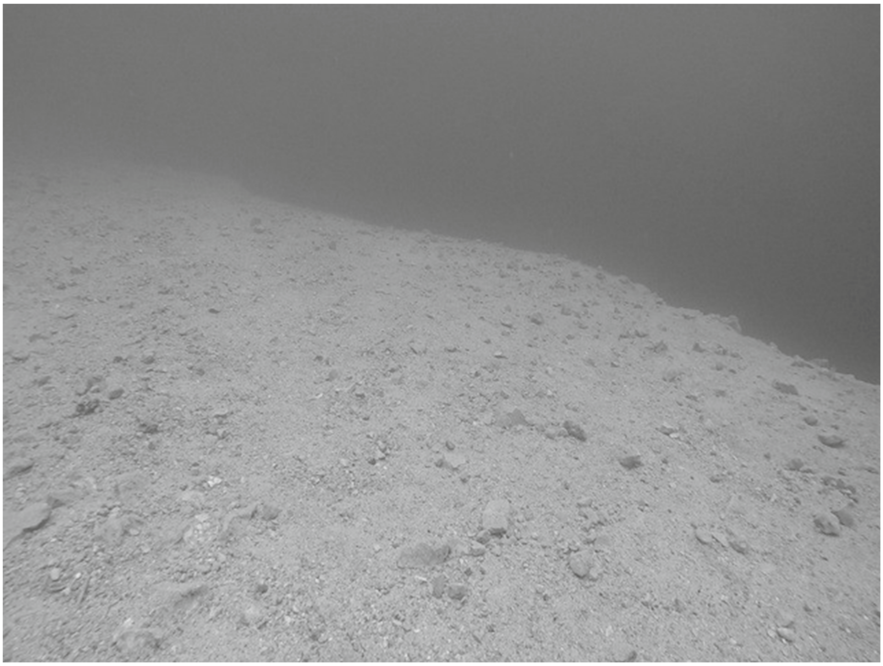
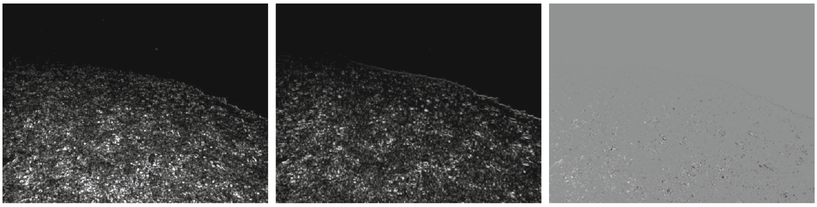


**Fig. 2.** Input gray-scale image captured in Zakrzówek lake at depth of 15 m

Calculated values are now summed in square blocks of arbitrary size. For images shown in this paper, block size of 32 was chosen, as it provides most satisfactory end results. As a last step, block-summed coherence image is filtered with thresholding filter, which sets to maximum all the pixels being larger than the middle point between largest and smallest value of the block. This operation may be expressed by the following equation:

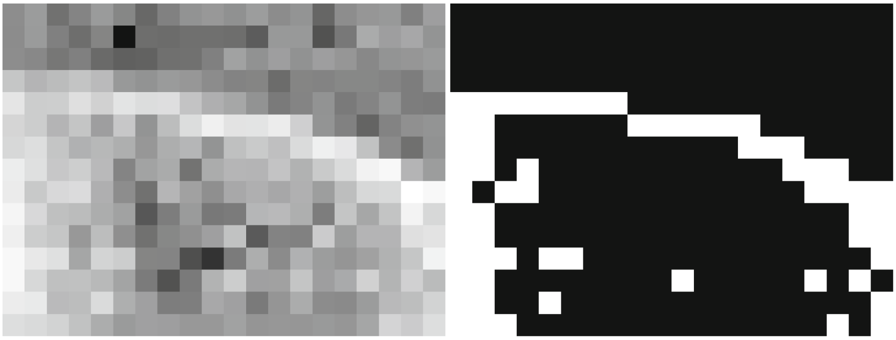
$$b(x,y) = \begin{cases} 1, & \text{when } C_{block}(x,y) > \frac{max(C_{block})-min(C_{block})}{2} \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where $b(x,y)$ represents pixel in output binary image at horizontal coordinate $x$ and vertical coordinate $y$, and $C_{block}$ represents block-summed coherence image. Final binary image juxtaposed with block-summed coherence frame may be seen in Fig. 4. Willing to further proof proper operation of the procedure proposed, second underwater input image and its final binary image generated is given in Fig. 5. As it can be judged by visual inspection, solution given shows strong potential for application in underwater robot manoeuvring. Resultant binary images yield good approximation of obstacles appearing on the path of the exploratory machine and as such, may be used to avoid unwanted collisions.



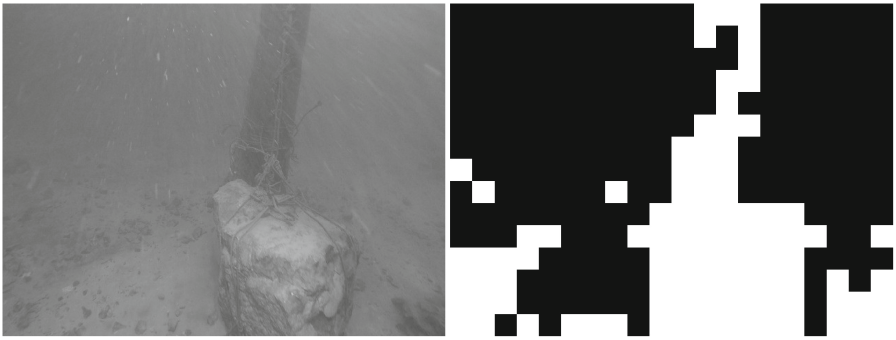
(a) $T_{xx}$ (b) $T_{yy}$ (c) $T_{xy}$

**Fig. 3.** Structural tensor components calculated



(a) Summed in blocks (b) After thresholding

**Fig. 4.** Coherence image summed in blocks and after thresholding



(a) Input image (b) Binary map

**Fig. 5.** Second input image from Zakrzówek lake and resultant binary map

## 3 System Architecture

Proposed image content analysis for underwater robot manoeuvring was implemented as a software application. Its general operation flow was given in Fig. 1. Most of the components were implemented on CPU and hardware acceleration was utilised only for computationally intensive structural tensor computation. As for any GPU accelerated image processing task, CUDA threads were mapped onto input image's pixels. Architecture of this mapping may be found in Fig. 6. It should be pointed out that each block of threads is dedicated for a single image row. Number of 384 threads per block was chosen due to its best performance characteristics for tested use-cases. In case of images wider than 384 pixels, single thread from a given block is used to serve pixels at rows indexes being multiplication of its original position. However, not every thread must process equal number of pixels and hence, images of all possible widths may be used. In addition to proper CUDA threads utilisation, texture memory of GPU was used to assure best possible computational throughput. Its presence is justified as almost any image processing operation shows strong spatial locality and this type of memory is optimised just for those kinds of tasks.

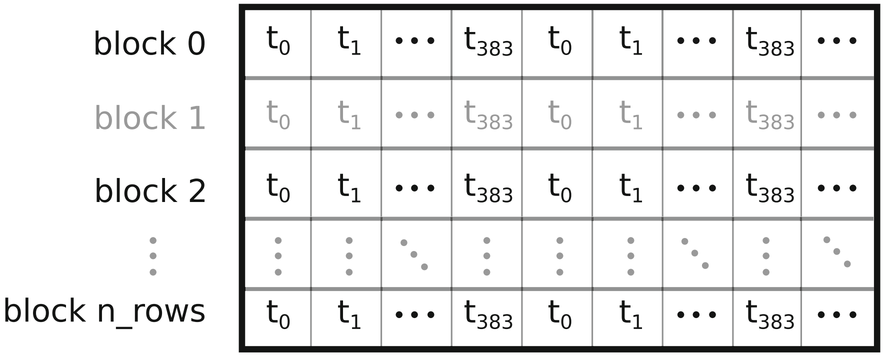


**Fig. 6.** Mapping of CUDA threads over input image's pixels ($t_n$ represents n-th thread and *n_rows* means number of input image's rows)

## 4 Experimental Results

As execution time constitutes the crucial factor in real-time applications, it was the main scope of the experimental tests conducted. More specifically, execution time of the most computationally intensive step of the algorithm proposed was measured. In this case, it was the structural tensor computation that was consuming major part of processing resources. For better visualisation of GPU predominance over standard CPU implementations, tests were done on 3 platforms, namely: Jetosn's Tegra K1 GPU, Jetson's ARM Cortex A-15 mobile CPU and standard Intel Xeon X5650 CPU. Comparison with Cortex A-15 is justified as it posses mobile properties necessary for self-standing applications. On the other hand, Xeon X5650 is given to show how mobile GPU is capable of outperforming more standard processing platforms, when used in image processing tasks. In order to verify what is the influence of input image size, measurements were performed for 3 standard resolutions: VGA (640 × 480), HD (1280 × 720) and Full HD (1920 × 1080). Additionally, results were averaged over 5 iterations to assure measurements stability. Figure 7 shows execution times achieved. Detailed information about the results may also be found in Table 1. It is readily visible that Tegra K1 GPU outperforms both CPUs. What is more, its superiority becomes more relevant for higher resolutions. At Full HD, GPU executes almost 20 times faster, as compared to ARM Cortex A-15, and more than 11 times for Xeon X5650. On top of that, Tegra K1 is capable of processing up to 40 Full HD frames per second, being the figure sufficient to treat this application the real-time solution. It is worth mentioning that good performance scaling versus input image size for GPU, comes from its massively parallel architecture. It provides sufficient number of processing resources dedicated for each pixel, even for resolutions as high as 2 MP. Following this consideration, it can be assumed that there is some theoretical boundary for input image size, which makes GPU execution time rise faster than for 3 presented cases. Nevertheless, good performance figures for resolutions up to 2 MP proofs to be sufficient for most practical applications.

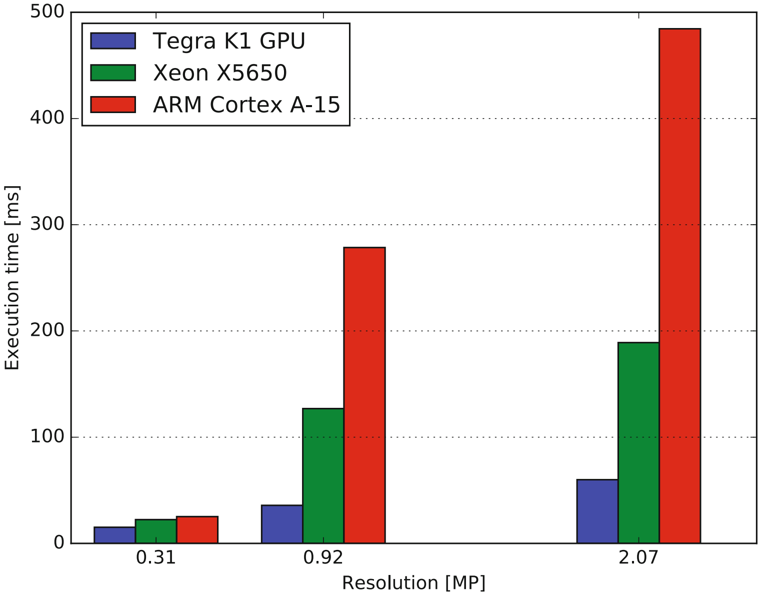


**Fig. 7.** Comparison of ST computation execution time when ran on Tegra K1 GPU, ARM Cortex-A15 mobile CPU or Xeon X5650 standard CPU

**Table 1.** ST computation frames per second throughput comparison for selected input image resolutions (MP stands for *megapixel* meaning 1 million pixels); additionally compared with average power consumption

| Image resolution | Frames per second | | |
|---|---|---|---|
| | Tegra K1 GPU | Xeon X5650 | ARM Cortex A-15 |
| VGA (0.31 MP) | 65 | 27 | 16 |
| HD (0.92 MP) | 44 | 7 | 5 |
| Full HD (2.07 MP) | 39 | 3 | 2 |
| Avg. power cons. [W] | 5 [7] | 95 [1] | 0.6 [9] |

## 5 Conclusions

In this paper, usability of GPU-based platform, combined with structural tensor calculus, was investigated as the mean to perform real-time image content analysis for underwater robot manoeuvring. Given the laboratory execution time results, it was proven that NVIDIA Jetson TK1, with Tegra K1 GPU on-board, does constitute efficient platform for real-time image processing. Additionally, taking into account its mobile properties, it was shown to be a good solution for computer vision provision in self-standing exploratory machines. On top of that, structural tensor calculus was presented as a tool for extraction of information about nearby obstacles in underwater environs. It is important to point out that lower power consumption figures of on-board computer vision provider means not only better utilisation of resources, but also possibility to use the for other application-critical tasks. Having verified Jetson TK1 serviceableness for real-time image analysis and given structural tensor calculus functionality for obstacle avoidance, it is concluded that solution proposed, may be used in potential submarine exploratory application. It is thus planned to practically verify this assumption.

