# PATTERNS 2015

The Seventh International Conferences on Pervasive Patterns and Applications

ISBN: 978-1-61208-393-3

March 22 - 27, 2015

Nice, France

**PATTERNS 2015 Editors**

Fritz Laux, Reutlingen University, Germany

Herwig Manaert, University of Antwerp, Belgium

# PATTERNS 2015

# Forward

The Seventh International Conferences on Pervasive Patterns and Applications (PATTERNS 2015), held between March 22-27, 2015 in Nice, France, continued a series of events targeting the application of advanced patterns, at-large. In addition to support for patterns and pattern processing, special categories of patterns covering ubiquity, software, security, communications, discovery and decision were considered. It is believed that patterns play an important role on cognition, automation, and service computation and orchestration areas. Antipatterns come as a normal output as needed lessons learned.

The conference had the following tracks:

- Patterns basics
- Patterns at work
- Discovery and decision patterns

Similar to the previous edition, this event attracted excellent contributions and active participation from all over the world. We were very pleased to receive top quality contributions.

We take here the opportunity to warmly thank all the members of the PATTERNS 2015 technical program committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to PATTERNS 2015. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the PATTERNS 2015 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope PATTERNS 2015 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of pervasive patterns and applications. We also hope that Nice, France provided a pleasant environment during the conference and everyone saved some time to enjoy the charm of the city.

## PATTERNS 2015 Chairs

### PATTERNS Advisory Chairs

Herwig Manaert, University of Antwerp, Belgium
Fritz Laux, Reutlingen University, Germany
Michal Zemlicka, Charles University - Prague, Czech Republic
Alfred Zimmermann, Reutlingen University, Germany
Richard Laing, Robert Gordon University, UK
Ricardo Sanz, UPM ASlab, Spain
Mauricio Hess-Flores, University of California, Davis, USA

### PATTERNS Research/Industry Chairs

Teemu Kanstren, VTT, Finland
Markus Goldstein, Kyushu University, Japan
Zhenzhen Ye, IBM, Essex Junction, USA
René Reiners, Fraunhofer FIT - Sankt Augustin, Germany
Nguyen Vu Hoàng, Vertigo, France
Alexander Mirnig, University of Salzburg, Austria
Juan C Pelaez, Defense Information Systems Agency, USA

### PATTERNS Publicity Chairs

Bastian Roth, University of Bayreuth, Germany
Steve Strauch, IAAS at University of Stuttgart, Germany
Jaap Kabbedijk, Utrecht University, Netherlands

# PATTERNS 2015

## Committee

### PATTERNS Advisory Chairs

Herwig Manaert, University of Antwerp, Belgium
Fritz Laux, Reutlingen University, Germany
Michal Zemlicka, Charles University - Prague, Czech Republic
Alfred Zimmermann, Reutlingen University, Germany
Richard Laing, Robert Gordon University, UK
Ricardo Sanz, UPM ASlab, Spain
Mauricio Hess-Flores, University of California, Davis, USA

### PATTERNS Research/Industry Chairs

Teemu Kanstren, VTT, Finland
Markus Goldstein, Kyushu University, Japan
Zhenzhen Ye, IBM, Essex Junction, USA
René Reiners, Fraunhofer FIT - Sankt Augustin, Germany
Nguyen Vu Hoàng, Vertigo, France
Alexander Mirnig, University of Salzburg, Austria
Juan C Pelaez, Defense Information Systems Agency, USA

### PATTERNS Publicity Chairs

Bastian Roth, University of Bayreuth, Germany
Steve Strauch, IAAS at University of Stuttgart, Germany
Jaap Kabbedijk, Utrecht University, Netherlands

### PATTERNS 2015 Technical Program Committee

Ina Suryani Ab Rahim, Pensyarah University, Malaysia
Junia Anacleto, Federal University of Sao Carlos, Brazil
Andreas S. Andreou, Cyprus University of Technology - Limassol, Cyprus
Annalisa Appice, Università degli Studi di Bari, Italy
Martin Atzmueller, University of Kassel, Germany
Senén Barro, University of Santiago de Compostela, Spain
Rémi Bastide, University Champollion / IHCS - IRIT, France
Bernhard Bauer, University of Augsburg, Germany
Noureddine Belkhatir , University of Grenoble, France

Hatem Ben Sta, Université de Tunis - El Manar, Tunisia
Silvia Biasotti, Consiglio Nazionale delle Ricerche, Italy
Félix Biscarri, University of Seville, Spain
Cristian Bonanomi, Universita' degli Studi di Milano, Italy
Mohamed-Rafik Bouguelia, LORIA - Université de Lorraine, France
Julien Broisin, IRIT - Université Paul Sabatier, France
Michaela Bunke, University of Bremen, Germany João Pascoal Faria, University of Porto, Portugal
Michelangelo Ceci, University of Bari, Italy
M. Emre Celebi, Louisiana State University in Shreveport, USA
Jian Chang, Bournemouth University, UK
William Cheng-Chung Chu(朱正忠), Tunghai University, Taiwan
Loglisci Corrado, University of Bari, Italy
Bernard Coulette, Université de Toulouse 2, France
Karl Cox, University of Brighton, UK
Jean-Charles Créput, Université de Technologie de Belfort-Montbéliard, France
Sergio Cruces, University of Seville, Spain
Mohamed Dahchour, National Institute of Posts and Telecommunications - Rabat, Morocco
Jacqueline Daykin, Royal Holloway University of London, UK
Angelica de Antonio, Universidad Politecnica de Madrid, Spain
Sara de Freitas, Coventry University, UK
Vincenzo Deufemia, Università di Salerno - Fisciano, Italy
Kamil Dimililer, Near East University, Cyprus
Zhong-Hui Duan, University of Akron, USA
Alberto Egon, Federal University of Rio Grande do Sul (UFRGS), Brazil
Giovanni Maria Farinella, University of Catania, Italy
Eduardo B. Fernandez, Florida Atlantic University - Boca Raton, USA
Simon Fong, University of Macau, Macau SAR
Francesco Fontanella, Università di Cassino e del Lazio Meridionale, Italy
Pawel Forczmanski, West Pomeranian University of Technology, Poland
Dariusz Frejlichowski, West Pomeranian University of Technology, Poland
Christos Gatzidis, Bournemouth University, UK
Markus Goldstein, Kyushu University, Japan
Gustavo González, Mediapro Research - Barcelona, Spain
Pascual Gonzalez Lopez, University of Castilla-La Mancha, Spain
Carmine Gravino, University of Salerno, Italy
Christos Grecos, University of the West of Scotland, UK
Yann-Gaël Guéhéneuc, École Polytechnique - Montreal, Canada
Pierre Hadaya, UQAM, Canada
Brahim Hamid, IRIT-Toulouse, France
Sven Hartmann, TU-Clausthal, Germany
Kenneth Hawick, Massey University, New Zealand
Mauricio Hess-Flores, University of California, USA
Christina Hochleitner, AIT - Austrian Institute of Technology, Austria

Władysław Homenda, Warsaw University of Technology, Poland
Samuelson W. Hong, Zhejiang University of Finance & Economics, China
Chih-Cheng Hung, Southern Polytechnic State University-Marietta, USA
Shareeful Islam, University of East London, UK
Yuji Iwahori, Chubu University, Japan
Slinger Jansen (Roijackers), Utrecht University, The Netherlands
Agnieszka Jastrzebska, Warsaw University of Technology, Poland
Jinyuan Jia, Tongji University, China
Xingpeng Jiang, Drexel University, USA
Maria João Ferreira, Universidade Portucalense - Porto, Portugal
Jaap Kabbedijk, Utrecht University, Netherlands
Hermann Kaindl, TU-Wien, Austria
Abraham Kandel, University South Florida - Tampa, USA
Teemu Kanstren, VTT, Finland
Alexander Knapp, Universität Augsburg, Germany
Sotiris Kotsiantis, University of Patras, Greece
Adam Krzyzak, Concordia University, Canada
Richard Laing, The Scott Sutherland School of Architecture and Built Environment/ Robert Gordon University - Aberdeen, UK
Robert Laramee, Swansea University, UK
Fritz Laux, Reutlingen University, Germany
Hervé Leblanc, IRIT-Toulouse, France
Gyu Myoung Lee, Liverpool John Moores University, UK
Haim Levkowitz, University of Massachusetts Lowell, USA
Fotis Liarokapis, Coventry University, UK
Pericles Loucopoulos, Harokopio University of Athens, Greece / Loughborough University, UK
Herwig Manaert, University of Antwerp, Belgium
Elio Masciari, ICAR-CNR, Italy
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Fuensanta Medina-Dominguez, Carlos III University of Madrid, Spain
Alexander G. Mirnig, University of Salzburg, Austria
Ivan Mistrík, Independent Consultant. Heidelberg, Germany
Laura Monroe, Los Alamos National Laboratory, USA
Paula Morais, Universiadade Portucalense - Porto, Portugal
Fernando Moreira, Universidade Portucalense, Portugal
Antonino Nocera, University Mediterranea of Reggio Calabria, Italy
Jean-Marc Ogier, Université de la Rochelle, France
Krzysztof Okarma, West Pomeranian University of Technology, Poland
Hichem Omrani, CEPS/INSTEAD, Luxembourg
Jerry Overton, Computer Sciences Corporation, USA
Ana Paiva, University of Porto, Portugal
Eric Paquet, National Research Council / University of Ottawa, Canada
Vicente Palazón González, Universitat Jaume I, Spain
João Pascoal Faria, University of Porto, Portugal

Galina Pasko, Uformia, Norway
Rodrigo Paredes, Universidad de Talca, Chile
Giuseppe Patane', CNR-IMATI, Italy
Photis Patonis, Aristotle University of Thessaloniki, Greece
Juan C Pelaez, Defense Information Systems Agency, USA
Christian Percebois, IRIT/Université de Toulouse, France
Gabriel Pereira Lopes, Universidade Nova de Lisboa, Portugal
Luciano Pereira Soares, Insper, Brazil
José R. Pires Manso, University of Beira Interior, Portugal
Agostino Poggi, Università degli Studi di Parma, Italy
Sylvia C. Pont, Delft University of Technology, Netherlands
Giovanni Puglisi, University of Catania, Italy
Francisco A. Pujol, Universidad de Alicante, Spain
Mar Pujol, Universidad de Alicante, Spain
Claudia Raibulet, University of Milano-Bicocca, Italy
Giuliana Ramella, CNR - National Research Council, Italy
Theresa-Marie Rhyne, Consultant, USA
Alessandro Rizzi, Università degli Studi di Milano, Italy
Marcos A. Rodrigues, Sheffield Hallam University, UK
José Raúl Romero, University of Córdoba, Spain
Agostinho Rosa, Technical University of Lisbon, Portugal
Gustavo Rossi, UNLP - La Plata, Argentina
Ozgur Koray Sahingoz, Turkish Air Force Academy, Turkey
Lorenza Saitta, Università del Piemonte Orientale, Italy
Antonio-José Sánchez-Salmerón, Universidad Politécnica de Valencia, Spain
Maria-Isabel Sanchez-Segura, Carlos III University of Madrid, Spain
Kurt Sandkuhl, Jönköping University, Sweden
Hans-Jörg Schulz, Fraunhofer IGD Rostock, Germany
Isabel Seruca, Universidade Portucalense - Porto, Portugal
Caifeng Shan, Philips Research, The Netherlands
Karolj Skala, Ruder Boškovic Institute Zagreb, Croatia
Karina Sokolova (Perez), University of Technology of Troyes, France
Carina Soledad, Universidad de La Laguna, Spain
Michael Stal, Siemens, Germany
Janis Stirna, Stockholm University, Sweden
Mu-Chun Su, National Central University, Taiwan
Sam Supakkul, Sabre Inc., USA
Stella Sylaiou, Hellenic Open University, Greece
Ryszard Tadeusiewicz, AGH University of Science and Technology, Poland
Dan Tamir, Texas State University, USA
Shanyu Tang, China University of Geosciences - Wuhan City, P. R. China
Horia-Nicolai Teodorescu, "Gheorghe Asachi" Technical University of Iasi / Romanian Academy, Romania
Daniel Thalmann, Nanyang Technological University, Singapore

## Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

## Table of Contents

# Android Permission Usage: a First Step towards Detecting Abusive Applications

Karina Sokolova*, Marc Lemercier*

*University of Technology of Troyes
Troyes, France
email:{karina.sokolova, marc.lemercier}@utt.fr

Charles Perez†

†PSB Paris School of Business
Paris, France
email:cperez@faculty-esgms.fr

*Abstract*—**Thousands of mobile applications are available on mobile markets and actively used everyday. One of the mobile market leaders – Android – does not verify the security of applications published on its market and assumes that users will carefully judge the applications themselves using the information available on the marketplace. A common assumption is that the list of permissions associated with each application provides users with security and privacy indications, but previous works have shown that users are barely able to understand and analyse those permission lists. Very few works propose solutions that could help users in deciding whether or not to install an application. Despite Android permissions' lack of user-friendliness, they are an important source of information. In this work, we analyse permissions used by a large set of applications for different Android market categories and define the core permission patterns characterising each one. The patterns obtained are a first step towards building an indicator for detecting normal and possibly over-privileged applications on the market.**

***Keywords*–patterns, usage, mobile applications, Android, Google Play, permissions, Network science, graph analysis, data mining, category**

## I. Introduction

Today the mobile market is constantly growing; an increasing number of mobile applications are made available to users every day. Application distribution and security methods differ from one company to another, but the permission system – an explicative additional privilege for each sensitive service or piece of data – is often a common factor. Two mobile market leaders – Android and iOS – use a different approach to permission systems: iOS proposes a very limited number of permissions and gives users control over granting or revoking a single permission; Android proposes very large number of permissions, and users have to accept all of them at once before installing an application. iOS applications are checked for malicious or abusive codes before they are made available on the market. Android applications are uploaded directly on the market and it is up to users to judge each application using data available on the market, such as permission lists, comments and ratings provided by other users.