**Acknowledgement.** This work was done as a realisation of the research project financed by The Polish National Science Centre (NCN) and has the following project number UMO-2014/15/B/ST6/00609.

## References

1. Intel xeon x5650 cpu: Review, benchmarks, and specs (2010), http://processors.specout.com/l/230/Intel-Xeon-X5650
2. Bigün, J., Granlund, G.H., Wiklund, J.: Multidimensional orientation estimation with applications to texture analysis and optical flow. IEEE Trans. Pattern Anal. Mach. Intell. **13**(8), 775–790 (1991). http://dblp.uni-trier.de/db/journals/pami/pami13.html#BigunGW91
3. Chikkerur, S.: Cuda implementation of a biologically inspired object recognition system (2008). http://code.google.com/p/cbcl-model-cuda/

4. Cyganek, B.: An Introduction to 3D Computer Vision Techniques and Algorithms. John Wiley & Sons (2009)
5. Jacobi, M.: Autonomous inspection of underwater structures. Robot. Auton. Syst. **67**(C), 80–86 (2015). http://dx.doi.org/10.1016/j.robot.2014.10.006
6. Jahne, B.: Digital Image Processing: Concepts, Algorithms, and Scientific Applications, 4th edn. Springer-Verlag New York Inc., Secaucus (1997)
7. Kelion, L.: Ces 2014: Nvidia tegra k1 offers leap in graphics power (2014). http://www.bbc.com/news/technology-25618498
8. Mitzel, D., Pock, T., Schoenemann, T., Cremers, D.: Video super resolution using duality based TV-L1 optical flow, pp. 432–441. Springer, Heidelberg (2009). http://dx.doi.org/10.1007/978-3-642-03798-6_44
9. Shimpi, A.L.: The arm vs x86 wars have begun: In-depth power analysis of atom, krait & cortex a15 (2013). http://www.anandtech.com/show/6536/arm-vs-x86-the-real-showdown
10. Wetzl, J., Taubmann, O., Haase, S., Köhler, T., Kraus, M., Hornegger, J.: GPU accelerated time-of-flight super-resolution for image-guided surgery. In: Tolxdorff, T., Deserno, T.M. (eds.) Bildverarbeitung für die Medizin, pp. 21–26 (2013). http://www5.informatik.uni-erlangen.de/Forschung/Publikationen/2013/Wetzl13-GAT.pdf

# Distributed DBSCAN Algorithm – Concept and Experimental Evaluation

Adam Merk(✉), Piotr Cal, and Michał Woźniak

Department of Systems and Computer Networks,
Wrocław University of Science and Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{Adam.Merk,Michal.Wozniak}@pwr.edu.pl

**Abstract.** One of the most popular clustering algorithm is DBSCAN, which is known to be efficient and highly resistant to noise. In this paper we propose its distributed implementation. Distributed computing is a very fast growing way of solving problems in big datasets using a multinode cluster, rather than parallelization in one computer. Using its features in proper way, can lead to higher performance and, what is probably more important, higher scalability. In order to show added value of this way of designing and implementing algorithms we compare our results with GPU parallelization. On the basis of the obtained results We formulate the propositions how to improve our solution.

**Keywords:** Distributed computing · Clustering · Unsupervised learning · Big data

## 1 Introduction

Nowadays, clustering is one of the most popular technique for data analysis and it is being used in many fields, as image analysis, data mining, data compression or bioinformatics. Clustering, as a process, is a way of retrieving some information about the data and its structure [1]. However, there can be some problems in this task, such as: how to define similarity of instances, how to estimate number of clusters, which we want to obtain, how to deal with noisy datasets and finally, how to get better performance and scalability. Nowadays we have many capabilities and algorithms, which can resolve above problems, but on the other hand, we live in a world of data growing in a fast pace to an enormous size. In this paper, we propose solution based on a distributed computing for DBSCAN algorithm.

DBSCAN (density-based spatial clustering of applications with noise) was proposed in 1996 and is one of the most popular density-based clustering algorithms [2]. The main advantages of the density-based methods are that, they do not have to know the expected number of clusters, they work well in a noisy datasets and they are quite simple. In real-world data noise is an unavoidable

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_49

problem [3] and may have two main sources: implicit errors from measurement tools and random errors from processes or experts in data gathering (e.g. in document digitalization process) [4]. Noise, with high probability, will affect the output of algorithm and computing time [5]. Of course, in clustering process it does not have such a big impact as in classification, but still the results will be erroneous. Dealing with noise is, naturally, one of the tasks of data pre-processing, but it can be time- and resource-consuming. The way in which DBSCAN is dealing with this problem will be described in more details later in this paper. However, it is a very simple method, which makes the computational complexity relatively low.

As it was said before, DBSCAN is not a very complicated algorithm. The main idea is to check whether in a given distance ($\varepsilon$) from every point(instance) in a dataset is enough points (*minPts*) to create a dense region. If condition is met, the cluster is created and it is further expanded if points in its $\varepsilon$-neighborhood also meet condition (have enough points in their $\varepsilon$ distance). In the worst case, algorithm defined this way has runtime complexity of $O\left(n^2\right)$, because for every instance in dataset, DBSCAN has to find all objects within its $\varepsilon$-neighborhood, what means we need to visit every point in dataset. There are some possibilities to reduce this complexity, such as using an indexing structure (for example R*-Tree) or materializing the distance matrix, what results in shorter computation time, but increases memory complexity. Taking into consideration all of the above and a fact that nowadays we gather a lot of data, which may have large dimensionality, the algorithm can run slow in a standard implementation.

There are a few ways to improve DBSCAN, such as parallel implementations as presented in [6–9] or preprocessing with GPU [1], where different parallel computation concepts were used, but the implementations use only one physical computer. In works [6–9] the authors used master-slave technique, where data is divided between processors, then every processor locally performs DBSCAN algorithm and finally the results were merged. To obtain better performance, authors used mentioned earlier R*-Tree structure, special data-partition algorithm or disjoint-set data structure. Authors of [1] presented a slightly different approach, in which they use GPU, which "*consists of hundred or thousand small cores which can be used for general purpose computation (GPGPU)*". Each of the above implementations has, however, scalability limited by the number of processors, cores or memory. Nowadays, we can achieve better (or even infinite) scalability by using distributed computing.

Dealing with massive data is a well-known problem and there are a lot of methods of analyzing and processing such a big datasets. One way to deal with this problem is using reduced dataset (especially in classification problems). Using this method can reduce the computation time and size of database. [16] However, in clustering problem we can not reduce the dataset. In this case one of the most popular and efficient way is to use Apache Hadoop [11,12] with MapReduce [10] or Apache Spark [13,14]. MapReduce is a designed by Google and is a framework for, so called, big data processing. Main objective of its implementation was to deliver to a user a simple way of programming in distributed environment, such as cluster of computers. MapReduce consists of two steps: **Map** and **Reduce**. In **Map** step the data is read from the HDFS (*Hadoop Distributed File System*) and transformed. Each node in a cluster is responsible for this process and uses a different (one or more) partitions of dataset. Then, in **Reduce** step, the results from every node are send to the reductor and merged into one result through an user-defined function. In most cases reductor is one, main node, but in more complex problems, we can have more than one reduce operation and therefore we can also have more than one reductor. It is possible to use MapReduce without an Apache Hadoop, but Hadoop is an open-source and one of the most popular implementations of this framework. However, using Hadoop is in many cases not a good solution, because it is based on disk operations and they are not very efficient. The solution to this problem is using Apache Spark, which main advantage over Hadoop is that it is using in-memory computation. The drawback is that we need a lot of RAM if we want to make computations on very large data. However, Spark is designed to overcome Hadoop's performance and it has shown to perform faster than Hadoop. The idea of how the Spark is working is similar to Hadoop. The main program (called the driver) controls workers and collects results from them, same as reducer. Nodes read data from the HDFS to memory, performs some computations and save the result to disk. The main difference between Hadoop and Spark is that Spark keeps the data in memory until the processes on workers are done, so querying it repeatedly is more efficient than reading it from the disk every time (RAM has better answer time than read operation from HDD).

Using above solutions for distributed computing seems like natural way of achieving better scalability and performance in DBSCAN algorithm. There are two main objectives of this paper: the first one is to verify, whether distributed implementation will obtain at least the same times of computing and exactly the same results as GPU parallelization proposed in [1] and the second one is to obtain better scalability. In the next section we shortly describe DBSCAN algorithm and its distributed implementation, indicated also the problems to be solved in further works. Then we show and discuss experiment results and, finally, the last section concludes the work and formulates recommendation for further works.

## 2 Algorithm

### 2.1 DBSCAN

DBSCAN is a density-based algorithm, where user needs to define two parameters: $\varepsilon$ and minimum number of points to create a cluster - *minPts*. In the first step random point (P) is selected and algorithm computes distance between this point and every other point in dataset. It is worth mentioning, that nothing stands in the way to define a distance function as an input parameter of algorithm, but in this work we follow originally proposed algorithm [2] and define this function inside algorithm. If the distance between two points in dataset is equal or less than $\varepsilon$, this points become neighbors. If in $\varepsilon$-neighborhood of P is at least as many points as we defined as *minPts* (it is called a dense region), the cluster is created. If the condition of *minPts* is not met, it means there is less points in $\varepsilon$-neighborhood of P, all points are labeled as noise. However, these noise points can be found later in sufficiently sized $\varepsilon$-neighborhood of a different point and become a part of another cluster. If a cluster has started, the next step is to verify if algorithm can expand this cluster or should go to next point outside this cluster. This is done simply by checking if distance and *minPts* condition is met for every point (P') within this cluster. If verification is positive (condition is fulfilled) algorithm expands this cluster for every point in $\varepsilon$-neighborhood of P'. If the cluster is expanded to maximum size, it means we cannot find more points that are in dense region of cluster, the expansion is finished and every point is labeled as visited. Then algorithm randomly selects another point in dataset, which is not labeled as visited and repeat the procedure. If there is no more not visited points, algorithm stops. Pseudocodes of DBSCAN algorithm and expanding function are presented below.