Some research has shown that users are often incapable of judging a mobile application's legitimacy simply by looking at the permissions requested by it prior to installation [1][2]. Android permissions are often very specific and contain technical terms. Some permissions are so widely used that users do not even pay attention to them when viewing the list. The result is that users often have not grasped the meaning and accept the list regardless of its permissions. This leads to important privacy and security issues that can be exploited by applications abusively requiring permissions. It is important to note that mobile applications can be intrusive without necessarily being malicious; some may want to collect users' information to 'unfairly' improve their customer relationship management (CRM).

In this paper, we aim to identify normal application permission usage patterns by application category. The Android market groups similar applications with similar ends into categories. Different functionalities require different data and services, which in turn imply different permissions. Our hypothesis is that categories on the market containing applications with similar functionalities will also require similar groups of permissions. We identify central and core permissions for categories and discuss related functionalities. We believe that such permission patterns can help create a measurement that allows users to compare easily more and less intrusive applications.

The remainder of the paper is organized as follows: Section 2 presents background on Android; Section 3 presents related works; the methodology is presented in Section 4; and Section 5 presents the results. The paper ends with a discussion about future works and a conclusion.

## II. Background

Android is an open-source operating system owned by Google. Since 2010, it has been a leader on the mobile market and used widely on smartphones, tablets and, more recently, on smart objects.

Android applications are available to users via the market store – GooglePlay. Google does not verify Android applications when they arrive on the market, and users should carefully check all available information to judge if an application is trustworthy and can be installed.

To help users evaluate applications, Android embeds a permission system security mechanism. Applications have very limited rights when accessing system services, sensitive data or sensors; therefore, developers must explicitly add permissions for each protected interface into a compulsory file called 'AndroidManifest'. By doing so, each application is associated with a list of permissions.

Android has a predefined list of permissions that developers can use. According to our analyses, Android 4.4 currently (November 2014) contains 229 permissions: 30 normal, 48 dangerous, 11 development, 70 signature and 70 signature or

ISBN: 978-1-61208-393-3

system permissions from which third-party applications may only use 89 Android permissions.

Native Android permissions have a similar prefix based on a predefined hierarchy: 'android.permissions.*'. For example, the permission for Internet access is defined as 'android.permission.INTERNET'. Only five Android permissions are prefixed with 'com.android.*': browser, alarm, launcher and voicemail-related permissions.

Android allows communication and data/functionalities to be shared between applications. Developers can define custom permissions to protect new interfaces in order to share services or data. Custom permissions can be named freely: Google does not impose any naming rules.

The Android permission system and permission usage are a valuable source of information. The next section presents the works related to permission analyses and their limitations.

## III. Related works

Very few works propose solutions to help users judge whether a given application abusively uses permissions and represents a potential threat.

Some authors propose monitoring the data flow of Android applications and reporting permission usage to users [3][4][5]. These solutions are not proactive, as the application must already be installed, and it is not clear that users could easily adopt this solution.

The authors of [6] proposed searching for a justification for permission usage in an application description using natural language processing (NLP) techniques, warning the user if this is not found. A proof of concept was carried out on three Android permissions. Further work improved the detection and number of supported permissions [7].

In [8], the authors created a crowd-sourcing system that collects and analyses iOS application configurations to provide users with privacy recommendations. This approach could be applied to Android applications; however, currently, the revocation of an individual permission is not possible with Android – users do not have any control over the permission list of an installed application. Moreover, Android's permission list is much longer and more technical than that of iOS, which makes it very laborious for a user to configure. Furthermore, this would be an application control solution, not an application choice or judgment solution.

Several works have been done on Android permission analyses.

In 2009, the authors of [9] analysed the permissions of the top 50 free Android applications using a self-organizing map (SOM). The authors provide some statistics on permission usage, identifying a series of pairs of correlated permissions and providing correlations between permissions and categories.

In [10], the authors analysed the permissions of the top 100 Android applications and found that most permissions were used occasionally, in response to the action made on the graphical user interface (GUI). The paper highlighted that only 5% of applications legitimately required some of the permissions granted permanently.

The authors of [11] analysed the permissions of Android's most popular and novel applications from both official and non-official markets. The authors analysed the interdependency of the number of permissions and the application popularity, price, availability of the developer's website and availability of privacy policies. They also analysed the number of permissions for applications with similar names.

None of the previous works have focused on analysing patterns in permission usage. Moreover, current analysis has only been performed on a very limited number of applications.

Few tools have analysed Android permissions for security purposes, one of them being the *Kirin* tool, which analysed AndroidManifest files to identify dangerous permission combinations and flag potential malware before installation [12]. The related paper identified two dangerous permissions and seven combinations of permissions, which were added to *Kirin*'s installation privacy policy. *SCAndroid* went even further, using source code analysis to identify if permissions were really used together by an application [13].

These latter works focused on the technical challenges related to embedding a permission-pattern-based tool in the Android system, but they did not discuss permission patterns directly. It is important to note that dangerous permission combinations were defined manually by the authors.

In [14], the authors used probabilistic methods to identify patterns for high- and low-ranked applications. The authors noted that pattern identifications by category would improve results.

Recent research [15] used statistical methods to identify the top 40 risky permissions and performed clustering techniques to identify patterns and detect malicious applications.

The authors of [16] analysed permission usage over a set of 1,227 clean and 49 malicious application families. The authors generated a list of permission patterns unique to Android malware but did not process the patterns of 'clean' or non-threatening applications. Due to calculation costs, this research only obtained patterns with a maximum of 4 permissions.

The authors of [17] used 999 Android applications to build a graph based on the co-occurrence of permissions in different application categories. The authors focused on determining in which cases approaches such as [12] could be applied to malware detection. The most frequent groups of permissions for each category were identified by a modularity-optimizing classification algorithm and were considered to be a normal request for an application from a given category. The authors compared these groups with dangerous permission combinations from [12] and found that some of the presumably risky combinations in groups in fact are legitimate. The authors noted that there was a bias in the analysis, namely that very popular permissions form important clusters in many different categories.

## IV. Methodology

This section presents the methodology applied to obtain categories permission patterns.

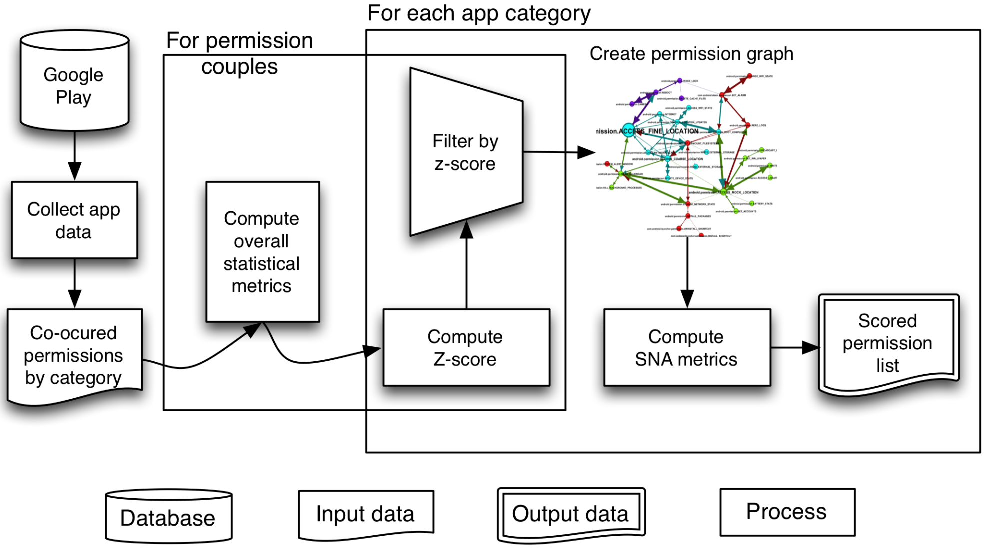


Figure 1. Methodology representation.

### A. Overview

In this document, we aim to address some of the issues raised by previous works. Our objective was to analyse the permission usage of a large set of applications (not explicitly malicious) available on the official Android market. First, we identified significant patterns for the official Android market application categories. In order to do so, we performed a statistical and graph analysis that allowed patterns to be identified without limiting the number of items involved in a pattern. We built a methodology to avoid over-connecting the most popular permissions, only keeping track of the most significant patterns for each category. Finally, we assessed the graph obtained for further avenues of analysis.

A diagram of the methodology is shown in Figure 1. First, we compiled applications and related data from Google Play and prepared the data for processing. We evaluated pairs of permissions that co-occurred in each category. Then, we performed a statistical analysis and $\mathcal{Z} - score$ to determine the significance of each pair in each category. We filtered our dataset, keeping only the most significant permission pairs. Finally, we created a graph of the most significant permissions by category and used graph analysis to determine the importance of each individual permission. As an output, we obtained a scored list of the most significant permissions by category.

The following sections explained the dataset and the methodology in detail.

### B. Dataset and initial observations

We compiled application data from the Google Play store using a publicly available non-official application programming interface (API) and a script written in PHP language published under the GNU General Public License [18]. We modified the script to match it to our objectives and stored the harvested data within a MySQL relational database.

We collected multiple types of information about applications available on Google Play: name, description, package name, version, users' note, number of downloads, price, category, number of screenshots, author and the list of permissions as defined in the manifest. For each category, we obtained the category's name and related description.

After launching a script to collect data, we obtained a sample of 9,512 applications related to 35 categories containing between 190 and 590 applications each. In our sample, we observed a set of 2,133 unique permissions, with 292 permissions identified as Android native permissions (263 matched the prefix 'android.permissions.*' and 29 matched the prefix 'com.android.*'). The other permissions are assumed to be custom permissions.

We compared the list obtained with a list of permissions extracted from Android 4.4 and found 157 permissions that did not match currently available Android permissions. These permissions were instead third-party application permissions, such as those for mobile device management (e.g., 'android.permission.sec.*'), old permissions from previous Android versions (e.g., 'com.android.launcher.permission.READ_SETTINGS'), permissions for Android in-app payment and licence libraries, and many misspelled permissions.

To carry out further analysis, we filtered our dataset to only keep permissions available in the Android 4.4 system.

Table I shows the top 10 permissions and the percentage of applications requiring these permissions. The INTERNET permission is the most required, as observed in previous works.

In the next section, we present the methodology we applied to obtain relevant patterns from our dataset.

### C. Analyses

Our proposal was to analyse significant co-occurrences of permission pairs for each application category. To do so, for each category $\mathcal{C}$ we created a graph denoted $G_{\mathcal{C}}(N_C, E_C)$, where the set of nodes $N_{\mathcal{C}}$ corresponded to the permissions, and the set of edges $E_C$ represented two commonly used permissions in the category. This common usage was identified if both permissions were observed together in the AndroidManifest file of at least one application in the category.

It is important to note that although some permission pairs may have been used jointly in many categories, they were not necessarily relevant to it [9]. For example, INTERNET and ACCESS_NETWORK_STATE were used commonly in many categories but were not relevant when trying to create a permission fingerprint for a category. For this reason, the significance of the usage of a permission pair in a category had to be moderated by (1) the average use of this pair across categories and (2) weighted with respect to how regularly it appeared across multiple categories.

To quantify these observations, which have also been pointed out in previous works [17], we proposed scoring the weight between a permission pair $\mathcal{A} : (perm_i, perm_j)$ in a category $\mathcal{C}$ with the standard score or $\mathcal{Z} - score$ defined in equation 1.

$$\mathcal{Z}_{\mathcal{A}}^{\mathcal{C}} = \frac{\mathcal{A}_{\mathcal{C}} - \mu_{\mathcal{A}}}{\sigma_{\mathcal{A}}} \qquad (1)$$

Where :

$\mathcal{A}$ is a permission pair $(perm_i, perm_j)$

$\mu_{\mathcal{A}}$ and $\sigma_{\mathcal{A}}$ are the mean and standard deviation of the usage of pair $\mathcal{A}$ across all categories.

$\mathcal{A}_{\mathcal{C}}$ is the observed usage of the pair in the category $\mathcal{C}$.

From equation 1, we propose defining the weight between two permissions $e_{\mathcal{C}}(perm_i, perm_j)$ for each $E_{\mathcal{C}}$ in graph $G_{\mathcal{C}}(N_{\mathcal{C}}, E_{\mathcal{C}})$ corresponding to a specific category $\mathcal{C}$ as follows:

$$e_{\mathcal{C}}(perm_i, perm_j) = \mathcal{Z}^{\mathcal{C}}{}_{(perm_i, perm_j)} \qquad (2)$$

We obtained a graph of permissions with weighted relations for each category. We then filtered the edges of each graph by weight to highlight only the most significant patterns. We have removed edges whose $\mathcal{Z}-score$ stands below threshold 2. This threshold, for a normally distributed population, allows only 2,3% of the most relevant edges to be kept track of. Finally, we filtered the nodes, keeping only nodes with a non-null degree.