**Algorithm 1.** DBSCAN algorithm

```
Data:
Dataset - D,
distance - $\varepsilon$,
minimum number of points to create dense region - minPts
begin
    $C \longleftarrow 0$
    for each point P in dataset D do
        if P is visited then
            Continue to next P
        end
        else
            mark P as visited
            nbrPts $\longleftarrow$ points in $\varepsilon$-neighborhood of P (distance function)
            if sizeof(nbrPts) < minPts then
                mark P as NOISE
            end
            else
                $C \longleftarrow NewCluster$
                Call Expand Cluster Function(P, nbrPts, C, minPts)
            end
        end
    end
end
```

## 2.2 Distributed Computing

Algorithm described above is a classic implementation and needs some modifications to work in distributed environment. We can run it in such environment and we will probably get better results, but just because we will use more RAM

**Algorithm 2.** Expand Cluster Function

```
Data:
Point in dataset - P
Neighbor points - nbrPts
Current cluster - C
distance - $\varepsilon$,
minimum number of points to create dense region - minPts
begin
    add P to cluster C
    for each point P' in nbrPts do
        if P' is not visited then
            mark P' as visited
            nbrPts' ⟵ points in $\varepsilon$-neighborhood of P' (distance function)
            if sizeof(nbrPts') $\geq$ minPts then
                nbrPts ⟵ nbrPts + nbrPts'
            end
        end
        if P' is not a member of any cluster then
            add P' to cluster C
        end
    end
end
```

or CPU. Proposed modifications are based on paper [1], because GPU parallelization is the closest equivalent of distributed computing. As in the mentioned paper, we also need to create a neighborhood matrix to compute results, and what is more important, to send them later to reducer. We assume that each node has the whole dataset available and computes distances to every point in dataset, but not for every point. In other words, using distributed computing we will make only $\frac{1}{N}$ computing on each node to receive the same results, where N is the number of nodes in cluster. Downside of this solution is that we demand that whole dataset is available for every node at anytime. Therefore, it is necessary to implement fully distributed implementation, where dataset is divided among nodes in cluster. Idea is to build cluster independently on every node (in **Map** phase) and then collect the results by using special function (in **Reduce** phase). Function collecting results works in the same way as DBSCAN algorithm, with this difference that we measure distance (and density) based on boundary points of every cluster. If boundary point of one cluster is in a dense region of boundary point of second cluster, then those clusters should be merged. For Map phase we use standard implementation of DBSCAN because, all we need to do is to load data into RAM and compute clusters. Pseudocode for reduce step is presented below.

This implementation have some drawbacks, which will be described wider in conclusions of this paper.

**Algorithm 3.** Reduce step for Distributed DBSCAN

```
Data:
Clusters collected from nodes - CC,
distance - $\varepsilon$,
minimum number of points to create dense region - minPts
begin
    for each cluster C in CC do
        for each point P in C do
            if P is visited then
                Continue to next P
            end
            else
                mark P as visited
                nbrPts $\longleftarrow$ points in $\varepsilon$-neighborhood of P that are not in C
                if sizeof(nbrPts) $\geq$ minPts then
                    Merge C with every cluster, to which points nbrPts belongs
                end
            end
        end
    end
end
```

## 3 Experiment

There are two main goals of the experiment. First one is to verify the possible speedup of distributed implementation versus parallelization on GPU proposed in [1]. Second one is to verify whether the proposed implementation causes higher scalability of algorithm.

### 3.1 Set-Up

For the purpose of first part of experiment, we use the same artificial generated dataset as in [1] (artificial dataset). It contains 20 000 observations and two dimensional continuous feature. For the second experiment, we use Higgs dataset [15], which contains 11 millions instances and 8 features. The difference is that we recomputed algorithm in different machine, to get results comparable with distributed implementation. For GPU parallelization test were carried out on NVIDIA GeForce GTX 970, Intel Core i7 CPU and 16GB RAM (denoted GeForce+i7), while for distributed implementation we used 10 node cluster with 25vcores i5 processor and 256GB RAM on each Node. The time was always measured for GPU parallelization and distributed computing.

### 3.2 Results

Before we can compare computing times of both algorithms, we had to check whether both algorithms give exactly the same results. Figure 1 below shows the visualization of artificial dataset.

The output of both algorithms were exactly the same, so we can compare time of computing. For this purpose we compute distributed implementation for 2, 4, 8 and 10 nodes and GeForce+i7. Figure 2 shows the result for this comparison.

As we can see in Fig. 2 distributed implementation does not get much better times. In fact, for 2 Node configuration we get worse results in this dataset and for the rest configurations we can see slightly better results but only when we reach size of about 15 000 objects. Of course, if we look at 10 Node configuration (bottom line) we see that for 20 000 objects we get about 3 times speedup. However the most important thing in this figure is the shape of series. We can see that time for GPU parallelization starts to grow very fast from 5 000 objects to 20 000, from about 0.15 s to almost 3.5 s. We can also see that for the same objects increase 10 Node configuration time grows only from about 0.11 s to 0.9 s.

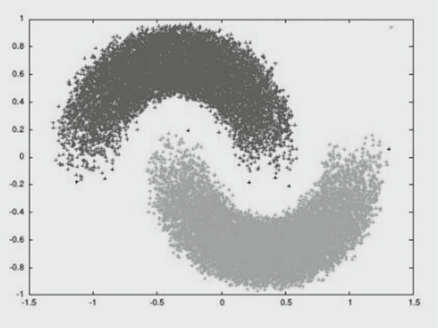


**Fig. 1.** Artificial dataset for first experiment

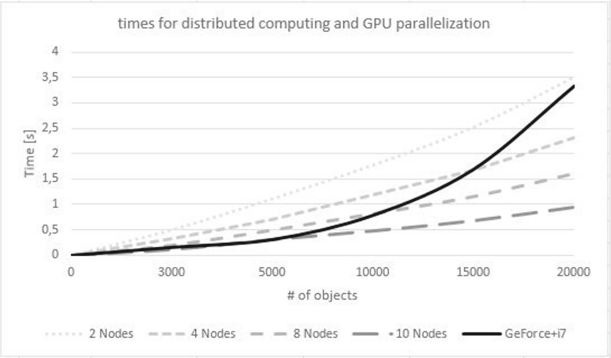


**Fig. 2.** Time comparison for distributed computing and GPU - first dataset

This suggests that for much bigger dataset, GPU parallelization will not get results in acceptable times (or in worst case will not get any results at all). That is why we have made second experiment. Dataset we choose have 11 millions of instances and 28 float attributes.

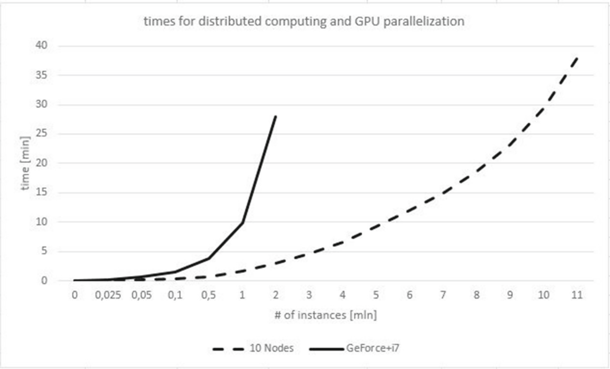


**Fig. 3.** Time comparison for distributed computing and GPU - Higgs dataset

Figure 3 shows times of computing for much bigger set. First important thing is that, GPU parallelization could not handle bigger set than 2 millions of instances because of not enough space in GPU memory and RAM. Second conclusion is that times for bigger set are much better for distributed implementation. For half a million of instances distributed implementation is over 4 times faster and gives results in about 44 s, while GPU parallelization gives the same results in about 230 s. For one million of instances distributed implementation is about 6 times faster (1.5 min versus 10 min) and for 2 millions times are about 10 times better for distributed computing. Then we cannot compare results, because as it was said earlier, GPU parallelization cannot return any results.

## 4 Conclusions

Distributed computing seems to be the best option for getting better results and scalability for DBSCAN algorithm, but only for bigger datasets. As we showed, for small datasets using such algorithm is not good. This is because results are very similar and we must remember that using distributed computing is connected in most cases with using Hadoop and MapReduce or Spark, so we need additional time to make our environment ready. For example, Spark needs about 4 s to start its instances. Therefore, if we have relatively small dataset on which DBSCAN can be computed in less than 10 s, we should use not distributed algorithms. What is important, scalability of distributed computing also is not endless. However, suggested implementation of distributed DBSCAN has, some space for optimization. First of all, we need to consider optimization of **Reduce** phase, because in this implementation, in worst case we will visit every point of cluster again. There is a relatively easy way to handle this. At the **Map** phase we need to return not only clusters but also set of boundary points. Using those sets we can verify much faster, whether point is in dense region of other cluster. Most speedup which we get in this implementation comes from divided dataset in map phase, but we lose a lot of time in reduce phase. Another thing is, that we can use some distributed data structure to speed up this algorithm and that is probably the best way to consider for further works.

## References

1. Cal, P., Woźniak, M.: Data preprocessing with GPU for DBSCAN algorithm. In: Proceedings of the 8th International Conference on Computer Recognition Systems, CORES 2013, pp. 793–801 (2013)
2. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis, E., Han, J., Fayyad, U.M. (eds.) Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD 1996), pp. 226–231. AAAI Press (1996)
3. Wang, R.Y., Storey, V.C., Firth, C.P.: A framework for analysis of data quality research. IEEE Trans. Knowl. Data Eng. **7**, 623–640 (1995)
4. Zhu, X., Wu, X.: Class noise vs attribute noise: a quantitative study. Artif. Intell. Rev. **22**, 177–210 (2004)
5. Wu, X., Zhu, X.: Mining with noise knowledge: error-aware data mining. IEEE Trans. Syst. Man Cybern. **38**, 917–932 (2008)
6. Brecheisen, S., Kriegel, H.-P., Pfeifle, M.: Parallel density-based clustering of complex objects. In: PAKDD 2006, pp. 179–188. Springer, Heidelberg (2006)
7. Li, H., Chen, M., Gao, X.: Parallel dbscan with priority r-tree. In: Information Management and Engineering (ICIME) (2010)
8. Patwary, M.A., Palsetia, D., Agrawal, A., Liao, W.-K., Manne, F., Choudhary, A.: A new scalable parallel dbscan algorithm using the disjoint-set data structure. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC 2012, pp. 62:1–62:11. IEEE Computer Society Press, Los Alamitos (2012)
9. Xu, X., Jäger, J., Kriegel, H.-P.: A fast parallel clustering algorithm for large spatial databases. Data Min. Knowl. Discov., 263–290 (1999)
10. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. In: OSDI 2004, pp. 137–150 (2004)
11. White, T.: Hadoop, The Definitive Guide. O'Reilly Media Inc. (2012)
12. Apache Hadoop Project, "Apache Hadoop" (2016). http://hadoop.apache.org/, Accessed December 2016
13. Karau, H., Konwinski, A., Wendell, P., Zaharia, M.: Learning Spark: Lightning-Fast Big Data Analytics. O'Reilly Media, Incorporated (2015)
14. Spark, A.: Lightning-fast cluster computing, "Apache Spar" (2016). https://spark.apache.org/, Accessed December 2016
15. Baldi, P., Sadowski, P., Whiteson, D.: Searching for exotic particles in high-energy physics with deep learning. Nat. Commun. **5**, 2 July 2014
16. Porwik, P., Doroz, R.: Self-adaptive biometric classifier working on the reduced dataset. In: 9th International Conference on Hybrid Artificial Intelligence Systems (HAIS), Salamanca. Spain Book Series, LNCS, vol. 8480, pp. 377–388 (2014)

# Combining Active Learning and Self-Labeling for Data Stream Mining

Łukasz Korycki[1] and Bartosz Krawczyk[2(✉)]

[1] Department of Systems and Computer Networks, Wrocław University of Science and Technology, 50-370 Wrocław, Poland

[2] Department of Computer Science, Virginia Commonwealth University, Richmond, VA 23284, USA

bkrawczyk@vcu.edu

**Abstract.** Data stream mining is among the most vital contemporary data science challenges. In this work we concentrate on the issue of actual availability of true class labels. Assumption that the ground truth for each instance becomes known right after processing it is far from being realistic, due to usually high costs connected with its acquisition. Active learning is an attractive solution to this problem, as it selects most valuable instances for labeling. In this paper, we propose to augment the active learning module with self-labeling approach. This allows classifier to automatically label instances for which it displays the highest certainty and use them for further training. Although in this preliminary work we use a static threshold for self-labeling, the obtained results are encouraging. Our experimental study shows that this approach complements the active learning strategy and allows to improve data stream classification, especially in scenarios with very small labeling budget.

**Keywords:** Machine learning · Data stream mining · Active learning · Semi-supervised learning · Self-labeling

## 1 Introduction

Data streams are dynamic collections of data. Here, we assume that instances arrive continuously over time, we need to process them within a limited time as soon as they become available and that the volume of data is potentially unbounded. They became a significant part of the so-called big data era, linking themselves to $5V$'s describing these massive collection of information: Volume, Velocity, Veracity, Variety and Value. Especially first two of these characteristics are important from the streaming perspective. Volume is connected to ever-growing size of the incoming data. Velocity described that instances arrive over time, often in high-speed and with varying frequencies. Additionally, we must assume a potential presence of a phenomenon known as concept drift that may influence the stream characteristics over time [7]. These factors make canonical

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_50

machine learning algorithms incapable of efficient stream processing, as they were designed for static cases.

In data stream mining, one may see four main trends among methods for handling non-stationary data. Drift detectors are external algorithms that monitor certain properties of the stream and correlate changes in them to the incoming change [15]. This allows to train new classifier on most recent instances and replace the old one with it when the degree of change has become too severe. Sliding window solutions are based on keeping a predefined buffer of instances that are used for the classification process [13]. Window is constantly updated with new instances and old ones are discarded in order to keep a track on the stream progress. On-line classifiers are able to process the stream instance by instance, using a limited amount of computational resources [6]. Each instance is processed once and discarded, thus leading to a reduced memory usage. Finally, ensemble approaches utilize a number of classifiers together with one of the previous solutions [8]. Due to their flexible structure they can easily accommodate for drifts in streams, as well as offer improved predictive efficacy. All of these methods must be characterized by a limited computational resource consumption, therefore implementing them in high-performance environments is becoming more and more popular [5].

Despite a high number of algorithms proposed for solving this problem, most of them work under a naive assumption that the ground truth for each instance becomes known right after processing it. In real-life scenarios we have limited access to labels, as labeling is connected with certain cost (e.g., using expert's knowledge). Therefore, we must wisely use the scarce budget that is under our disposal. This has lead to developing unsupervised [1], semi-supervised [9] and active learning solutions [12] in order to minimize the need for labeled instances.

In this paper, we propose to combine active learning strategy with self-learning approach. Self-learning is a branch of semi-supervised learning that allows classifiers to label an instance if they display a high certainty regarding its class [11]. It has been shown to display good results in stationary scenarios, therefore we decided to investigate its usefulness for non-stationary data streams. As we assume to work under a highly limited budget, each additional information that may improve the performance of classifier without imposing any cost is highly valuable. Therefore, we allow the classification system to to update itself with instances for which it displays high certainty, without a label query. Of course, we are aware of the fact that an error in self-labeling may propagate itself along with the stream. However, we will show that an interplay between active learning and self-labeling leads to potential improvements in quality of classifiers trained from scarcely labeled streams. This is a preliminary work, where we propose to use static self-learning with no adaptation to potential changes in data. However, results on a number of real-life streams prove that, despite its simplicity, such an approach may be beneficial to learning systems.

## 2 Combining Active Learning and Self-labeling

We present a hybrid approach to data streams classification. It combines two mentioned strategies working under budget constrains: active learning [14] and self-labeling [11]. These methods are dedicated to the environments with a vast amount of unlabeled samples and only few labeled. They make use of annotated data to utilize objects without labels. Both aim to reduce the overall cost of labeling process, but they do it in different ways.

The general active learning framework has been presented in [12]. For each incoming sample $X$ an estimation of current label spending $\hat{b}$ is checked. If it does not exceed a given budget, there is a chance for querying the sample, using a chosen strategy. Otherwise, the sample is discarded. In our framework we add a self-labeling strategy, which tries to classify discarded objects on its own and use them for a training without any additional cost. We assume that the semi-supervised learning step will, in some cases, adjust a model more precisely. For now, we do not use any concept drift detector and adapt the classifier in a blind way [7]. The hybrid algorithm is presented below.

**Algorithm 1.** The general hybrid algorithm which combines active learning and self-labeling

```
Data: labeling budget $B$, QueryStrategy ($parameters$),
      SelflabelingStrategy ($parameters$)
Result: classifier $L$ at every iteration
Initialization: $\hat{b} \leftarrow 0$
repeat
    receive incoming instance $X$;
    if $\hat{b} < B$ then
        if QueryStrategy ($X, parameters$) $= true$ then
            request the true label $y$ of instance $X$;
            update labeling expenses $\hat{b}$;
            update classifier $L$ with $(X, y)$;
        else if SelflabelingStrategy ($X, parameters$) $= true$ then
            get a class label $\hat{y}$ predicted by classifier $L$;
            update classifier $L$ with $(X, \hat{y})$;
        end
    end
until stream ends;
```

For the active learning strategy we choose uncertainty sampling [2]. This group of methods is probably the most intuitive approach to querying the unlabeled data. The idea is to ask only for these objects about which a classifier is very uncertain. It relates to a distance from a decision boundary in the analyzed problem space. Objects which are close to the boundary are potentially more ambiguous than these which are far from it. According to the Bayesian Decision Theory, for a given object $X$, a classifier will assign low posterior probability

$p(y|X)$ to each class $y$ whose region co-creates the boundary. The indications for all classes can be combined using, for example, the maximum a posterior (the lower value the higher uncertainty) or an entropy measure (the higher value the higher uncertainty) [10]. The easiest algorithm uses a fixed threshold $\theta \in (0, 1\rangle$ to decide if an object should be queried, using the formula:

$$p(\hat{y}|X) \leq \theta. \tag{1}$$

In fact, such solution will be sufficient only for the changes that appears near the decision boundary and it will not satisfy the basic requirements for an active learning strategy [12]. More effective solutions have been proposed and exhaustively described in [12]. We incorporate one of them – a variable threshold with randomization. In this method the threshold $\theta$ is modified by a factor $s \in (0, 1\rangle$. The uncertainty value is calculated using the maximum rule.

**Algorithm 2.** The variable threshold with randomization strategy

```
Data: incoming instance $X$, trained classifier $L$, threshold adjustment
      step $s \in (0, 1\rangle$, variance of the threshold randomization $\sigma$
Result: $labeling \in \{true, false\}$
Initialization: $\theta \leftarrow 1$, store the latest value during operation
$\hat{y} \leftarrow \arg\max_y p(y|X)$, where $y \in \{1, ..., c\}$;
$\theta_r \leftarrow \theta \times \eta$, where $\eta \in N(1, \sigma)$ is a random multiplier;
if $p(\hat{y}|X) \leq \theta_r$ then
    decrease the uncertainty region $\theta \leftarrow \theta(1 - s)$;
    return $labeling \leftarrow true$
else
    increase the uncertainty region $\theta \leftarrow \theta(1 + s)$;
    return $labeling \leftarrow false$
end
```

The dynamic threshold value increases $\theta \leftarrow \theta(1+s)$ when a model becomes more stable and less samples tend to give low enough confidence to be queried. The threshold decreases $\theta \leftarrow \theta(1 - s)$ when a potential concept drift appears and more samples become ambiguous. The first step spreads the region of interest, the second one prevents the budget from being exhausted during the drift. To ensure that samples, which are even further from the decision boundary, will be queried regularly enough, the randomization of $\theta$ is added. It is performed using the normal distribution $N(1, \sigma)$, where $\sigma$ is a variance of the randomization. Such approach should quiet effectively react to changes in any part of the problem space and manage the budget rationally.

Our self-labeling module is based on a confidence measure. The self-labeling strategy performs unsupervised learning only for these object which a classifier can classify with a high confidence. Similar measures, as for the uncertainty sampling, can be used [11]. For decision making we incorporate the simplest strategy that uses a fixed confidence threshold $\gamma \in (0, 1\rangle$ and the maximum rule for combining the posterior probabilities. It decides basing it on:

$$p(\hat{y}|X) \geq \gamma. \tag{2}$$

It is worth saying that the chosen strategies and their measures are, in fact, complementary. The algorithm will ask an expert for very uncertain data samples and, obviously, will not use them for a self-training. On the other hand, it will discard the objects for which it will be very confident and will use them for the training without additional supervision.

## 3 Experimental Study

In this section we present experiments which has been conducted to evaluate the proposed hybrid framework. First, data streams are described. Then the evaluation methodology is introduced. Finally, we present obtained results and discuss them briefly.

### 3.1 Datasets

The algorithm has been tested on real data streams. Since it is dedicated to evolving environments, datasets should contain some concept drifts. All of the presented streams are supposed to be non-stationary, however, it is not directly determined what kind of drift will occur and when it will happen. It can be assumed, taking into consideration an environment or a context from which the data streams have been gathered. We use four different real-life stream datasets (their details are given in Table 1):

- **Sensor**[1] – this data stream was collected from 54 environmental sensors set in Intel Berkeley Research Lab. They were used to record such factors like temperature, humidity or light for 2 months. A class label for each row is a sensor ID. During a day, conditions in rooms were changing, so was the underlying data distribution.
- **Cover Type**[2] – this dataset consists of predictions of forest cover types, based on cartographic variables and some remotely sensed attributes. The analyzed areas are located in the Roosevelt National Forest of northern Colorado. A concept drift is supposed to appear, due to some seasonal, incremental changes.
- **Spam**[3] – it is a textual data stream created on the basis of Spam Assasin Collection, with features extracted using the bag of words procedure and then selected according to their informative value. It is stated that the stream's concept changes gradually, due to spam content and filters adaptation.

[1] http://www.cse.fau.edu/~xqzhu/stream.html.

[2] http://archive.ics.uci.edu/ml.

[3] http://mlkd.csd.auth.gr.

- **Airlines**[4] – it is a data stream with flight delays predictions. They are based on, among others, a flight source and destination, its length, date and an airline. The stream probably consists of some seasonal, incremental changes, due to different weather conditions, and abrupt drifts caused by, for example, airplanes accidents.

**Table 1.** Summary of the used data streams.

| Name | Instances | Attributes | Classes | Type |
|---|---|---|---|---|
| Sensor | 2 219 803 | 5 | 54 | Numeric |
| Cover type | 581 012 | 54 | 7 | Numeric, Nominal |
| Spam | 9324 | 499 | 2 | Textual |
| Airlines | 539 383 | 7 | 2 | Numeric, Nominal |

### 3.2 Set-Up

In streaming environments it is impractical to use such demanding evaluation methods like cross validation. It is commonly known to be a very reliable approach, however, for continuous and massive data streams it is very time-consuming. Some methods dedicated to the on-line evaluation, for example hold-out, may be insufficient for evolving data streams. We use a simple, yet effective method called interleaved test-then-train with a sliding window $\omega$ for the most recent samples [3]. In this approach each sample is used for testing and later for updating a model. The accuracy measure is incrementally recalculated. It provides a relatively good measurement sensitivity, but it highly depends on the optimal window size. We choose $\omega = 1000$ according to the results from [3]. The on-line accuracy rate is defined as a ratio of correctly classified objects within the window. We also calculate the overall accuracy achieved on all data samples.