TABLE I
TOP 10 OF PERMISSIONS USAGE. EACH PERMISSION IS PREFIXED ORIGINALLY WITH 'ANDROID.PERMISSION".

| **Permission** | **Applications (%)** |
|---|---|
| INTERNET | 91,88 |
| ACCESS_NETWORK_STATE | 83,31 |
| WRITE_EXTERNAL_STORAGE | 60,39 |
| READ_EXTERNAL_STORAGE | 60,29 |
| READ_PHONE_STATE | 49,92 |
| WAKE_LOCK | 33,01 |
| ACCESS_WIFI_STATE | 31,47 |
| VIBRATE | 30,07 |
| ACCESS_COARSE_LOCATION | 27,73 |
| ACCESS_FINE_LOCATION | 27,42 |

Afterwards, we computed several graph metrics and algorithms in order to highlight the patterns for each category graph.

The first step was to compute a weighted modularity-based clustering algorithm to highlight potential functionalities represented by a common permission usage [19][20]. The modularity regrouped the graph's elements into communities. This score increased as the number of edges within communities increased and the number of edges between these communities decreased. The clustering algorithm used greedy optimisation to build communities in a way that maximized the modularity score.

Secondly, the betweenness centrality was computed on the permission graph in order to detect the most crucial permissions for an application in a category. In the domain of social network analysis, the betweenness centrality of a node $v$ is measured as the ratio of the number of shortest paths between any node pairs $(s, t)$ that pass throughout $v$ by the number of shortest paths between these pairs. Mathematically, it is defined as stated in equation 3 below [21].

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \qquad (3)$$

Where :

$\sigma_{st}$ is the number of shortest path between two nodes $s$ and $t$

$\sigma_{st}(v)$ is the number of shortest paths between two nodes $s$ and $t$ that pass through $v$.

The betweenness centrality measured the capacity of a permission to belong to many of the shortest paths. A high betweenness centrality indicated that this permission was required to perform multiple tasks or for the main functionality of applications in the category. More social network analysis measures (degree, closeness centrality, PageRank, etc.) were tested, but they are not discussed in this paper.

In the next section, we present the results and patterns obtained from this analysis.

## V. RESULTS

We present below a set of results obtained from the analysis of permissions by categories using our dataset.

### A. Number of relevant pairs by category

Relevant patterns formed by relevant pairs exist for each application category. Table II displays the number of permissions pairs $(perm_i, perm_j)$ named as relevant pairs that were statistically significant (w.r.t. $\mathcal{Z}-score > 2$) for each category.

We could observe that six categories covered a very large set of relevant pairs (up to 1,000). This showed that these categories were very broad and covered many different types of functionalities. An abusive application belonging to these categories could be harder to detect using the permission list than abusive applications belonging to a category that exhibited a more reasonable set of relevant pairs. We noted that two

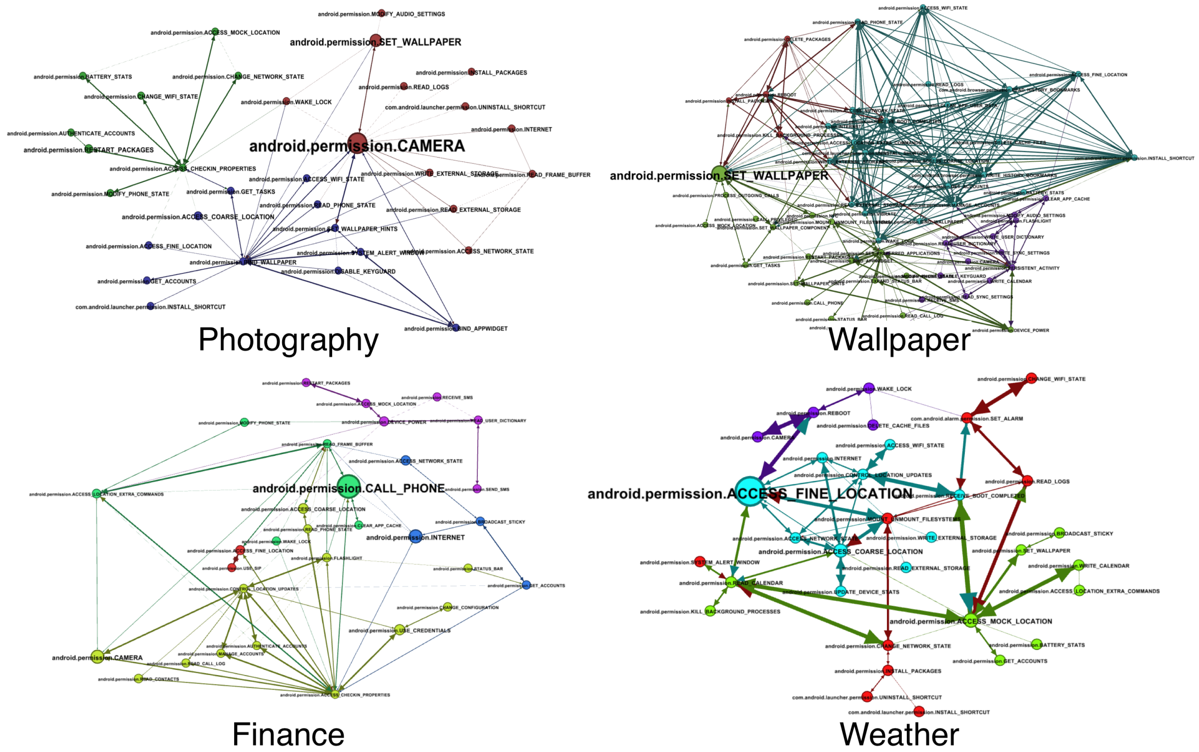

android.permission.MODIFY_AUDIO_SETTINGS
android.permission.ACCESS_MOCK_LOCATION
android.permission.SET_WALLPAPER
android.permission.BATTERY_STATS
android.permission.CHANGE_NETWORK_STATE
android.permission.INSTALL_PACKAGES
android.permission.CHANGE_WIFI_STATE
android.permission.READ_LOGS
android.permission.WAKE_LOCK
com.android.launcher.permission.UNINSTALL_SHORTCUT
android.permission.AUTHENTICATE_ACCOUNTS
android.permission.INTERNET
android.permission.RESTART_PACKAGES
android.permission.CAMERA
android.permission.ACCESS_CHECKIN_PROPERTIES
android.permission.WRITE_EXTERNAL_STORAGE
android.permission.READ_FRAME_BUFFER
android.permission.ACCESS_WIFI_STATE
android.permission.MODIFY_PHONE_STATE
android.permission.GET_TASKS
android.permission.READ_PHONE_STATE
android.permission.READ_EXTERNAL_STORAGE
android.permission.ACCESS_COARSE_LOCATION
android.permission.SET_WALLPAPER_HINTS
android.permission.ACCESS_FINE_LOCATION
android.permission.SYSTEM_ALERT_WINDOW
android.permission.ACCESS_NETWORK_STATE
android.permission.BIND_WALLPAPER
android.permission.DISABLE_KEYGUARD
android.permission.GET_ACCOUNTS
com.android.launcher.permission.INSTALL_SHORTCUT
android.permission.BIND_APPWIDGET
Photography
android.permission.SET_WALLPAPER
android.permission.INSTALL_SHORTCUT
android.permission.PERSISTENT_ACTIVITY
android.permission.SET_WALLPAPER_HINTS
android.permission.EXPAND_STATUS_BAR
android.permission.WRITE_CALENDAR
android.permission.READ_SYNC_SETTINGS
android.permission.CALL_PHONE
android.permission.RECEIVE_SMS
android.permission.READ_CALL_LOG
android.permission.STATUS_BAR
android.permission.DEVICE_POWER
Wallpaper
android.permission.RESTART_PACKAGES
android.permission.RECEIVE_SMS
android.permission.ACCESS_MOCK_LOCATION
android.permission.MODIFY_PHONE_STATE
android.permission.DEVICE_POWER
android.permission.READ_USER_DICTIONARY
android.permission.READ_FRAME_BUFFER
android.permission.ACCESS_NETWORK_STATE
android.permission.ACCESS_LOCATION_EXTRA_COMMANDS
android.permission.CALL_PHONE
android.permission.SEND_SMS
android.permission.ACCESS_COARSE_LOCATION
android.permission.CLEAR_APP_CACHE
android.permission.BROADCAST_STICKY
android.permission.READ_PHONE_STATE
android.permission.INTERNET
android.permission.WAKE_LOCK
android.permission.ACCESS_FINE_LOCATION
android.permission.FLASHLIGHT
android.permission.USE_SIP
android.permission.STATUS_BAR
android.permission.CONTROL_LOCATION_UPDATES
android.permission.GET_ACCOUNTS
android.permission.CHANGE_CONFIGURATION
android.permission.USE_CREDENTIALS
android.permission.AUTHENTICATE_ACCOUNTS
android.permission.MANAGE_ACCOUNTS
android.permission.CAMERA
android.permission.READ_CALL_LOG
android.permission.READ_CONTACTS
android.permission.ACCESS_CHECKIN_PROPERTIES
Finance
android.permission.CHANGE_WIFI_STATE
android.permission.WAKE_LOCK
android.permission.REBOOT
com.android.alarm.permission.SET_ALARM
android.permission.DELETE_CACHE_FILES
android.permission.CAMERA
android.permission.ACCESS_WIFI_STATE
android.permission.INTERNET
android.permission.CONTROL_LOCATION_UPDATES
android.permission.READ_LOGS
android.permission.ACCESS_FINE_LOCATION
android.permission.RECEIVE_BOOT_COMPLETED
android.permission.MOUNT_UNMOUNT_FILESYSTEMS
android.permission.BROADCAST_STICKY
android.permission.ACCESS_NETWORK_STATE
android.permission.WRITE_EXTERNAL_STORAGE
android.permission.ACCESS_COARSE_LOCATION
android.permission.SET_WALLPAPER
android.permission.SYSTEM_ALERT_WINDOW
android.permission.READ_EXTERNAL_STORAGE
android.permission.WRITE_CALENDAR
android.permission.READ_CALENDAR
android.permission.UPDATE_DEVICE_STATS
android.permission.ACCESS_LOCATION_EXTRA_COMMANDS
android.permission.KILL_BACKGROUND_PROCESSES
android.permission.ACCESS_MOCK_LOCATION
android.permission.CHANGE_NETWORK_STATE
android.permission.BATTERY_STATS
android.permission.GET_ACCOUNTS
android.permission.INSTALL_PACKAGES
com.android.launcher.permission.UNINSTALL_SHORTCUT
com.android.launcher.permission.INSTALL_SHORTCUT
Weather

Figure 2. Permissions' graphs obtained for the categories Photography, Wallpaper, Finance and Weather.

categories possessed less than 20 significant pairs; these were very specific.

### *B. Significant patterns and centrality results for a category samples*

Due to page number restrictions, we present the results related to only four of the categories – Photography, Wallpaper, Finance and Weather. We have chosen categories with an average number of relevant pairs. Corresponding patterns are highlighted in Figure 2. On each graph, the colour of the nodes is defined by the result of the modularity-based clustering algorithm. The weight of the link corresponds to the $\mathcal{Z}-score$, and the size of the nodes is proportional to the betweenness centrality. Graphs and data for all categories are available on [22].

*Photography:* As one would assume, 'CAMERA' is a central permission within the photography category. It is used with the 'READ_EXTERNAL_STORAGE' and 'WRITE_EXTERNAL_STORAGE' permissions, which allow photos taken to be saved and modified. The 'ACCESS_NETWORK_STATE', 'ACCESS_WIFI_STATE' and 'INTERNET' permissions enable photo sharing.

'SET_WALLPAPER' is the second most important permission, which allows the photo to be added as wallpaper on the main screen. We can distinguish a pattern grouping together wallpaper management permissions: 'SET_WALLPAPER_HINTS' and 'BING_WALLPAPER'.

'WAKE LOCK' prevents the screen from locking when the application is in use. This functionality seems relevant in camera-related applications.

Many applications in this category allow screenshots to be taken as well as photos. Shortcut management permissions allowing the creation of shortcuts can be used to take photos as well as for screenshots.

The presence of location-related permissions indicates that this information will be attached to the picture taken. The 'GET_ACCOUNTS' permission corresponds to a server-based user-specific service which probably backs up the photos taken on the server or shares photos with different services, such as social networks.

We noted an increased presence of system permissions that are not available for third-party applications. This indicates that many photography applications are built-in.

*Wallpaper:* The results for the Wallpaper category (APP_WALLPAPER) give a high number of significant permissions due to the diversity of animated wallpapers and the functionalities accessed and provided by animated wallpapers.