In our experiments, we compare the hybrid framework (SL) with the original active learning algorithm (AL), which uses the variable threshold with randomization. We use Hoeffding Adapative Tree [4] as the base classifier. The fixed confidence threshold was arbitrarily set to $\gamma = 0.9$, the variable threshold to $s = 0.01$ and the standard deviation of randomization to $\sigma = 1$. We compare the frameworks for different budget values $B = \{1\%, 5\%, 10\%, 20\%, 50\%\}$, which is a factor of samples queried for the true labels.

### 3.3 Results and Discussion

Table 2 presents averaged accuracies over the entire stream for examined AL and SL strategies. Detailed depictions of changes in accuracies following the stream progress are presented in Figs. 1, 2, 3, and 4, for each dataset respectively. For clarity of presentation, each figure shows only three budget settings: 1%, 10% and 50%.

[4] http://moa.cms.waikato.ac.nz/datasets.

**Table 2.** Overall accuracies of AL and SL given a budget.

| Dataset | Strategy | $B = 1\%$ | $B = 5\%$ | $B = 10\%$ | $B = 20\%$ | $B = 50\%$ |
|---|---|---|---|---|---|---|
| Sensor | AL | 8% | 13% | 18% | **37%** | 48% |
| | SL | **10%** | **15%** | **24%** | 34% | **56%** |
| Cover type | AL | 60% | **66%** | **73%** | **76%** | **80%** |
| | SL | **65%** | 61% | 65% | 68% | 78% |
| Spam | AL | 28% | 55% | 75% | 82% | **89%** |
| | SL | **76%** | **86%** | **78%** | **83%** | 85% |
| Airlines | AL | **60%** | **60%** | **60%** | **62%** | **63%** |
| | SL | 56% | 57% | 56% | 59% | 62% |

As we can see from the results combining active learning with self-labeling is able to work well in three out of four streams. Only for the Airlines dataset we observe a drop in accuracy regardless of the budget size being evaluated. This can be contributed to high variance or noise in incoming data that results in classifier updating itself with incorrect content, thus leading to a drop in accuracy.

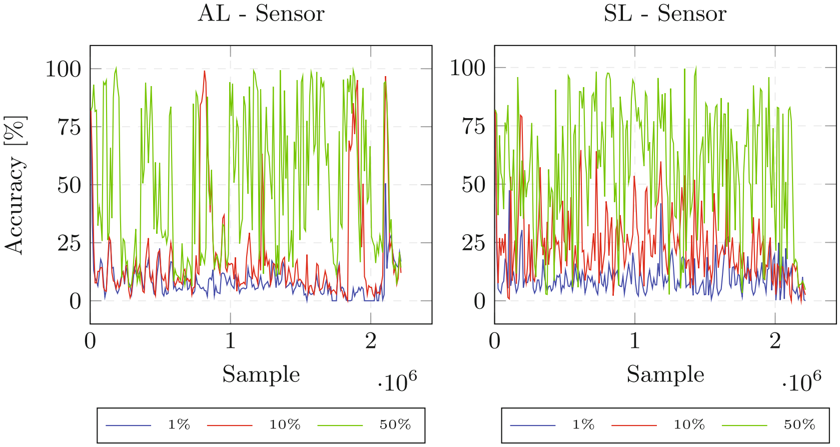


**Fig. 1.** Accuracy of AL and SL given a budget on Sensor data stream.

For Cover Type dataset, we may observe that SL strategy is able to improve upon standard AL solution only for $B = 1\%$, yet offering a significant gain in accuracy of 5%. Although fact that it works well only for one budget setting may seem discouraging at first, one must notice that small budgets are actually the more realistic ones. When assuming that one is being able to label 30% or 50% of instances in the entire stream, it is hard to really consider high labeling cost.

In real scenarios, where a company would need to employ an expert to provide labels, budgets on the level of 1% or even smaller are much more probable. This additionally show that for such limited budgets AL cannot efficiently query enough samples and each additional information significantly contributes to the classification performance. This allows to back-up our claim that SL will prove their usefulness in such difficult conditions.

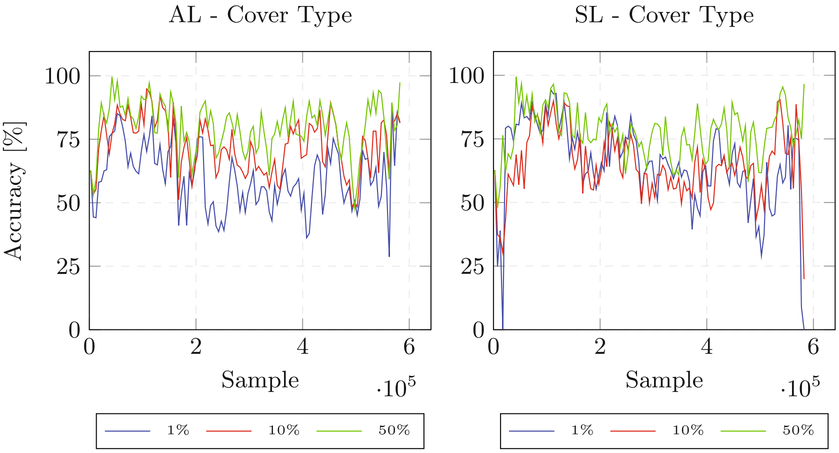


**Fig. 2.** Accuracy of AL and SL given a budget on Cover Type data stream.

For the remaining two datasets (Sensor and Spam), we are able to observe gains in accuracy due to using SL for most of budget sizes. This is especially vivid in case of Spam data, where we are able to gain 48% of accuracy for budget $B = 1\%$. We connect this performance with the nature of analyzed streams.

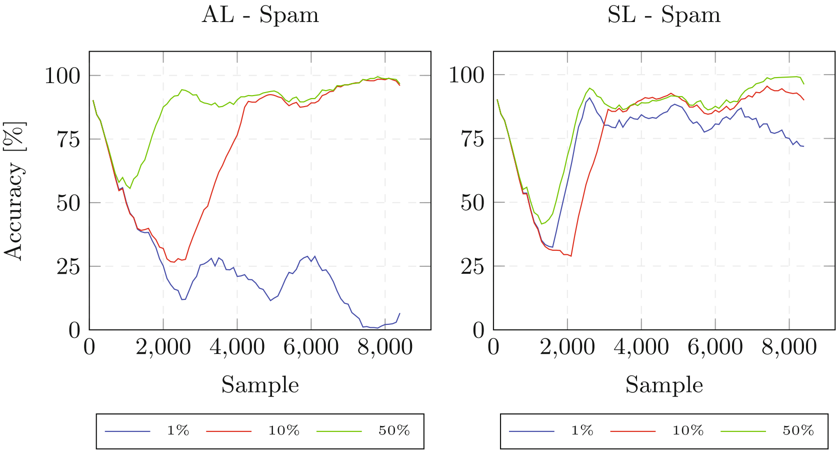


**Fig. 3.** Accuracy of AL and SL given a budget on Spam data stream.

In real-life situations the drift is not ever-present, but appears in intervals. Therefore, in stationary parts of the stream AL itself is not able to provide enough instances for the classifier to achieve stable adaptation. Therefore, by providing larger set of training instances, we are able to better recover after drift and adapt to the current concept. This allows us to conclude that there is a need for further developing combination of AL and SL approaches that are able to switch between stationary and non-stationary parts of the processed streams.

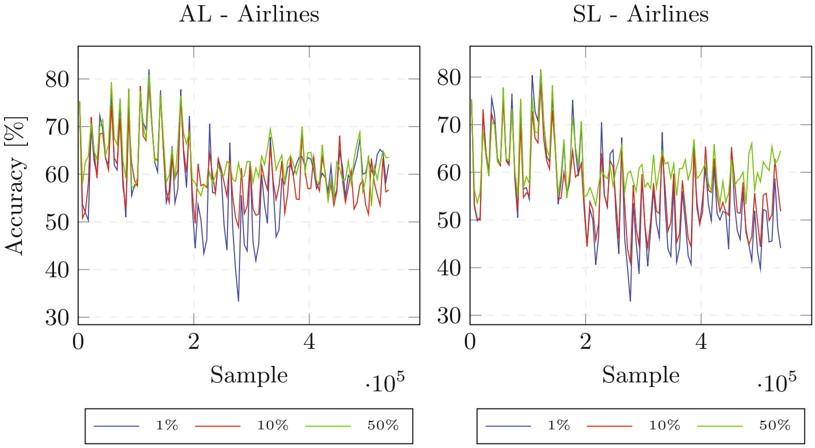


**Fig. 4.** Accuracy of AL and SL given a budget on Airlines data stream.

## 4 Conclusions and Future Works

In this paper, we have addressed the issue of learning from data streams under constrained budget. We discussed realistic scenario, in which one does not have access to class labels for all of incoming instances and can label only a small subset of them. We proposed to augment the active learning strategy with a self-labeling approach that allowed for classifier to automatically label and utilize for training instances characterized by a high certainty. As a result, our classifier was able to increase the number of utilized instances for adapting to non-stationary streams with no direct cost. Our strategy was complementary to active learning, as it could select most uncertain instances for expert labeling and most certain ones for automatic labeling. In this work we have used a simple static threshold for self-labeling procedure that despite its simplicity was able to improve the performance of classifiers when working under a very small budget. Preliminary results obtained encourage us to develop more advance self-labeling strategies for drifting data streams that will take into account the dynamic and non-stationary nature of incoming concepts.

**Acknowledgments.** This work was partially supported by the Polish National Science Center under the grant no. DEC-2013/09/B/ST6/02264.

## References

1. Abdallah, Z.S., Gaber, M.M., Srinivasan, B., Krishnaswamy, S.: Anynovel: detection of novel concepts in evolving data streams. Evolving Syst. **7**(2), 73–93 (2016)
2. Aggarwal, C.C., Kong, X., Gu, Q., Han, J., Yu, P.S.: Active learning: a survey. In: Data Classification: Algorithms and Applications, pp. 571–606 (2014)
3. Bifet, A., de Francisci Morales, G., Read, J., Holmes, G., Pfahringer, B.: Efficient online evaluation of big data stream classifiers. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 59–68 (2015)
4. Bifet, A., Gavaldà, R.: Adaptive Learning from Evolving Data Streams, pp. 249–260 (2009)
5. Cano, A., Zafra, A., Ventura, S.: Parallel evaluation of pittsburgh rule-based classifiers on gpus. Neurocomputing **126**, 45–57 (2014)
6. Czarnecki, W.M., Tabor, J.: Online extreme entropy machines for streams classification and active learning. In: Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015, Wroclaw, Poland, 25–27 May 2015, pp. 371–381 (2015)
7. Gama, J.A., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM Comput. Surv. **46**, 1–37 (2014)
8. Krawczyk, B., Minku, L.L., Gama, J., Stefanowski, J., Woźniak, M.: Ensemble learning for data stream analysis: a survey. Inf. Fusion **37**, 132–156 (2017)
9. Nguyen, H., Ng, W.K., Woon, Y.: Concurrent semi-supervised learning with active learning of data streams. Trans. Large-Scale Data Knowl.-Centered Syst. **8**, 113–136 (2013)
10. Settles, B.: Active learning literature survey. Computer Sciences Technical report. University of Wisconsin-Madison (2009)
11. Triguero, I., García, S., Herrera, F.: Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Knowl. Inf. Syst. **42**(2), 245–284 (2015)
12. Žliobaitė, I., Bifet, A., Pfahringer, B., Holmes, G.: Active learning with drifting streaming data. IEEE Trans. Neural Netw. Learn. Syst. **1**, 27–39 (2014)
13. Woźniak, M.: A hybrid decision tree training method using data streams. Knowl. Inf. Syst. **29**(2), 335–347 (2011)
14. Woźniak, M., Ksieniewicz, P., Cyganek, B., Kasprzak, A., Walkowiak, K.: Active learning classification of drifted streaming data. In: International Conference on Computational Science 2016, ICCS 2016, 6–8 June 2016, San Diego, California, USA, pp. 1724–1733 (2016)
15. Woźniak, M., Ksieniewicz, P., Cyganek, B., Walkowiak, K.: Ensembles of heterogeneous concept drift detectors - experimental study. In: Computer Information Systems and Industrial Management - 15th IFIP TC8 International Conference, CISIM 2016, Vilnius, Lithuania, September 14-16, 2016, Proceedings, pp. 538–549 (2016)

# Drift Detection Algorithm Using the Discriminant Function of the Base Classifiers

Robert Burduk(✉)

Department of Systems and Computer Networks,
Wroclaw University of Science and Technology,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
robert.burduk@pwr.edu.pl

**Abstract.** Recently, several approaches have been proposed to deal with the concept drift detection. In this paper we propose the new concept drift detection algorithm based on the decision templates. The decision templates are obtained from the outputs of the base classifier that form an ensemble of classifiers. Experiments on several publicly available data sets verify the effectiveness of the proposed algorithm.

**Keywords:** Drift detection · Multiple classifier system · Decision templates

## 1 Introduction

Many approaches in machine learning assume that training data have a stationary source. Unfortunately, such an assumption is often violated in real-world dynamic environments. To deal with this problem, many classification methods of data stream were proposed. Data streams are characterized by huge volumes of records fast changes and the frequent need for the quick, real-time response or analysis [7]. Therefore, the characteristics of data streams are often not stable but are change in time. This problem is defined as the concept drift. There are three possible kinds of the drift: conditional, feature, or dual drift [9]. In the case of the conditional drift a change occurs in a priori probability of classes. In the case of the feature drift class conditional probabilities can change, but in the dual drift class conditional probabilities and a priori probability of classes may change. Apart from the differences in the cause and effect of concept changes, there are several ways of how such changes occur. The discussion of this problem can be found in [4,5]. The text is organized as follows: after this introduction, in Sect. 2 the idea of drift detection is presented. Section 3 contains the description of the proposed method for detecting the concept drift. The experimental results on several data sets are presented in Sect. 4. Finally, conclusions from the experiments and future research proposals are presented.

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_51

## 2 Related Works

The drift detector is created in order to detect the concept drift and alarm the base classifier that its model should be rebuilt or updated. Below, we discuss the recently proposed method to detect the drift. The Drift Detection Method (DDM) proposed by Gama et al. uses the Binomial Distribution [7]. In each iteration the online classifier calculates the decision class which is either true or false. So, for the set of examples an error is identified from Bernoulli trials. For each example in data stream we have to update two registers to keep track of error rate. First is $p_{min}$ and the second is denoted $s_{min}$. These two are used to identify the warning level condition and the alarm level condition. DDM works best on data streams with the sudden drift as gradually changing concepts can pass without triggering the alarm level. EDDM is a modification of DDM proposed by Baena-Garcia et al. [2]. In this algorithm the same warning-alarm mechanism is used as the one recommended in DDM. However, it uses the distance error rate instead of classifier's error rate. EDDM works better than DDM for the slow gradual drift, but it is more sensitive to noise. Another drawback is that it considers the thresholds and searches for the concept drift when a minimum of 30 errors have occurred. ADWIN - Bifet et al. proposed the method which uses sliding windows of variable size, that are recomputed online according to the rate of the change detected from the data in these windows [3]. The window (W) is dynamically enlarged when there is no clear modification in the context, and it shrinks when a modification is detected. ADWIN works only for one-dimensional data. A separate window must be maintained for each dimension, for n-dimensional raw data, which results in handling more than one window. The Paired Learners, proposed by Stephen Bach et al., uses two learners: stable and reactive [1]. The stable learner makes predictions based on all of its experience, while the reactive one predicts basing on a window of recent examples. It uses the interplay between these two learners and their accuracy differences to cope with the concept drift. Ross et al., proposed the drift detection method based on Exponentially Weighted Moving Average (EWMA) [1], which is used for identifying an increase in the mean of a sequence of random variables. In EWMA, the probability of incorrectly classifying an instance before the change point and the standard deviation of the stream are known. The design of a change detector is a compromise between detecting true changes and avoiding false alarms. This is accomplished by carrying out statistical tests that verify if the running error or class distribution remain constant over time. For numeric sequences the first proposed tests where the Cumulated Sum (CUSUM) and the Geometric Moving Average (GMA). The CUSUM test raises an alarm if the mean of the input data is significantly different from zero, while GMA checks if the weighted average of examples in a window is higher than a given threshold.

## 3 Concept Drift Detection

### 3.1 Decision Templates

The Decision template (DT) is one possible approach to build the MCSs. DT was proposed in [8]. In this MCS model DT are calculated based on the training set, one per class label. In the operation phase the similarity between each DT and outputs of base classifiers for object $x$ is computed. The class label with the closest DT is assigned to object $x$. For $K$ base classifier their outputs are arranged in the decision profile (DP):

$$DP(x) = \begin{bmatrix} p_1(1|x) & \vdots & p_1(M|x) \\ \vdots & \vdots & \vdots \\ p_K(1|x) & \vdots & p_K(M|x) \end{bmatrix}. \tag{1}$$

During learning of the base classifiers we obtain $m$ decision profiles, where $m$ is the number of objects from the learning set. From the decision profiles we calculate $DT_i$ with is the mean of the decision profiles belongs to the class label $i$. $DT_i$ for the class label $i$ is the centroid of the class label $i$ in the intermediate feature space. $DT_i$ can be regarded as the expected $DP(x)$ for the class label $i$.

### 3.2 Proposal Concept Drift Detection

Now we present the concept drift detection which uses DTs. The proposed method uses the *similarity* between the current $PD(x)$ and all $DT_i$. We use Euclidean distance for calculating the similarity but other measures can also be applied. At the beginning we calculate squared Euclidean distance between $PD(x)$ and each $DT_i$ as follows:

$$ED_i(PD(x), DT_i) = \frac{1}{KM} \sum_{k=1}^{K} \sum_{m=1}^{M} (p_k(m|x) - dt_i(k,m))^2, \tag{2}$$

where $dt_i(k,m)$ is $k,m$ entry in the decision template $DT_i$. Then, the rate DD (the drift detection) is calculated as follows:

$$DD = \sum_{i=1}^{M} ED_i. \tag{3}$$

The value of this ratio lies in the range $DD \in [0,..,M]$. Continuous increase of $DD$ for the new object suggests that the class distribution or class labels are changing. It means that the concept drift occurred. We propose the warning level as equal to $M/2$ and the drift level equal to $0.75M$. We assume that these values occur for the following $l$ new observations.

## 4 Experimental Studies

In this section we describe the evaluation of the proposed method of the drift detection. As we want to know how our algorithms work in the stream environment, we have chosen used datasets that have the sudden and incremental drift. In the experiential research we used two data sets from the UCI repository [6], one data set form Keel repository and the two generated randomly – are the so called Banana and Highleyman sets. The numbers of attributes and examples are presented in Table 1. In the experiments 9 base classifiers were used. The classifiers $\Psi_1$ is the multilayer perceptron model with 21 hidden units and $\Psi_2$ is a multilayer perceptron model with 7 hidden units. The classifier $\Psi_3$ uses the Support Vector Machines models with Decomposed Quadratic Programming estimation method and the classifier $\Psi_4$ uses the Least Squares Regression model to predict the class label. The classifier labelled as $\Psi_5$ uses the decision trees algorithms, with the splitting rule based on the gini index, the number of branches equal to 6 and the depth of the precision tree having at most 8 levels. The classifier $\Psi_6$ uses entropy as the splitting rule, the number of branches equal to 2 and the depth of the precision tree having at most 6 levels. The $\Psi_7$ is the gradient boosting decision tree. Last two of them work according to $k - NN$ rule where $k$ parameter is equal to 5 or 7 and are labelled as $\Psi_8$ and $\Psi_9$ respectively. We uses SAS Enterprise Miner 12.3 for building base learning model and SAS Base Software 9.4 for other experiments. Tables 2, 3, 4, 5 and 6 show the total number of changes detected by the learning algorithms with proposed drift detection methods for supervised problem. The experiments were repeated 10 times.

**Table 1.** Description of data sets selected for the experiments

| Data set | Example | Attribute |
|---|---|---|
| Banana | 2000 | 2 |
| Highleyman | 400 | 2 |
| MAGIC Gamma Telescope | 19020 | 11 |
| Phone | 961 | 6 |
| Pima Indians Diabetes | 768 | 8 |

The experiments shows that the proposed method of detecting the drift often detects the sudden drift. However, the method does not allow the perfect drift detection – the value was equal to 10. The experiments also show that increasing the chunk size influences the increase in the number of the detected drifts. The number of the detected changes also affects the used data set. The worst result is the most numerous data set – MAGIC. Therefore, for larger data sets it is recommended to increase the chunk size. In comparison with other methods (DDM, CUSUM) the proposed approach (DD) does not allow to achieve better results. In most cases, we obtain a smaller number of the detected drifts.

**Table 2.** Total number of changes detected by proposed the method (10 drifts) as a function of the chunk size – Banana data set

| Chunk size | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Measure | Drift type | Detector | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| No. of detected drifts | Sudden | DD | 1 | 1 | 1 | 3 | 5 | 5 | 5 | 6 | 6 | 7 |
| | | DDM | 4 | 5 | 5 | 5 | 6 | 6 | 7 | 7 | 7 | 9 |
| | | CUSUM | 4 | 5 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 8 |
| | Incremen. | DD | 0 | 0 | 2 | 1 | 2 | 3 | 3 | 4 | 5 | 5 |
| | | DDM | 1 | 2 | 3 | 5 | 6 | 5 | 7 | 7 | 8 | 9 |
| | | CUSUM | 2 | 5 | 5 | 6 | 6 | 7 | 7 | 8 | 7 | 8 |
| No. of false detect | Sudden | DD | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | DDM | 1 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | CUSUM | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Incremen. | DD | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | DDM | 2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | CUSUM | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 3.** Total number of changes detected by proposed the method (10 drifts) as a function of the chunk size – Highleyman data set

| Chunk size | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Measure | Drift type | Detector | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| No. of detected drifts | Sudden | DD | 3 | 3 | 4 | 5 | 6 | 7 | 7 | 8 | 9 | 9 |
| | | DDM | 3 | 4 | 5 | 5 | 6 | 7 | 7 | 10 | 10 | 10 |
| | | CUSUM | 4 | 5 | 5 | 5 | 7 | 6 | 6 | 7 | 10 | 9 |
| | Incremen. | DD | 2 | 1 | 2 | 3 | 5 | 4 | 5 | 6 | 6 | 6 |
| | | DDM | 1 | 2 | 3 | 5 | 5 | 5 | 7 | 7 | 9 | 9 |
| | | CUSUM | 2 | 5 | 5 | 5 | 6 | 7 | 7 | 8 | 8 | 8 |
| No. of false detect | Sudden | DD | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | DDM | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | CUSUM | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Incremen. | DD | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | DDM | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | CUSUM | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Similar observations can been reported for the false detection. The proposed method achieves slightly worse results. The comparison with other detectors is also the disadvantage of DD method. However, the achieved results do not improve the results of other methods related to experimental research on other base classifiers or other parameters. The results presented in Tables 7 and 8