'SET_WALLPAPER'is the most central permission; we also find the wallpaper-related permissions in the pattern. File system and package management permissions can be observed, due to the different personalisation options proposed by a single wallpaper application, as well as shortcut and widget management permissions. We find many functionality-related permissions due to the different built-in functionalities: phone calls, SMS, calendar, settings, application list, contacts, bookmarks, cache – those functionalities are often included as a widget or fast access to wallpaper. External storage permissions allow personalisation images to be stored locally, and network-related permissions allow additional information such as weather to be obtained or new images downloaded. We also

TABLE II
NUMBER OF RELEVANT COUPLES OF PERMISSIONS ($\mathcal{Z} - score > 2$) FOR EACH CATEGORY OF APPLICATION

| Category | # of relevant couples | Category | # of relevant couples |
|---|---|---|---|
| COMMUNICATION | 3,620 | WEATHER | 124 |
| TOOLS | 2,826 | PHOTOGRAPHY | 116 |
| APP_WIDGETS | 2,318 | ARCADE | 106 |
| PRODUCTIVITY | 2,306 | MEDIA_AND_VIDEO | 92 |
| BUSINESS | 1,028 | CASUAL | 84 |
| PERSONALIZATION | 1,024 | RACING | 84 |
| LIFESTYLE | 738 | SPORTS_GAMES | 80 |
| SOCIAL | 738 | SPORTS | 78 |
| APP_WALLPAPER | 548 | TRANSPORTATION | 74 |
| TRAVEL_AND_LOCAL | 322 | SHOPPING | 66 |
| ENTERTAINMENT | 242 | COMICS | 64 |
| MEDICAL | 160 | BOOKS_AND_REFERENCE | 60 |
| HEALTH_AND_FITNESS | 158 | CARDS | 48 |
| LIBRARIES_AND_DEMO | 152 | GAME_WIDGETS | 46 |
| MUSIC_AND_AUDIO | 144 | NEWS_AND_MAGAZINES | 20 |
| FINANCE | 126 | GAME_WALLPAPER | 18 |

identify the 'WAKE_LOCK', 'ACCESS_FINE_LOCATION' and 'VIBRATE' permissions in this category.

*Finance:* The most central permissions for the Finance category are 'CALL_PHONE' and 'INTERNET'. Permissions used for calls, including Voice over IP (VoIP) calls and SMS, available to contact a bank or service manager. The 'INTERNET' permission would appear to be necessary in order to access up-to-date banking information. We can distinguish many account- and authentication-linked permissions due to the sensitivity of the financial information and the need for secure usage. Localisation permissions also appear in the pattern, probably to apply different location-dependent billing criteria or to identify the nearest offline office. The camera is often used for QR codes and making deposits in finance applications.

*Weather:* The central weather permissions is 'ACCESS_FINE_LOCATION', which gives the longitude and latitude so that the weather in the user's location can be obtained. All location- and network-related permissions are included in the pattern. 'ACCESS_MOCK_LOCATION' could indicate developer testing or be for locations given by the user. One can also see background process, shortcut and wallpaper permissions, which indicate that the weather application can be wallpaper-embedded. Permissions related to external storage are needed for heavy image storage. Weather applications are often system applications, and some system permissions are observed in the pattern.

## VI. DISCUSSION AND FUTURE WORK

The state of the art's most commonly used permission indicator is the simple occurrence of permissions. To underline how our methodology has improved this, we proposed comparing our results to the top 5 most frequent permissions obtained for the same category.

We present the 'Finance' category as an example. Table III presents the top 5 permissions according to occurrence, and Table IV presents the top 5 permissions according to betweenness centrality. One can see that the top 5 'Finance' permissions of Table III correspond to the top 5 permissions for all categories presented in the Table I. This shows that even if those permissions are highly used in the 'Finance' category, they are not specific to it.

TABLE III
TOP 5 FREQUENT PERMISSIONS FOR FINANCE CATEGORY

| | Occurrence (%) | Betweenness |
|---|---|---|
| INTERNET | 91.19 (Rank 1) | 361 (Rank 2) |
| ACCESS_NETWORK _STATE | 75.15 (Rank 2) | 202 (Rank 6) |
| WRITE_EXTERNAL _STORAGE | 49.32 (Rank 3) | 98 (Rank 16) |
| READ_EXTERNAL _STORAGE | 49.12 (Rank 4) | 98 (Rank 17) |
| READ_PHONE _STATE | 32.88 (Rank 5) | 69 (Rank 19) |

TABLE IV
TOP 5 PERMISSIONS ACCORDING TO BETWEENNESS CENTRALITY FOR FINANCE CATEGORY

| | Occurrence (%) | Betweenness |
|---|---|---|
| CALL_PHONE | 11.74 (Rank 12) | 470 (Rank 1) |
| INTERNET | 91.19 (Rank 1) | 361 (Rank 2) |
| CAMERA | 10.96 (Rank 14) | 360 (Rank 3) |
| USE_CREDENTIALS | 3.52 (Rank 21) | 257 (Rank 4) |
| ACCESS_COARSE _LOCATION | 19.18 (Rank 8) | 231 (Rank 5) |

We noted that our pattern contains these permissions, but not as highly ranked; the top five permissions from our results (Table IV) show that 'Finance' permissions are often online (Internet) services and need secure authentication (use credentials). Banking applications tend to include direct bank-application contact (call phone), deposits (camera) and lists of office or cash withdrawal locations (access coarse location).

Our pattern is more accurate than simple frequency analysis in defining a particular category and allows category-related functionalities to be detected. The use of the $\mathcal{Z} - score$ is particularly well-adapted to this purpose, since it allows how relevant a permission pair is to a category to be measured with respect to overall usage in units of standard deviation.

We observed many wrong, misspelled or old permissions in the applications. We feel that the system would benefit from automatic permission validity verification based on the list of valid Android permissions; rules for defining custom permissions could simplify the verification. Documentation for the Android permission system is incomplete, as many Android 4.4 extracted permissions were not found on the official website.

When we observed permission patterns by category, they often represented a particular functionality. This could permit the purpose of permission usage and the functionalities of an application to be determined automatically.

Some categories obtained a very large number of significant permissions, which means they may have been too broad. The division of these categories into subcategories would provide a more precise view of the applications.

Our patterns could permit an automatic classification of applications into categories and could also be used to measure how an application rates with regard to normal permission usage in a particular category. Applications using non-core or rare permissions can be penalised. Such indicators could be included in mobile markets to label abusive or non-threatening applications, comparing them to expected patterns.

## VII. Conclusion

We analysed Android permission usage for each application category belonging to the Google Play store. We proposed a graph-based solution to characterise each category using patterns of the most significant permissions, taking into account the category and the overall usage of each permission combination. We scored permissions using betweenness centrality in order to obtain the most- and least- central permissions for each category. The identified patterns and permission scores could be used in the mobile market to detect abusive or non-threatening applications.

## References

[1] P. Kelley et al., “A conundrum of permissions: Installing applications on an android smartphone,” in Financial Cryptography and Data Security, ser. Lecture Notes in Computer Science, J. Blyth, S. Dietrich, and L. Camp, Eds. Springer Berlin Heidelberg, 2012, vol. 7398, pp. 68–79.

[2] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, “Android permissions: User attention, comprehension, and behavior,” in Proceedings of the Eighth Symposium on Usable Privacy and Security, ser. SOUPS ’12. New York, NY, USA: ACM, 2012, pp. 3:1–3:14.

[3] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, “Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones,” in Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, ser. OSDI’10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–6.

[4] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall, “These aren’t the droids you’re looking for: retrofitting android to protect data from imperious applications,” in CCS ’11: Proceedings of the 18th ACM conference on Computer and communications security. ACM Request Permissions, Oct. 2011.

[5] P. Berthomé and J.-F. Lalande, “Comment ajouter de la privacy after design pour les applications Android? (How to add privacy after design to Android applications?),” Jun. 2012.

[6] R. Pandita, X. Xiao, W. Yang, W. Enck, and T. Xie, “Whyper: Towards automating risk assessment of mobile applications,” in Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13). Washington, D.C.: USENIX, 2013, pp. 527–542.

[7] Z. Qu, V. Rastogi, X. Zhang, Y. Chen, T. Zhu, and Z. Chen, “Autocog: Measuring the description-to-permission fidelity in android applications,” in Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS ’14. New York, NY, USA: ACM, 2014, pp. 1354–1365.

[8] Y. Agarwal and M. Hall, “Protectmyprivacy: Detecting and mitigating privacy leaks on ios devices using crowdsourcing,” in Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services, ser. MobiSys ’13. New York, NY, USA: ACM, 2013, pp. 97–110.

[9] D. Barrera, H. G. Kayacik, P. C. van Oorschot, and A. Somayaji, “A methodology for empirical analysis of permission-based security models and its application to android,” in Proceedings of the 17th ACM Conference on Computer and Communications Security, ser. CCS ’10. New York, NY, USA: ACM, 2010, pp. 73–84.

[10] F. Roesner, T. Kohno, A. Moshchuk, B. Parno, H. J. Wang, and C. Cowan, “User-Driven Access Control: Rethinking Permission Granting in Modern Operating Systems,” Security and Privacy (SP), 2012 IEEE Symposium on, 2012, pp. 224–238.

[11] P. H. Chia, Y. Yamamoto, and N. Asokan, “Is this app safe?: A large scale study on application permissions and risk signals,” in Proceedings of the 21st International Conference on World Wide Web, ser. WWW ’12. New York, NY, USA: ACM, 2012, pp. 311–320.

[12] W. Enck, M. Ongtang, and P. McDaniel, “On lightweight mobile phone application certification,” in Proceedings of the 16th ACM Conference on Computer and Communications Security, ser. CCS ’09. New York, NY, USA: ACM, 2009, pp. 235–245.

[13] A. P. Fuchs, A. Chaudhuri, and J. S. Foster, “SCanDroid: Automated Security Certification of Android Applications,” Department of Computer Science, University of Maryland, College Park, Tech. Rep. CS-TR-4991, November 2009.

[14] M. Frank, B. Dong, A. P. Felt, and D. Song, “Mining Permission Request Patterns from Android and Facebook Applications,” in Data Mining (ICDM), 2012 IEEE 12th International Conference on, 2012, pp. 870–875.

[15] W. Wang, X. Wang, D. Feng, J. Liu, Z. Han, and X. Zhang, “Exploring Permission-Induced Risk in Android Applications for Malicious Application Detection,” Information Forensics and Security, IEEE Transactions on, vol. 9, no. 11, 2014, pp. 1869–1882.

[16] V. Moonsamy, J. Rong, S. Liu, G. Li, and L. Batten, “Contrasting Permission Patterns between Clean and Malicious Android Applications,” in Future Generation Computer Systems. Cham: Springer International Publishing, 2013, pp. 69–85.

[17] I. Rassameeroj and Y. Tanahashi, “Various approaches in analyzing Android applications with its permission-based security models,” in Electro/Information Technology (EIT), 2011 IEEE International Conference on, 2011, pp. 1–6.

[18] O. V. Koc, “android-market-api-php.” [Online]. Available: https://github.com/splitfeed/android-market-api-php[accessed:13-01-2015]

[19] H. Shiokawa, Y. Fujiwara, and M. Onizuka, “Fast algorithm for modularity-based graph clustering.” in AAAI, M. desJardins and M. L. Littman, Eds. AAAI Press, 2013.

[20] M. E. J. Newman, “Fast algorithm for detecting community structure in networks,” Physical Review E, vol. 69, no. 6, 2004, pp. 066 133–066 133.

[21] L. C. Freeman, “Centrality in social networks conceptual clarification,” Social Networks, vol. 1, no. 3, 1978, pp. 215–239.

[22] C. Perez and K. Sokolova, “Android permissions usage data.” [Online]. Available: https://sites.google.com/site/androidpermissionsanalysis/[accessed:14-01-2015]

# Using Patterns to Guide Teachers and Teaching Materials Evolution

George Moroni Teixeira Batista, Mayu Urata, Mamoru Endo and Takami Yasuda
Graduate School of Information Science
Nagoya University
Nagoya, Japan
tenno.kun@gmail.com
mayu@gsid.nagoya-u.ac.jp
endo@is.nagoya-u.ac.jp
yasuda@is.nagoya-u.ac.jp

Katsuhiro Mouri
Astronomy Section
Nagoya City Science Museum
Nagoya, Japan
mouri@nagoya-p.jp

***Abstract* - Computer-novice teachers can face various problems when using e-Learning systems. Computers and the internet are a vital part of our lives nowadays, and teachers cannot avoid using this kind of technology in reaching their students. The problem became more interesting when we try to find how can the teachers that have little experience with computers be helped to handle this technology? Usually institutions that have e-Learning platforms have separated technical group to operate the technology, however it takes too much time for the contents to be updated, and the students that really needed the updates may never see it. The employment of utilisation patterns can help teachers learn from the teaching experiences of their colleagues that have more experience with computer as they use the system, allowing them to evolve and become better professionals by learning, for example, how their colleagues are overcoming device or system interface barriers. The implementation of a system that allows the teachers themselves create and update content in real-time, allowed the teachers to adjust teaching materials which were out of context for the students, making them easier to understand in time for the students that really needed.**

***Keywords-e-Learning; teaching; materials; evolution; pattern.***

## I. INTRODUCTION

This research background began with the ELO (Escola de Línguas Online) project. The project was started in August 2007 by three undergraduate students from the Department of Foreign Languages and Translation of Brasilia University in Brazil. As stated by Boggs and Jones [1], "the Internet is a professional development tool for teachers", and they should take advantage of what this kind of technology has to offer. Following this idea, the objective of the project was to create an online platform for the Japanese language course to allow the students to interact, do activities online and review the content of the face-to-face lessons. After approximately six months the project was accepted by the University and became an official University project with the support of the Department's teachers. Afterwards, the other languages taught in the department were incorporated into the project: a total of seven languages, Japanese, English, Spanish, Portuguese, French, German and Italian. The development team had eight members, there were twenty-five teachers with the help of two tutors each in a total of fifty-five lectures, and there were more than four thousand users.