**Table 4.** Total number of changes detected by proposed the method (10 drifts) as a function of the chunk size – MAGIC data set

| Chunk size | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Measure | Drift type | Detector | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| No. of detected drifts | Sudden | DD | 0 | 0 | 0 | 1 | 1 | 1 | 3 | 2 | 3 | 4 |
| | | DDM | 3 | 4 | 5 | 5 | 5 | 7 | 7 | 8 | 8 | 8 |
| | | CUSUM | 4 | 5 | 4 | 5 | 7 | 6 | 6 | 5 | 8 | 8 |
| | Incremen. | DD | 2 | 1 | 2 | 3 | 5 | 4 | 5 | 6 | 6 | 6 |
| | | DDM | 1 | 2 | 3 | 5 | 5 | 6 | 7 | 6 | 9 | 9 |
| | | CUSUM | 2 | 5 | 5 | 5 | 6 | 7 | 7 | 7 | 7 | 7 |
| No. of false detect | Sudden | DD | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| | | DDM | 2 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | | CUSUM | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Incremen. | DD | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | | DDM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | CUSUM | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 5.** Total number of changes detected by proposed the method (10 drifts) as a function of the chunk size – Phone data set

| Chunk size | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Measure | Drift type | Detector | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| No. of detected drifts | Sudden | DD | 2 | 2 | 2 | 5 | 5 | 6 | 5 | 7 | 9 | 8 |
| | | DDM | 3 | 4 | 5 | 5 | 5 | 7 | 7 | 9 | 9 | 10 |
| | | CUSUM | 4 | 5 | 4 | 5 | 7 | 6 | 7 | 9 | 9 | 10 |
| | Incremen. | DD | 0 | 0 | 1 | 2 | 2 | 2 | 3 | 3 | 4 | 5 |
| | | DDM | 2 | 2 | 3 | 5 | 5 | 6 | 7 | 6 | 9 | 10 |
| | | CUSUM | 2 | 4 | 4 | 5 | 6 | 7 | 7 | 7 | 10 | 10 |
| No. of false detect | Sudden | DD | 2 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| | | DDM | 2 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | | CUSUM | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Incremen. | DD | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| | | DDM | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | CUSUM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

indicated that the methods used in this work have little influence on the accuracies of the classification. The classification error in each data set are very similar, usually differing by less than 1%. It should also be noted that better results of classification has been obtained for the incremental drift.

**Table 6.** Total number of changes detected by proposed the method (10 drifts) as a function of the chunk size – Pima data set

| Chunk size | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Measure | Drift type | Detector | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| No. of detected drifts | Sudden | DD | 1 | 1 | 2 | 5 | 5 | 6 | 5 | 7 | 8 | 8 |
| | | DDM | 3 | 5 | 5 | 5 | 5 | 7 | 7 | 8 | 9 | 9 |
| | | CUSUM | 4 | 4 | 4 | 5 | 7 | 6 | 7 | 9 | 9 | 9 |
| | Incremen. | DD | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 4 | 4 |
| | | DDM | 2 | 2 | 3 | 5 | 4 | 6 | 7 | 8 | 8 | 9 |
| | | CUSUM | 2 | 4 | 4 | 5 | 5 | 7 | 7 | 7 | 8 | 18 |
| No. of false detect | Sudden | DD | 2 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | | DDM | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | | CUSUM | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Incremen. | DD | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | | DDM | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | CUSUM | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 7.** Classification error – incremental drift

| Data set | DD | DDM | CUSUM |
|---|---|---|---|
| Banana | 0.94 ± 1.3 | 0.95 ± 1.5 | 0.95 ± 1.4 |
| Highleyman | 0.89 ± 1.1 | 0.89 ± 1.5 | 0.89 ± 1.7 |
| MAGIC Gamma Telescope | 0.80 ± 0.7 | 0.79 ± 1.1 | 0.8 ± 0.6 |
| Phone | 0.75 ± 1.8 | 0.74 ± 1.0 | 0.74 ± 1.2 |
| Pima Indians Diabetes | 0.71 ± 0.5 | 0.72 ± 1.2 | 0.71 ± 1.4 |

**Table 8.** Classification error – sudden drift

| Data set | DD | DDM | CUSUM |
|---|---|---|---|
| Banana | 0.90 ± 1.7 | 0.91 ± 1.6 | 0.91 ± 1.8 |
| Highleyman | 0.84 ± 1.5 | 0.84 ± 1.8 | 0.83 ± 2.0 |
| MAGIC Gamma Telescope | 0.74 ± 1.3 | 0.73 ± 1.7 | 0.74 ± 1.6 |
| Phone | 0.72 ± 2.5 | 0.72 ± 1.6 | 0.71 ± 1.5 |
| Pima Indians Diabetes | 0.64 ± 1.2 | 0.65 ± 1.6 | 0.64 ± 1.6 |

## 5 Conclusion

In this paper we propose the new concept drift detection algorithm. The presented algorithm is based on the analysis of the decision templates returned from the base classifiers during the training process. In the paper experiments on five data set were carried out. The results indicate that, for each data set the proposed algorithm is better for the sudden drift than for the incremental drift. That conclusion is correct for the same chunk size used in the experiments. The experiments also show that increasing the chunk size increases the number of the detected drifts. The comparison with other detectors shows that the proposed method does not allow to achieve better results than DDM and CUSUM methods. Future studies should address to other parameters in the experiments such as base classifiers and parameters of the proposed method.

**Acknowledgments.** This work was supported in part by the Polish National Science Center under the grant no. DEC-2013/09/B/ST6/02264.

## References

1. Bach, S.H., Maloof, M.A.: Paired learners for concept drift. In: 2008 Eighth IEEE International Conference on Data Mining, pp. 23–32. IEEE (2008)
2. Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavalda, R., Morales-Bueno, R.: Early drift detection method. In: Fourth International Workshop on Knowledge Discovery from Data Streams, vol. 6, pp. 77–86 (2006)
3. Bifet Figuerol, A.C., et al.: Adaptive learning and mining for data streams and frequent patterns (2009)
4. Brzezinski, D., Stefanowski, J.: Reacting to different types of concept drift: the accuracy updated ensemble algorithm. IEEE Trans. Neural Netw. Learn. Syst. **25**(1), 81–94 (2014)
5. Dries, A., Rückert, U.: Adaptive concept drift detection. Stat. Anal. Data Min. ASA Data Sci. J. **2**(5–6), 311–327 (2009)
6. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
7. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS (LNAI), vol. 3171, pp. 286–295. Springer, Heidelberg (2004). doi:10.1007/978-3-540-28645-5_29
8. Kuncheva, L.I.: Switching between selection and fusion in combining classifiers: an experiment. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **32**(2), 146–156 (2002)
9. Tsymbal, A.: The problem of concept drift: definitions and related work. Computer Science Department, Trinity College Dublin, 106 (2004)

# An Algorithm for Detecting the Instant of Olfactory Stimulus Perception, Using the EEG Signal and the Hilbert-Huang Transform

Edward Puchala and Maciej Krysmann(✉)

Department of Systems and Computer Networks,
Wroclaw University of Science and Technology, Wroclaw, Poland
{edward.puchala,maciej.krysmann}@pwr.edu.pl
http://www.kssk.pwr.edu.pl

**Abstract.** The paper describes approach to instant of olfactory stimulus perception detection. Classification of olfactory stimuli in EEG is complex, but very important task. It allows to describe cognitive process and help in medical diagnosis process. Due to chemical - electrical nature of olfactory perception, there is need of solution which provide detection of beginning stimuli in EEG signal. Other way classification of olfactory stimuli would be more complex, due to not accurate in objects localization in learning set. Therefore the paper proposes utilization of Hilbert-Huang transformation in pre-processing. Proposed approach is evaluated and it have proven it's usability.

**Keywords:** Hilbert-Huang transform · Instant detection · Olfactory stimulus

## 1 Introduction

The classification of olfactory stimuli is a difficult and complex task. According to different sources, the number of olfactory stimuli is estimated to range from 10 thousands to a billion [1]. The recording, analysis and interpreting of olfactory stimuli not only are vital for the description of cognitive processes, but also enable the implementation of medical diagnostics systems dedicated to, for example, detecting mental problems in patients. Depression states are known to affect the registering of olfactory stimuli and their interpretation. Moreover, the detection and analysis of changes in the perception of smells over time can be an effective tool aiding the diagnosis of other (e.g. neurological) diseases. The reaction to an olfactory stimulus is first chemical and then it is registered in the brain as a specific electrical impulse. The diversity of aromatic compounds to a significant extent makes it difficult to record the instants of reaction to olfactory stimuli. Moreover, the reaction varies depending on the time of the day,

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9_52

the quality of the air, the physical condition, the degree of "nose fatigue", etc. Therefore, as part of this research an attempt was made to use electroencephalograms (EEG) to detect smells, especially to detect the instants of reaction to external stimuli [2–6]. Considering the well-known requirements concerning computerized classification systems, learning processes should be taken into account first. In order for such a classification to be of satisfactory quality it is necessary to acquire the best possible training sets covering the period from the instant at which a smell appears in the environment to the instant when it is registered by the human being. EEG traces were subjected to the Hilbert-Huang transformation [7–10] as part of pre-processing in order to determine the instants when the reaction to an external stimulus occurred. This provided the basis for determining the component vectors of the identified object's characteristics and designing, building and testing an algorithm aiding diagnostic decision making.

## 2 Hilbert-Huang Transform

In this chapter, Hilbert-Huang transform for EEG signals, will be presented. Hilbert-Huang transform consists of two fundamental processes:

1. Method of empirical mode decomposition – EMD,
2. Hilbert spectral analysis – HSA.

An empirical mode decomposition is a method with which any EEG signals set is decomposed into finite number of intrinsic mode functions (IMF). Intrinsic mode functions should have two properties:

1. The total number of maxima and minima must be equal or differ by one from the number of zero crossing,
2. The mean value of two envelopes (defined by the local maxima and local minima) at any point of IMF should be equal zero (Fig. 1).

For the feature extraction process for EEG signals so called the fission approach (presented below) will be applied. In this case the feature vectors of EEG signals consist of the features from each intrinsic mode functions. Fission approach for EEG signals:

1. Determine all maxima and minima of $EEG(t)$,
2. Using cubic spline curve connect all maxima to obtain the upper envelopes $UE(t)$,
3. Using cubic spline curve connect all minima to obtain the lower envelopes $LE(t)$,
4. Determine average signal $m_1(t)$:

$$m_1(t) = \frac{UE(t) + LE(t)}{2}, \tag{1}$$

5. Compute:

$$h_1(t) = EEG(t) - m_1(t), \tag{2}$$

6. Iterate steps 1–5 for obtaining:

$$h_{1,1}(t) = h_1(t) - m_{1,1}(t), \tag{3}$$

where $m_{1,1}(t)$ denotes average signal for upper and lower envelopes of $h_1(t)$,

7. Iterate 1–5 (according to the *point 6*) until:

$$h_{1,k}(t) = h_{1,k-1}(t) - m_{1,k}(t) \tag{4}$$

then: first intrinsic mode function will be defined as:

$$h_{1,k}(t) = IMF_1(t) \tag{5}$$

The formula in *point 7* defines stopping criteria. Another form of stopping criteria is presented below:

$$\frac{\sum_{t=0}^{T} |(h_{1,k-1}(t) - h_{1,k}(t)) \times (h_{j,k-1}(t) - h_{1,k}(t))|}{\sum_{t=0}^{T} [h_{1,k}(t) \times h_{1,k}(t)]} < \varepsilon \tag{6}$$

where $\varepsilon$ is a stopping parameter,

8. Compute the residual $r_1(t)$:

$$r_1(t) = EEG(t) - IMF_1(t) \tag{7}$$

and repeat whole procedure for $r_1(t)$,

9. Iterate points 1–8 until $r_k(t)$ will be a monotonic function and:

   - the total number of maxima and minima must be equal or differ by one from the number of zero crossing,
   - the mean value of two envelopes (defined by the local maxima and local minima) at any point of IMF should be equal zero.