Standard CMS (Content Management System) software like Moodle were used at the beginning of the project, but there were several problems, especially with the content editing interface. Another problem faced by teachers was the fact that they could not update or make versions of the teaching materials to make them more suitable for the requirements of their real classes. To solve these problems, an original system was created using the Adobe Flex framework for the content navigation and edition. The system allows the teachers to edit the content in a WYSIWYG (What You See Is What You Get) interface in real-time, create versions, and share and manage teaching materials.

In the current stage of the research, we are also focusing on the possibility of using the system in different fields of study. To achieve this, we began working with the Nagoya City Science Museum in Japan. The Museum produces many kinds of content for visitors, for the Museum website, and for journals and magazines. Because of the great variety of the Museum's public, it also needs a system to create and manage the different versions of the content, as stated by Iwazaki et al [2]; also, having a different theme every month, the Museum needs to create different versions of the contents to match the different themes and guide each type of visitor to the right place. This makes the Museum a good environment in which to test and evaluate the system content management features.

Until now the research focused on the creation of the editing interface and the database to store and manage the versions of the teaching materials. However, since many teachers were using computers for the first time, or had very little experience, those features were not enough to help them learn how to handle the technology. Therefore, to achieve that goal, we are now focusing on the creation of utilisation patterns that the users can use as guidelines to help them to learn about the system features, gain editing skills, and make decisions to solve problems based on the experience of other users, helping them to become better teaching professionals.

ISBN: 978-1-61208-393-3

This paper is divided in six sections, this section is the Section 1, the introduction. In Section 2, there will be the description of the research objectives. In Section 3, the system concept will be explained. The Section 4 describes examples of teaching materials evolution patterns, and how the users can use them to learn more about the system. The Section 5 is about the evaluation tests and its results. Finally, in the Section 6, the conclusion about the current stage of the research and what is expected to be done from now is discussed.

## II. OBJECTIVE

The research objective is to define patterns of utilisation that allows users to improve their abilities by observing other users' experiences; the other users' experiences can be observed through the visualisation of content evolution patterns stored in the system database. As stated by Aral [3], "social influence as part of a dynamic system in which a variety of feedback loops continuously affects behaviour in a constantly evolving fashion". Other users' feedback can allow the users to become better professionals as they apply what they get from that feedback to their work.

In this research, we created a DTMS (Dynamic Teaching Materials System). The system can be used to create teaching materials that can be edited in real-time even during the classes. This allows the users to easily apply the necessary updates to the contents as soon as they receive feedback or realise something from their own experience. All the updates and versions of the contents are stored in the database, as well as the contents' evolution patterns, and they can be accessed and used by other users.

## III. SYSTEM CONCEPT

In the beginning of the project, we tried using standard CMS software, like Moodle [4], Joomla [5], eFront [6] and Dokeos [7]. They were good for managing course access and user accounts, but the interface was too complicated for most of the users to create content or even to navigate. Since many of the teachers were using computers for the first time, or had too little experience, this kind of problem was unavoidable. Additionally, it was also observed that, as stated by Jones and Lynch [8], "web-based systems need to be adapted and evolve following an educational context that is continually changing"; teaching materials need to be updated continuously, to prevent them from became outdated for the new students every semester, but this was very difficult as the teachers could not perform the updates by themselves because of the interface; or, as in the case of interactive content, they needed to know some programming language to make the necessary changes.

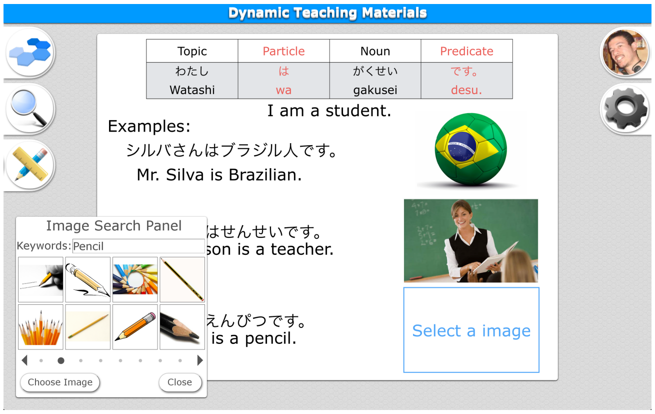
The teachers needed at least four features: a WYSIWYG interface that could create interactive and multimedia content, real-time editing to allow the updates to reach the students who really needed them, content version management for the different types of classes and students, and all the necessary functions to edit and manage the content in one screen, because teachers do not have time to spend navigating through the platform during class. We could not find a standard CMS that had all the required features at the same time, and the feature of editing the content in the same screen in which the content is being visualised was not to be found in any of the CMS we tested. Additionally, as stated by Alexander [9], the fact of teachers cannot handle new technologies can become a great barrier in the development of e-Learning projects. Further as stated by Roberts, Romm and Jones [10], nowadays technologies like computers are a basic part of students daily life, and teachers should be able to handle them; however, since it is not their speciality, we had to find a way to help teachers to learn how to handle the technology.

To solve the problems we propose an original system for the content edition and version management, the DTMS. As stated by Gillani [11], since language teachers are not information technology specialists, the editing interface needed to be very simple, allowing the users to do almost everything with just the mouse, or even to create, interactive content without programming languages; the simplicity of the new editing interface allows the users to make updates in real-time, in the same screen they use to show the content to students; no navigation is needed, and all the updates and versions are stored in the database. As stated by Krug [12], "it is really important to make the interface uniform in order that the users do not get lost in some parts of the application due to the different patterns of navigations or the menus". This approach was used to create menus to access contents, which helps stop the users from getting lost in the long link lists and other different navigation patterns in the Moodle interface.

Technologies like Java, PHP, Adobe Flash, and Adobe Flex were tested, and the Adobe Flex framework was the technology that best fitted the requirements of the project; because Flex is free, most web browsers come with the necessary plugin by default, and the users do not need to download, install, or configure anything by themselves. The database was created with MySQL [13] and the connection between the Flex [14] interface and the database is in PHP [15].

A teaching material created using the DTMS is called a dynamic teaching material because it can be easily changed to become more suitable to the learning environment. A teaching material like a book is static: if it needs some kind of update the owner will have to buy another book that is a newer version of the book he/she already has; however the DTMS allows the user to create teaching materials that are dynamic. The teaching material itself can be updated, and more importantly, the teachers using the system can make the necessary updates themselves according to the necessities they have during classes; they do not need to wait for the book editors and publishers to release an updated version that maybe solve the problems they were having. In other words the dynamic teaching materials can evolve to meet the needs of the teachers and their students in real-time, in such a way that the updates will not be too late for the students that really need them.

System features for user feedback tasks, are still being developed. However, the teachers can basically use the system to create content that looks like  slides, as shown in Figure 1. The user can insert text, tables, arrows and other graphic shapes, as well as images and videos from his/her own computer or from the internet using the search feature as shown in Figure 1, which allows the user to search images and videos from inside the system interface.



Dynamic Teaching Materials

| Topic | Particle | Noun | Predicate |
|---|---|---|---|
| わたし<br>Watashi | は<br>wa | がくせい<br>gakusei | です。<br>desu. |

I am a student.

Examples:

シルバさんはブラジル人です。

Mr. Silva is Brazilian.

はせんせいです。

son is a teacher.

えんぴつです。

is a pencil.

Image Search Panel

Keywords: Pencil

Figure 1. System’s editing interface.

All the elements inserted in the slide can be configured to be interactive, responding to mouse clicks, drag and drop, and can also influence each other. For example, if the user drops a word in to another element, the element that received the drop can change to an image that represents the meaning of the word. The entire configuration is done through the system’s GUI (Graphical User Interface), and no programming language is required. After the slides are completed, they can be grouped in folders called “books”, and the “books” can also be grouped in folders called “courses”. That way the teacher can create the slides as if they are book pages, and group the books he/she wants to use as teaching materials to create a complete course.

The system utilisation process can basically be divided into three phases: the creation phase, the sharing phase and the teaching phase, as shown in Figure 2.

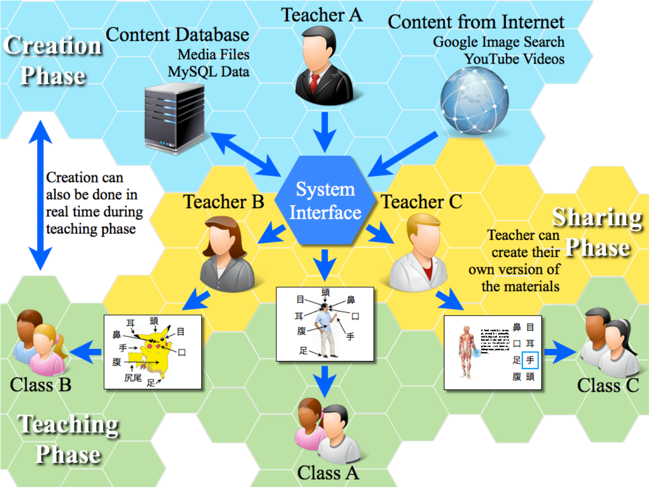


Figure 2. Dynamic Teaching Materials System.

The three phases are connected, creating a utilisation cycle that restarts after the third phase. In the case of a teacher, for example, the three phases cover the user activity from the preparation of the teaching materials to the utilisation of the teaching materials in the classes and getting feedback.

In the creation phase, the user can create a new original teaching material, or choose one that another user has already created, and edit it if necessary. The user can create or choose the teaching material version that is most appropriate for his/her teaching skills, the kind of lesson he/she pretend to do, and the kind of students who are going to participate. When creating teaching materials the users can make a variety of multimedia and interactive content, or something simpler like just text and images; it depends on his/her editing skills and what he/she thinks that is more appropriate. Even if the system has many editing features and an easy to use interface, the teaching materials instructional design is up to the user; the system cannot automatically help with this part since the system cannot predict the kind of situation in which the content will be used. However, before creating something new, the user can use the search feature to see if someone has already created the kind of content he/she needs, and use it as a template to help get some design ideas.

After finishing the content creation or editing, the users can tag the contents when saving it in the database. Tags are used for search, and also to help other users understand the scope of the teaching material; for example, the topic of the teaching material, the language, whether it is interactive, and the target students. Since each update and version of the content can have individual tags it helps the user searching for the content to choose the right version of the right content.

The second phase, the sharing phase, starts once the teaching material is saved in the database. As soon as the content is in the database it is being shared and can be accessed by anyone in the system. Users can access any content in the system database to study, use as a teaching material in the classroom, use as a template to create a new teaching material and so on. All the content created and stored in the system's database fall under the creative commons licence “Attribution-NonCommercial-ShareAlike 4.0 International”; everything can be shared and adapted, and the system keeps track of the authors of every version of all content automatically. In the database table, the content has columns to store the author's ID, the ID of the original content if the content is a version, and for updates it stores the ID of the first release of the content, and a date for every entry in the database. With this data the system can show who created the content, when the content was created, list all the updates, when the updates occurred, show all the different versions of the content, and who created the versions and when. This gives a complete view of what has happened to the content from its first release to the current time.

The teaching phase is the moment when the users actually use the content created using the DTMS to teach in a face-to-face class or online. In this phase the user can check if the teaching material is really suitable for his/her class style and target students. During the class the teacher can stll make changes in real-time if there is some problem found, createing a live update that will be available to the other users as soon as it is saved, like any other content in the system. During or after the class, the user can get feedback from other users and students about the content and the way it was used; this feedback can be comments during the class, emails sent to the user, or even comments in the comment space inside the DTMS interface. The user can make improvements in the content based on that feedback, allowing the content to evolve as it is used by him/her or by other users. When making improvements to the content the user starts to edit the content again, and that restarts the cycle from the creation phase again.

Inside the system interface, feedback occurs as in an online social networking system, where users can comment on the contents, mark what they liked, mark their favourite contents; and the system also stores the number of times each content was accessed. Combining this data with the historic view of the content updates, the system can show the users the pattern of the content's evolution, where they can see which version of the content was more popular, more accessed, the way it was used, if the user feedback, especially comments, had some influence in the subsequent updates or creation of new versions, and so on, showing how the content started, changed over time, what caused those changes, and what the public response was.

## IV. Evolution Patterns

The DTMS already allows the teaching materials to evolve by being adapted to the learning environment as necessary for the users. Now, we are trying to find a way to use the content evolution pattern to help the users evolve, learning more editing skills and content utilisation techniques by observing other users experience. This can allow the users to create better content and also update the old content to make them better using new acquired skills, making the contents evolve as well. As stated by Jones, Sharonn, and Power [16], the utilisation of patterns emphasises the experience and reuses it to help the users and the development team to learn from what happened in the past and be more prepared for possible problems, allowing them to evolve.

The teaching material evolution pattern is basically how the content is changed over time to try to solve the problems faced by the users using it. Since the same content can be used in different situations, it can face different problems and it can result in the creation of different versions, one for each problem solution. The teaching material can also be used by different teachers, and even if they face the same problem, the way they will try to solve the problem may vary, also resulting in different versions, one for each solution. This information is what compounds the teaching material evolution pattern. In other words, the teaching material evolution pattern shows the problems faced by each version or update of the teaching material and how it changed to try to solve the problems being faced.

The evolution pattern visualisation and some feedback features are still in development; however it is possible to exemplify the dynamic teaching materials evolution pattern and how the users can use it to learn more, by observing what has already happened naturally to some contents, Figure 3 demonstrates the evolution pattern of the content shown in Figure 1.

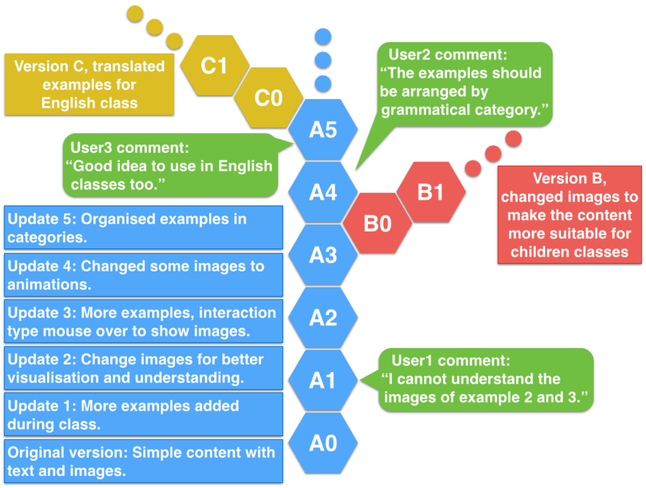

Version C, translated examples for English class
C1
C0
User2 comment: "The examples should be arranged by grammatical category."
A5
User3 comment: "Good idea to use in English classes too."
A4
B1
B0
Version B, changed images to make the content more suitable for children classes
Update 5: Organised examples in categories.
A3
Update 4: Changed some images to animations.
Update 3: More examples, interaction type mouse over to show images.
A2
Update 2: Change images for better visualisation and understanding.
A1
User1 comment: "I cannot understand the images of example 2 and 3."
Update 1: More examples added during class.
Original version: Simple content with text and images.
A0

Figure 3. Content evolution pattern with users' comments.

The content "A0" starts as a simple content with sentences using the new vocabulary students are learning in a Japanese class; the content also has images to illustrate the meaning of the sentence or its main word. When using the content in the class, the teachers had to add more examples to help with the explanation, and it generated the content's first update, "A1". After some time, the "user1" made a comment saying that some of the images used in the content were not clear, and to solve that problem the teacher changed the images to ones that were more easy to understand, generation the second update. After that the teacher decided to add more examples with text and images, but there were too many examples with images to show them all at the same time, so the teacher decided to put only the text and use the mouse-over interaction feature to show only the image of the sentence being read, thus saving space on the screen to contain all the examples at the same time.

After the third update, a different teacher liked the content "A3" and wanted to use it in his classes for children; however he thought that he needed more images for the children, so the he used the content "A3" as a template to create the content "B0", a version with the same texts but different images to make the content more suitable for children. The fourth update was made by changing some of the images to animations, for the best understanding of words like action verbs. After the fourth update, the "user2" made a comment stating that the examples should be arranged by categories; to solve that the teacher simply changed the positions of the examples on the screen to put the examples of the same category together, generating the fifth update. After that,“user3”, an English teacher, made a comment saying that the content could be good for English classes as well, and then he created an English version of the content translating the example sentences but keeping the images.

In this evolution pattern sample, we can see how the content changed over time and why the updates happened, when the content was updated in real-time to meet necessities during classes as in update 1, or updates because of user feedback like update 2, or updates because the author himself/herself detected a problem like in update 3. We can also see how the content was adapted to be used in different situations with different target students by other users.

Another example of a content evolution pattern can be seen in the context of the Nagoya City Science Museum, where the next evaluation tests are going to be held. In the case of the Museum there will be, for example, content to explain about the observation of stars (see Figure 4). The Museum curator needs to use the sky simulator function to generate the starry sky image and then apply filters to show only the necessary stars in the right size, form constellations connecting stars, and input some text labels for more detailed explanations about the stars movement or visible time periods, for example.

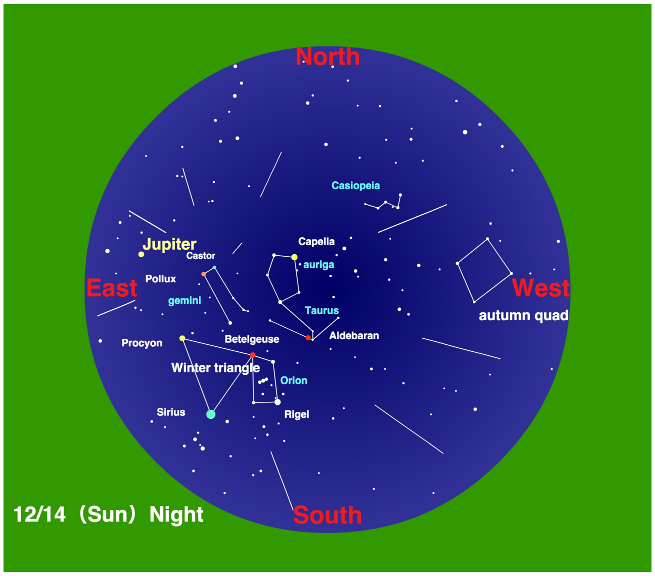


Figure 4. Content naming visible stars in the night sky

Once the content is saved in the database it can be reused as a template to create new content or updated if needed. For example, if the curator needs to show the same constellations but on a different day or time, he does not need to create everything from the scratch, he can just load the content that already has the necessary identifications and explanations, and simply reload the position of the stars for the new day or time, creating a new version of the content for a different date. Most of the content in the Museum is created in Japanese, but since the number of foreign visitors is high, the curators also need to create English and/or Chinese versions of the contents.

In this case, the changes that can happen are not necessarily improvements in the content, but adaptations to make it usable in different situations. The evolution pattern can show, for example, how the labels of the stars needed to be moved when changing the date; thus the curators can try to find better ways to place the labels.

The utilisation patterns being used as user guidelines should have the best practices for the users on two sides: the side of the user who created the content, and the side of the user who is observing the content’s evolution pattern, in a way in which the content’s author can really transmit his/her experience in an intelligible manner, and in which the other users can understand how to access the data stored in the database to actually see the evolution pattern. To help the users to better transmit their experiences and use the evolution patterns to learn and evolve, we suggest a utilisation patter based on the following points:

- Before creating new content, search the database to see if an other user has already created the necessary content; the user can create a version of another user’s content to best fit his/her needs if necessary.
- Every time a user creates or updates a content, he/she should at least use tags that describe the topic of the content, the target students and the content type.
- The author can use individual tags to show the differences between content updates and versions.
- Pay attention to the feedback of other users; they can highlight problems not yet found, or even show better ways to use or edit the content.
- Every time a user makes an update or a version of a content, he/she should write a comment explaining the objective of the update, what kind of problem he/she wanted to solve with the changes, as in the Figure 3 example updates list.
- The user comments are linked to the specific update that was being accessed by the user; it is useful to observe how the comments influenced the subsequent updates and versions.
- Before creating new content, use the search to see how other users are creating the same kind of content; the comments in the previous updates can reveal about possible problems faced by the users, and the updates themselves can show how the author tried to solved the problem. This can help prevent the same kind of problem from happening again.
- Take the time to see versions of your own content created by other users; they might contain hints on how to solve problems, as well as new edition and utilisation techniques.
- Whenever possible, give feedback to the author; it helps him/her to understand the pros and cons of his/her work.

There will be evaluation tests to finish defining the utilisation patterns that can help the users as guidelines for evolution, and also to make adjusts to the interface and check if there any new functions are necessary for the system. Since the beginning of the ELO project the development team was working side-by-side with the users to see exactly what they needed, and built the system based on the necessities that the users faced in their work. As stated by Vianna, Vianna, Adler, Lucena and Russo [17], having the development team enter the context of the users is very important in allowing the creation of something really original, and in the case of the Nagoya City Science Museum, we expect to keep doing the same kind of collaborative work, checking the real needs the Museum curators have in order to create the necessary contents.

## V. Evaluation Tests and Results

From the beginning, the system has been developed in collaborative work between the development team and the users; whenever possible there have been interviews and reunions conducted between the users and the development team to discuss the system's performance, what is good, what needs to be fixed and why the changes are needed.

The first test environment can be seeing in Table 1; the test was performed at the beginning of the ELO project when all the languages of the Department were being incorporated into the system. The teachers gave their evaluation of the system in interviews after using it in face-to-face classes and in online support activities. The test was focused on new navigation patterns for using the system in the face-to-face classes, what kind of teaching materials the teachers really needed to create or edit, and to evaluate the necessity of tutors in helping teachers prepare and set up the necessary equipment to use the system in the classroom, especially in the case of teachers with little experience with computers.

TABLE I. Brasilia University first test environment

| | |
|---|---|
| **Test period** | from February 2008 to April 2009. |
| **Development Team** | 8 members. |
| **Languages** | Japanese, Spanish, French, English, Italian, Portuguese and German. Total of 7 languages. |
| **Lectures** | 55 lectures, graduate and undergraduate level. |
| **Teachers** | 25 teachers. |
| **Tutors** | 2 per teacher, total of 50 tutors |
| **students** | 2907 students |
| **Total of users** | 2982 users |

The new content interface that implemented a new navigation pattern solved the problem the teachers and student were having in navigating the platform to access other contents. Before with the standard Moodle interface, the users could view the courses and the contents inside the courses as lists of links, but sometimes the lists became too long and it was difficult to find the contents, difficult to see which one was used in the class for revision, and also difficult to see which contents were related. The new interface had all the content in one screen, accessed by menus and submenus, organised by topic, which no need to navigate to different pages or areas of the online platform. The new interface also helped the teachers when using the system in face-to-face classes because the projectors had a screen resolution smaller than the normal displays, usually 800x600 pixels, and the new interface was designed to fit this smaller size; since no navigation was needed after entering the course page, the teachers had all they needed on the screen, saving time when showing the contents.

Even the teachers who had experience with computers stated that the support of the tutors to set up the equipment in the classroom was essential, since in Brasilia University the number of classrooms equipped with computers and other multimedia equipments is very low; almost every time, the teachers had to bring the equipment and set it up in normal classrooms, which is very time-consuming to do alone, and is unfeasible taking into consideration the total time for the class.

Regarding the types of teaching material needed by the teachers, the most required types for the face-to-face classes were text with images to illustrate the meaning of new vocabulary, animations or videos to illustrate action verbs and dialogue examples; for grammatical explanation the most popular contents were interactive contents that, for example, showed how the use of different grammatical forms changed the meaning of the sentence as the user moves the mouse cursor through the available options. The test results were very good for the stage of development that was focused on solving the problems the teachers were having with the Moodle interface.

The second test was conducted with a smaller group, just six Japanese teachers, twelve tutors, and one hundred and forty students in nine lectures. The test was conducted from July 2009 to December 2011, and focused on the real-time content editing interface, the creation of content versions, and how to store the versions on the database. During the period of the first test, the teachers made the instructional design of the contents, but most of the content was developed by tutors that had experience with computer, or were trained by the development team.

The system had good approval from the language teachers; they made comments like "the system was really good because we can do almost everything with just some mouse clicks", "the real-time editing makes the system work like an electronic blackboard" and "the possibility of creating different versions of the teaching materials in real-time solves the problem of teaching materials which are out of context for the students”. However we still had some problems because at that time the system did not yet have a content search engine; to use the content, the teachers needed to know the content name or the ID number to load it from the database. It was difficult for the teachers to know which content had versions and which version was the best to be used in their classes since the system did not have an interface to show the relationship between the contents' versions.

ISBN: 978-1-61208-393-3

After fixing the problems found by the tests, the current development stage is focused on content version management, the interface for showing the relationship between the content versions and the interface for showing the evolution pattern of the content. Future tests of the system will be done in collaboration with the Nagoya City Science Museum. We have already had some meetings with the Museum curators to discuss the necessary new functions and create a database structure that can store the changes and track the different versions of the contents. The next test will focus on the creation of different versions of the content, storing the creation process, and trying to define the best interface to show the evolution patterns of the contents, in a way that the users can see it, understand what has happened to the content and learn from the content authors' experience.

## VI. Conclusion and Future Work

The development of the system still has a long way to go. With the results of the tests done up to now, it is possible to say that the system has successfully accomplished the interface tasks of an easy to use real-time editing interface, creating an environment that allows teaching materials to evolve, and can also be used in different fields of study. All the problems faced by the teachers at the beginning of the project were solved with the implementation of the new content editing interface. The system allows the users to edit content, update it in real-time and create versions of the content, so that it can be used in different learning situations. However the content management part is still in its initial development, especially the interface.

The content search function that was one of the requests after the first test has also been completed, allowing the users to search for content using keywords, and even allow them to search for the content's previous updates and different versions. This is a useful feature but is not enough to show the evolution pattern of the contents to the users. The system features for user interaction and content feedback are also still in development. As stated by Grant, Facer, Owen and Sayers [18] and by Paiva [19], communication is a very important part of the learning process; hits is not just between teachers and students, but can also be applied to the situation of users learning new techniques from each other to use the system, and is what makes these features key to promoting the evolution of the contents and users.

After the implementation of the feedback features, the system needs to be tested to define the utilisation patterns that allow users to use the evolution patterns of the dynamic teaching materials as a learning tool to improve their abilities to edit and use teaching materials based other users experiences. After the implementation of all the features, the system should allow the users to create and share teaching materials, use those contents to teach, and interact with one another, giving feedback about the content they edit and use. Through these interactions, users can learn from one another's experience and help each other to evolve, becoming better teaching professionals who are more prepared for the different situations they may face during work.

## Acknowledgment

This work was supported by JSPS KAKENHI Grant Numbers 25280131, 24800030, and THE HORI SCIENCES AND ARTS FOUNDATION.

## References

[1] E. Boggs and D. Jones, "Lessons learnt in connecting schools to the internet", Australian Educational Computing, Vol. 9, No. 2, pp.29–32, Australian Council for Computers in Education, 1994.