The last function $r_k(t)$ is the component with a smallest frequency (trend). Completely decomposed $EEG(t)$ signals($D(t)$) can be represented as the sum of all $IMF_j(t)$ and residuals $R(t)$:

$$EEG(t) \sim D(t) = R(t) + \sum_{j=1}^{n} IMF_j(t). \tag{8}$$

After completing the fission procedure, for all $IMF_j(t)$ Hilbert transform $HT[IMF_j(t)]$ should be determined. Using $HT[IMF_j(t)]$ all intrinsic mode function are represented as an analytic signal:

$$z_j(t) = IMF_j(t) + i \times H[IMF_j(t)] = A_j(t) \times e^{-i\theta_j}(t). \tag{9}$$

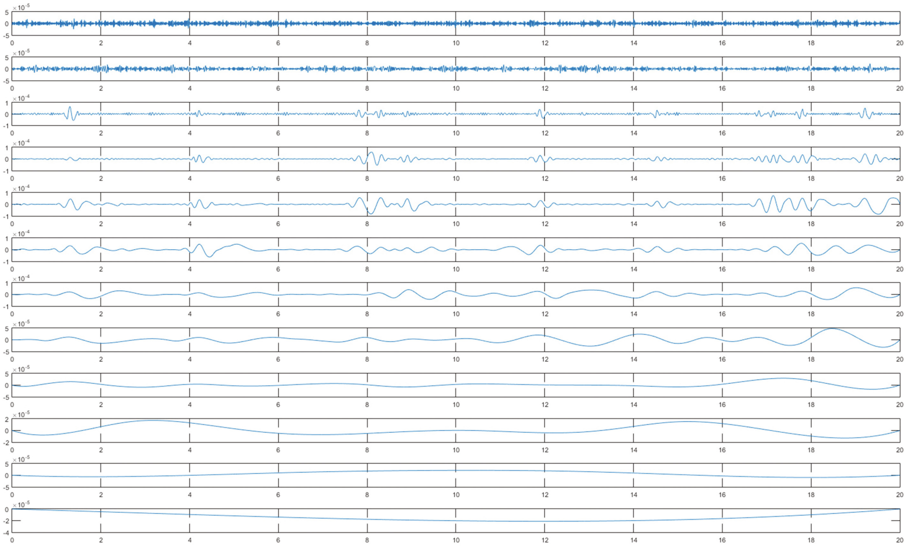


**Fig. 1.** EEG signal decomposition (20 s of signal)

## 3 Algorithm for Detecting Instant of Reaction to External Stimulus

In order to create the algorithm, signals from 6 electrodes, except the ones located in the optical cortex region, were examined. The focus was on the area responsible for olfactory stimuli. The following electrode locations were selected: Fp1, Fp2, F7, F8, T7 and T8 (Fig. 2).

The recording of signals from the electrodes mentioned above would begin simultaneously with the signal indicating the instant of administering (via a system of tubes) a chemical compound (vapour) to under the tested person's nose. The EEG signals recorded in this way were subjected to the Hilbert-Huang transform and then a search for any information revealing the reaction of the brain to the olfactory stimulus was carried out. When limonene was administered, a 250 ms long correlated segment would appear in the second IMFs for the signals coming from the electrodes located above areas Fp1 and Fp2 (Fig. 3). Further studies confirmed the existence of the correlation for a period of about 2 s since administering the compound, also for the other two chemical compounds (ethanol and toluene). Ultimately the following algorithm was proposed:

1. Using the Hilbert-Huang transform determine the IMF(2) for each second of the recordings from electrodes Fp1 and Fp2.

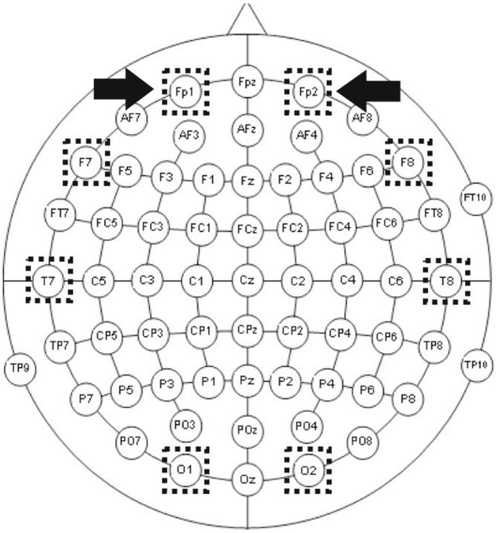


**Fig. 2.** Electrode locations: Fp1, Fp2, F7, F8, T7 and T8.

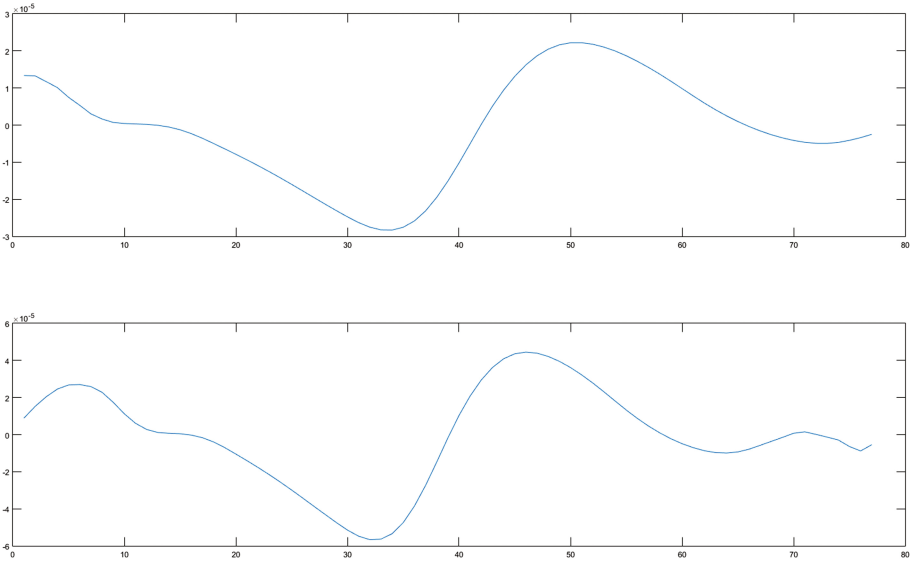


**Fig. 3.** Correlated IMFs from Fp1 and Fp2, 250 ms

2. Assuming a window with a length of 64 samples, overlapping by 32 samples (sampling frequency 256 Hz), examine correlations between the IMFs (2) for Fp1 and Fp2, using Pearson's correlation coefficient.
3. If the correlation is higher than 0.85, regard the beginning of the given window as the beginning of the reaction to the olfactory stimulus.

## 4 Data and Experiments

The data for the investigations were recordings from nine sessions (three sessions per each of the chemical compounds). Each single session (for each of the compounds) lasted 90 min, with the compound administered every 5 min. In this way during each session 18 EEG recording segments (objects), showing the reaction of the brain to the administration of an aromatic compound, were obtained. Even though the compounds administered to the tested person differed in their chemical structure, they were still aromatic compounds (ethanol, limonene and toluene). The experimental setup made it possible to verify the instant at which the compound reached the person. A sensor detecting the instant at which the compound appeared was mounted close to the tested person's nose. Also the instant of opening the reservoir containing the aromatic compound was recorded. At the same time the EEG signals were recorded. The device records 8 EEG channels from any sites on the head at a sampling rate of 256 Hz. The signals are collected from the areas marked in the figure. The algorithm described in this paper uses only two channels: Fp1 and Fp2. Ultimately, rejecting the incorrectly recorded (due to artefacts and other interference) objects, 16 objects per session were obtained. Considering that there were 3 sessions for each of the 3 compounds, a total of 144 objects were obtained. During experiments conducted to verify the discovered correlation the beginning of the stimulus was effectively detected for 131 objects, which amounts to the detection efficiency of 91%. In the other cases, no correlation was detected after the (marked) compound administration. This effect could have been due to the dispersion of the tested object or to interference by another stimulus. No dependence between the recording-correlation time distance and the kind of investigated stimulus was found.

## 5 Conclusions and Further Research

The presented application of the Hilbert-Huang transform in the detection of the instant of perception of an olfactory stimulus in the EEG signal has been found to be effective. The results show the detection to be stable and correct. By limiting the problem size - by reducing the number of IMFs to be determined and selecting EEG channels - it is possible to create a system which will operate on-line. As part of further research the algorithm described here will be implemented in a system classifying the EEG signal with regard to aromatic compounds and air. Moreover, it is necessary to extend the investigations to cover a larger number of persons, and other chemical compounds.

**Acknowledgments.** This work was supported by the statutory funds of the Department of Systems and Computer Networks, Wroclaw University of Science and Technology.

## References

1. Bushdid, C., Magnasco, M.O., Vosshall, L.B., Keller, A.: Humans can discriminate more than 1 trillion olfactory stimuli. Science **343**, 1370–1372 (2014)
2. Isaksson, A., Wennberg, A., Zetterberg, L.: Computer analysis of EEG signals with parametric models. Proc. IEEE **69**(4), 451–461 (1981)
3. Guger, C., Ramoser, H., Pfurtscheller, G.: Real-time EEG analysis with subject-specific spatial patterns for a brain-computer interface (BCI). IEEE Trans. Rehabil. Eng. **8**(4), 447–456 (2000)
4. Tolonen, U., Sulg, I.: Comparison of quantitative EEG parameters from four different analysis techniques in evaluation of relationships between EEG and CBF in brain infarction. Electroencephalogr. Clin. Neurophysiol. **51**(2), 177–185 (1981)
5. Gevins, A., Yeager, C., Zeitlin, G., Ancoli, S.: On-line computer rejection of EEG artifact. Electroencephalogr. Clin. Neurophysiol. **42**(2), 267–274 (1979)
6. Wennberg, A., Zetterberg, L.: Application of a computer-based model for EEG analysis. Electroencephalogr. Clin. Neurophysiol. **31**(5), 457–468 (1971)
7. Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.C., Tung, C.C., Liu, H.H.: The empirical mode decomposition and the Hilbert spectrum for nonlinear and non stationary time series analysis. Proc. R. Soc. A Math. Phys. Eng. Sci. **454**(1971), 903–995 (1998)
8. Li, H., Yang, L., Huang, D.: Application of Hilbert Huang transform to heart rate variability analysis. In: The 2nd International Conference on Bioinformatics and Biomedical Engineering, pp. 648–651 (2008)
9. Zong, C., Chetouani, M.: Hilbert-Huang transform based physiological signals analysis for emotion recognition. In: IEEE International Symposium on Signal Processing and Information Technology (ISSPIT) (2009)
10. Barnhart, B.L.: The Hilbert Huang: theory, applications, development. University of Iowa, Iowa Research Online. http://ir.uiowa.edu/etd/2670

# Author Index

M. Kurzynski et al. (eds.), *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, Advances in Intelligent Systems and Computing 578, DOI 10.1007/978-3-319-59162-9