[2] K. Iwazaki, et al. "Possibility and the Trial based on "connection" between the Science Museum and Universities or Visitors : Development of Exhibitions for the 50th Anniversary Event of the Nagoya City Science Museum", Journal of the Japan Information-culture Society, Vol.20, No. 1, pp.10-17, May 2013.

[3] S. Aral, "Identifying Social Influence: A Comment on Opinion Leadership and Social Contagion in New Product Diffusion", Marketing Science Articles in Advance, pp. 1-7, 2010.

[4] <moodle.org/> 2015.02.16.

[5] <www.joomla.org> 2015.02.16.

[6] <www.efrontlearning.net/> 2015.02.16.

[7] <www.dokeos.com/> 2015.02.16.

[8] D. Jones and T. Lynch, "A model for the design of web-based systems that supports adoption, appropriation, and evolution", Murugesan, S. and Deshpande, Y. (Eds.): Proceedings of the 1st ICSE Workshop on Web Engineering, 1999, pp.47–56, Los Angeles.

[9] S. Alexander, "e-Learning developments and experiences" (2001) <web.uct.ac.za/org/fawesa/confpaps/alex.pdf> 2014.06.05.

[10] T. Roberts, C. Romm, and D. Jones, "Current practice in web-based delivery of IT courses", APWEB 2000, Xi'an, China, 2000, pp.27–29.

[11] B. Gillani, "Learning Theories and the Design of E-Learning Environments", University Press of America, United States of America, 2003.

[12] S. Krug, "Don't Make Me Think! A Common Sense Approach to Web Usability", 2nd ed., New Riders, Berkeley, California USA, 2006.

[13] <www.mysql.com/> 2015.02.16.

[14] <www.adobe.com/products/flex.html> 2015.02.16.

[15] <php.net/> 2015.02.16.

[16] D. Jones, S. Sharonn, and L. Power, "Patterns: using proven experience to develop online learning", Proceedings of ASCILITE '99, Responding to Diversity, Brisbane: QUT, 1999, pp. 155–162.

[17] M. Vianna, Y. Vianna, I. Adler, B. Lucena, and B. Russo, "Design thinking Business innovation", 1st ed., MJV Press, Rio de Janeiro Brazil, 2012.

[18] L. Grant , K. Facer, M. Owen, and S. Sayers, "Opening Education: Social Software and Learning", Futurelab, United Kingdom, 2006.

[19] V. Paiva, "Letramento digital através de narrativas de aprendizagem de língua inglesa", in English: "Digital literacy through English language learning narratives" CROP., Vol. 12, pp. 1-20, (2007) <www.veramenezes.com/crop.pdf> 2014.07.18.

# Pedestrian Detection with Occlusion Handling

Yawar Rehman[1], Irfan Riaz[2], Fan Xue[3], Jingchun Piao[4], Jameel Ahmed Khan[5] and Hyunchul Shin[6]
Department of Electronics and Communication Engineering,
Hanyang University (ERICA Campus), South Korea
e-mail:{yawar[1], irfancra[2], fanxue[3], jameel[5]}@digital.hanyang.ac.kr, {kcpark1011[4], shin[6]}@hanyang.ac.kr

***Abstract*—Pedestrian detection in a crowded environment under occlusion constraint is a challenging task. We have addressed this task by exploiting the properties of rich feature set which gives almost all cues necessary for recognizing pedestrians. Using rich feature set results in higher dimensional space. We have used partial least square regression to work with more discriminative (lower dimensional) features than (higher dimensional) rich feature set. Part model is further applied to deal with occlusions. Our proposed method gives the accuracy of 98% at $10^{-4}$ false positives per window on INRIA pedestrian database, which is the best result reported so far, under the same false positives per window.**

***Keywords-Pedestrian detection; occlusion handling.***

## I. Introduction

Recent advances in computer vision show researchers interest in developing a system to detect pedestrians efficiently. Detecting pedestrian is a challenging problem and various methods have been proposed. The performance of the detector depends on how well the method works in complex environments such as crowded scenes, illumination variation, occlusion, and cluttering. Extensive literature can be found on the problem of object detection [1]. W. R. Schwartz et al. [2] solved the problem of human detection in reduced dimensional space. The Information in the feature vector was composed of three concatenated features, i.e., Co-occurrence matrices for texture information, Histogram of Oriented Gradients (HOG) for gradient information, and color information. Concatenation of these three features resulted in a feature vector of 170,820 dimensions. Partial Least Square (PLS) regression was used to reduce high dimensional feature space into discriminative reduced dimensional feature space. Quadratic Discriminant Analysis (QDA) model was designed for classification. A. Kembhavi et al. [3] also tackled vehicle detection problem in reduced dimensional space. Color properties of the vehicle and its surroundings were captured by using their method named color probability maps. Gradient information of an object was captured using HOG. Pair of pixels method was used to extract structural properties. Concatenation of all these features resulted in the final feature vector of 70,000 dimensions. PLS regression was used for lower dimensional feature space and QDA model was trained as a classifier for finding objects of interest. Q. Wang et al. [4] handled object tracking as a classification problem and worked it out in reduced dimension by creating different PLS subspaces. A model named adaptive appearance model was proposed which used different subspaces to deal with variation of poses, occlusion, and cluttering problems. M. A. Haj et al. [5] used discriminative properties of PLS lower dimensional space to solve the problem of head pose estimation. Different approaches were compared by [5] and the result obtained from PLS regression was reported the best. X. Wang et al. [6] developed a system which can detect pedestrians and also handles partial occlusion. The final feature vector comprises of gradient and texture features. Instead of using whole feature vector the author used contribution of chunks of feature blocks for the final decision. Linear SVM was used as a classifier and its decision functions were translated in terms of small feature blocks. Bias of the linear SVM decision function was learned from the training database under the constant bias distribution scheme proposed by the authors. The system was evaluated on INRIA pedestrian database and provided the state of the art results. P. Dollàr et al. [8] used boosted features for the task of pedestrian detection. Feature pool was created by using multiple channels such as gradient, intensity, and color features. Several first and second order features were calculated on a patch inside a detection window on different channels. Boosted classifier was trained as [9] on these features in order to classify the detection window while testing.

Our key contribution includes, the heuristic based integration of the part model with the root model. This integration of both models helped significantly in solving the occlusion cases and decreased the number of false negatives and false positives. This decrease tends to improve the efficiency of the system.

We demonstrate our proposed system on INRIA pedestrian database. INRIA pedestrian database was given by [1] when their detector performed almost ideal on the first ever MIT pedestrian database. INRIA dataset is still not fully explored and rigorously used in pedestrian detection evaluation. It contains 2,416 training positive windows cropped from 614 frames and 1,126 testing positive windows cropped from 288 frames. Both windows and frames are included in INRIA database. Training and testing negative frames are provided separately in INRIA database. Our system achieved the accuracy of 91% at $10^{-5}$ false positive per window (FPPW) and of 98% at $10^{-4}$ FPPW. Our system consists of two main models, Partial Least Square (PLS) model and Part model (PM). Partial Least Square is a dimension reduction technique which emphasizes supervised dimension reduction. PLS is helpful in providing discriminative lower dimensional feature space and avoiding the calculations containing thousands of extracted features. Part model ensures the search of a subject (i.e.,

ISBN: 978-1-61208-393-3

pedestrian) in parts rather than to be searched as a whole. PM is helpful in handling occlusions.

## II. Feature Extraction

We have used three types of features in our system, i.e., gradient features, texture features, and color features.

### *A. Gradient Features*

The first and foremost features which we have added in our feature set are gradient features. It is due to the fact that the research in object detection, specifically in human detection has increased significantly after the advent of HOG feature descriptor [1]. HOG was dedicated to human detection and it also provided the best results of its time.

For computing gradient features, we have used heavily optimized implementation of [8][10][11][12] which is similar to that of [1]. An image window is divided into 8x8 pixel blocks and each block is divided into 4 cells of 4x4 pixels. 9 bin HOG features per cell was then calculated obtaining 36 dimensional features per block. Each block is L2 normalized which resulted 4 different normalizations per cell. It is useful because it makes HOG descriptor illumination invariant. HOG also shows rotation invariant properties as long as rotation is within the bin size. Clipping value of histogram bin is set to 0.2 and trilinear interpolation is used for the placement of gradients into their respective bins.

### *B. Texture Features*

The texture information provides better results particularly in case of face detection because of discriminative texture on face (i.e., eyes, nose, mouth, etc). Including texture information in the pedestrian feature set will tend the system towards improvement in terms of detection because of the fact that there is considerable amount of discriminative texture inside human contour.

We have used Local Binary Pattern (LBP) [13] to estimate texture features. LBP is a simple yet efficient technique for calculating texture in an image. It assigns the value ‘1’ in 3x3 pixel neighborhood if each pixel’s intensity value in the neighborhood is greater than or equal to the center pixel’s intensity value, ‘0’ is assigned, otherwise. There are many variants of LBP but we have used the most stable one which was reported to achieve good results by many authors. 3x3 neighborhood produces 256 possible binary patterns which are too many for making reliable texture feature descriptor but in 256 possible binary patterns there exist total of 58 patterns, which exhibit at most two bit-wise transitions from ‘0’ to ‘1’ or from ‘1’ to ‘0’. These patterns are known as uniform patterns. Using uniform patterns instead of 256 patterns will remarkably reduce the texture feature vector size with marginal decrease in performance [13]. We have used the implementation of uniform patterns as was given by [14]. An image window is divided into the blocks of 8x8 pixels and for each block a 58 texture feature descriptor is calculated. The final texture feature set is obtained by concatenating features obtained from several blocks.

### *C. Color Features*

Color features play an important role in providing discriminative identities to objects. The dilemma is, when talking about pedestrian detection, better recognition rates and efficiency by including color information is doubted by some researchers because of the variability in clothing color. Instead [2][8][15] showed the importance of color features in pedestrian detection.

We have taken the samples of pedestrians and non-pedestrians (i.e., non-humans) from INRIA database and converted into LUV color space. Our intuition of selecting LUV came from the result reported by [8], that LUV outperformed other color spaces by achieving an accuracy of 55.8% alone (i.e., not combined with other features) in pedestrian detection. PLS regression is applied on L, U, and V space separately. PLS regression components shows maximum inter-class and intra-class variance. Human contour can be seen as silhouette by plotting them. U space showed dominant (red) peak at head region in all three PLS components. It is because variance of the head region in an image with respect to the surrounding region was maximum. During experimentation, we tried to include only U space as color information, but accuracy has decreased. In our opinion, the decrease in accuracy was due to lack of color information which also points to the fact that including color information plays a significant role in detection. We have exploited this by including LUV color space representation in our system.

The final feature vector reflecting different extracted information from an image window looks like:

$$F = [Gradient\ Texture\ Color] \tag{1}$$

## III. Partial Least Squares Model

We have accumulated rich feature set for all possible cues of pedestrians, which resulted in high dimensional feature space. In our experiments, the number of samples used for training the classifier are less than the dimension of rich feature space. The phenomenon when data dimensions remains greater than the number of samples is known as multicollinearity. Partial least squares regression addresses the problem of multicollinearity and reduces data dimensions. PLS regression uses class labels for producing latent components which makes lower dimensional space more discriminative. An idea of constructing latent variables is summarized here, for details reader is encouraged to refer [16][17].

There are two popular variants of PLS, Non-iterative partial least square (NIPALS) and Simple partial least square (SIMPLS). They both differ in matrix deflation process. We used SIMPLS regression in our experiments. Let $X^{N \times m}$ and $Y^{N \times n}$ be the two blocks of variables. PLS models the relationship between the sets of variables by maximizing the covariance between them through latent variables.

$$X = TP^T + E \tag{2}$$

$$Y = UQ^T + F \tag{3}$$

Where, $T^{Nxp}$ and $U^{Nxp}$ are score matrices; $P^{mxp}$ and $Q^{nxp}$ are loading matrices and $E^{Nxm}$ and $F^{Nxn}$ are residuals. The weight matrix in first iteration is calculated as,

$$w_1 = \bar{X}^T\bar{Y}/\|\bar{X}^T\bar{Y}\| \tag{4}$$

and till $k^{th}$ iteration it is calculated as,

$$\bar{X}_k = \bar{X}_{k-1} - t_{k-1}p_{k-1}^T \tag{5}$$

Where, $t$ and $p$ are the column vectors of matrix $T^{Nxp}$ and $P^{mxp}$, respectively, and $k$ represents the number of PLS factors. The dimension of an input image $x$ is reduced by projecting its feature vector on to the weight matrix obtained after $k$ iterations, where columns of $W=\{w_1,w_2,w_3,\ldots w_k\}$ represents PLS components. After projection, a low dimensional vector $z^{1xk}$ is obtained.

Principal component analysis (PCA) is a well-known technique for dimension reduction. It also addresses multicollinearity problem, but doesn't consider class labels of data for dimension reduction. PLS is a supervised dimension reduction technique which considers class labels for dimension reduction. This enables PLS to produce highly discriminative reduced dimensional data as it is evident from Figures 2 and 3. We have plotted first two components of both dimension reduction techniques to show their discriminative power in lower dimensional space.

Our system extracts three cues from an image patch which makes our high dimensional feature set. The total number of features extracted from an image patch are approximately fourteen thousand. With the help of PLS, we have reduced our feature set to only sixteen dimensions which are the best representation of our high dimensional data. Figure 1 shows the mean classification error at different dimensions. The performance of our system at sixteen lower dimensional features can be observed in Figure 5.

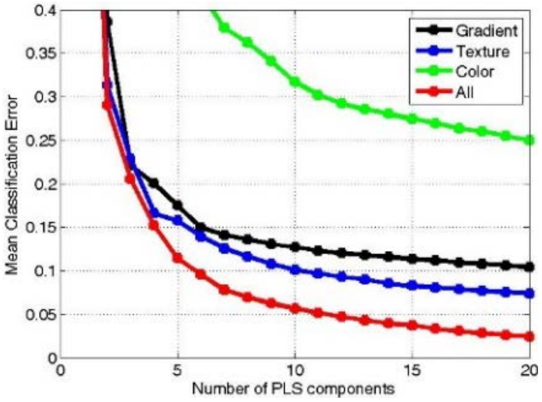


Figure 1. Mean square error vs PLS components

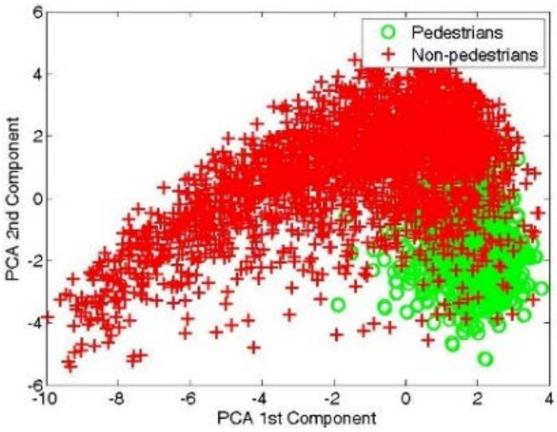


Figure 2. PCA lower dimensional space

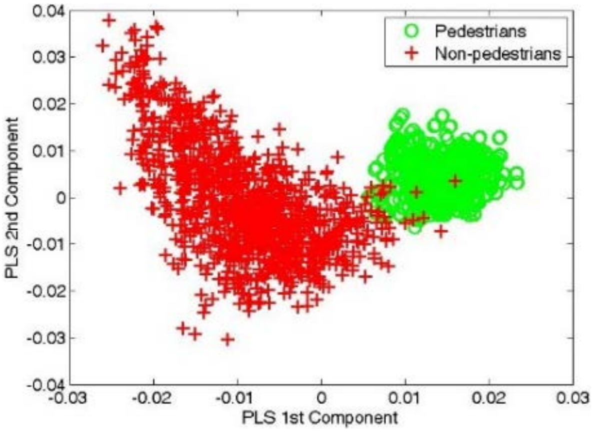


Figure 3. PLS lower dimensional space

Our PLS model gives the accuracy of 94% at $10^{-5}$ FPPW and accuracy of 96% at $10^{-4}$ FPPW on INRIA pedestrian database.

## IV. Part Model

Part models are generally used in pedestrian detection to handle occlusions. It is a common practice to divide human body into five parts (i.e., head, left torso, right torso, upper limbs, and lower limbs) and detect each of the part separately. Deformation schemes were also introduced by several authors in order to keep different parts glued together. In our case, we have used upper body part model. The model includes head, left torso, and right torso. We argue that, using upper body parts as a whole will give more discrimination among features because hardly any other object is represented with this structure. The structure of head, shoulders, arms, and torso (all connected) gives more discriminative feature property rather than to search them individually. Furthermore, to avoid complex deformation schemes [7], using only upper body as a part model is the best choice.

The upper body part model is designed using gradient, texture, and color features. Adding color features tends to improve the performance of detector, because of similar variance of color in face and hand regions. Final feature vector contains the information of gradients, texture, and color from head, shoulders, left, and right torso. The performance of our part model on INRIA pedestrian database is shown in Figure 6.

## V. PLS + PART MODEL (COMBINED MODEL)

Our approach for combining both models is based on simple heuristic. We have trained our classifier for PLS model on lower dimensional space which is very discriminative in nature. Linear SVM trained on lower dimensional data classifies efficiently and separates humans from non-humans almost accurately. Upon careful analysis, we came to know that the samples which were incorrectly classified by linear SVM either positives or negatives, their score lie in the vicinity of ‘0’. We generate our occlusion hypothesis that if a sample *‘q’* whose predicted score value *‘v’* lie between *th1* and *th2*, then it is considered to be an occlusion and upon meeting this condition our part model will be activated and final score *‘m’* returned by part model will be taken as true value of the sample *‘q’*. The heuristic for occlusion hypothesis is shown in Figure 4.

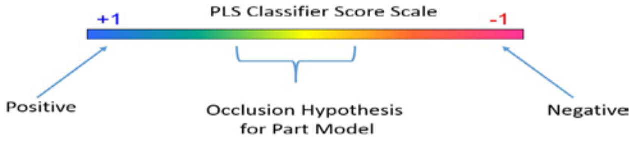


Figure 4. Heuristic for occlusion hypothesis

Figure 7 shows the performance of all our models on INRIA pedestrian database.

## VI. EXPERIMENTAL RESULTS

The comparison between HOG, variant of HOG (FHOG) introduced by [7], and our method is shown in Figure 8. Each of the classifier was trained as described in Section 3. Our system gives accuracy of 91% at $10^{-5}$ false positive per window (FPPW) and accuracy of 98% at $10^{-4}$ FPPW. Testing was done on 1,126 positive cropped windows and 105,500 negative cropped windows from negative images provided by INRIA dataset. According to the observations of [7], there are some cases in which the use of light insensitive features will give benefit and in other cases the use of light sensitive features will give benefit. FHOG consists of 32 features. 13 of them are the representations of 36 HOG features in reduced dimensional space which are light insensitive features and remaining features are light sensitive features. As we can see in Figure 8 that FHOG clearly dominates HOG. On the other hand, our method achieved the best accuracy in comparison to HOG and FHOG. To our knowledge, our system gives the best state of the art results at $10^{-4}$ FPPW on INRIA pedestrian database.

In our opinion, the reason for achieving the best results on INRIA dataset in FPPW evaluation metrics is that, our system was able to solve occluded cases with high confidence values, which in case of other state of the art detectors either produced false negatives or corrected that case with a lower confidence value. The time cost of projecting high dimensional feature vector onto the weight matrix and lacking of vertical occlusion handling can be counted as the limitations of the proposed system.

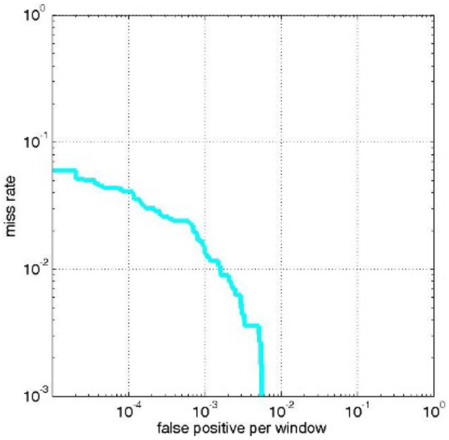


Figure 5. Performance of PLS model

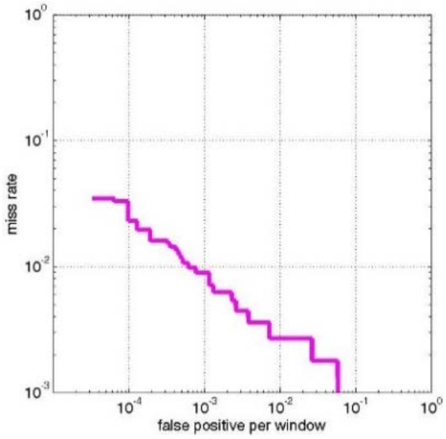


Figure 6. Performance of Part model

Table 1 translates Figure 8 by indicating maximum accuracy achieved by HOG, FHOG and the proposed method. Figure 9 shows some sample results of our proposed system.

## VII. CONCLUSION

We have developed a system which is capable of detecting human via monocular camera images efficiently. With the help of PLS, we are able to represent our rich feature set in more discriminative lower dimensional space. Part model is also integrated with our system for handling occlusions. We have achieved the detection rate of 98.1% at $10^{-4}$ FPPW which is the best result reported on INRIA pedestrian database.

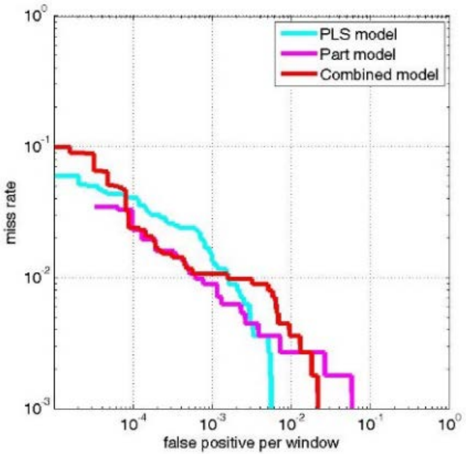


Figure 7. Performance of all models

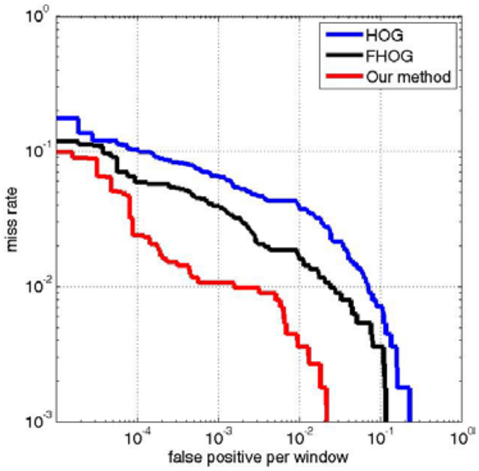


Figure 8. Comparison of our model with HOG and FHOG

TABLE I. (1-MISSRATE) AT FPPW VALUES

| *1-missrate* | *$10^{-5}$ FPPW* | *$10^{-4}$ FPPW* |
|---|---|---|
| HOG | 82.5 % | 90.0 % |
| FHOG | 88.2 % | 94.1 % |
| Ours | 90.5 % | 98.1 % |

We plan to further improve this detection rate by effectively adding another dimension of tracking and between-frames information into our system.

## ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MOE) (No. NRF-2013R1A1A2004421).

## REFERENCES

[1] Dalal, N., Triggs, B., "Histograms of oriented gradients for human detection," Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, June 2005, pp.886-893 vol. 1, 25-25.

[2] Schwartz, W.R.; Kembhavi, A.; Harwood, D.; Davis, L.S., "Human detection using partial least squares analysis," Computer Vision, 2009 IEEE 12th International Conference on, Sept. 29 2009-Oct. 2 2009, vol., no., pp.24-31.

[3] Kembhavi, A.; Harwood, D.; Davis, L.S., "Vehicle Detection Using Partial Least Squares," Pattern Analysis and Machine Intelligence, IEEE Transactions on , June 2011 vol.33, no.6, pp.1250-1265.

[4] Qing Wang; Feng Chen; Wenli Xu; Ming-Hsuan Yang, "Object Tracking via Partial Least Squares Analysis," Image Processing, IEEE Transactions on , Oct. 2012, vol.21, no.10, pp.4454-4465.

[5] Haj, M.A.; Gonzalez, J.; Davis, L.S., "On partial least squares in head pose estimation: How to simultaneously deal with misalignment," Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, June 2012, vol., no., pp.2602-2609, 16-21.

[6] Xiaoyu Wang; Han, T.X.; Shuicheng Yan, "An HOG-LBP human detector with partial occlusion handling," Computer Vision, 2009 IEEE 12th International Conference on , vol., no., pp.32-39.

[7] Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D., "Object Detection with Discriminatively Trained Part-Based Models," Pattern Analysis and Machine Intelligence, Sept. 2010, IEEE Transactions on , vol.32, no.9, pp.1627-1645.

[8] P. Dolla´r, Z. Tu, P. Perona, and S. Belongie, “Integral Channel Features,”, Proc. British Machine Vision Conf. 2009.

[9] P. Viola and M. J. Jones, “Robut Real-Time Face Detection”, International Journal of Computer Vision, 2004, Vol. 57, No. 2.

[10] P. Dollár, S. Belongie and P. Perona, “Fastest Pedestrian Detector in the West”, Proceedings of BMVC, 2010.

[11] P. Dollár, R. Appel and W. Kienzle, “Crosstalk Cascades for Frame-Rate Pedestrian Detection”, Proceedings of ECCV, 2012.

[12] Dollar, P.; Appel, R.; Belongie, S.; Perona, P., "Fast Feature Pyramids for Object Detection," Pattern Analysis and Machine Intelligence, IEEE Transactions on, Aug. 2014 vol.36, no.8, pp.1532-1545.

[13] Ojala, T.; Pietikainen, M.; Maenpaa, T., "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," Pattern Analysis and Machine Intelligence, IEEE Transactions on, Jul 2002, vol.24, no.7, pp.971-987.

[14] A. Vedaldi and B. Fulkerson, “VLFeat: An Open and Portable Library of Computer Vision Algorithms”, [Accessed from http://www.vlfeat.org], 2008.

[15] Walk, S.; Majer, N.; Schindler, K.; Schiele, B., "New features and insights for pedestrian detection," Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, vol., no., pp.1030-1037, 13-18.

[16] R. Rosipal and N. Kramer, “Overview and recent advances in partial least squares in Latent Structures Feature Selection”, Springer Verlag, 2006.

[17] S. Wold, “PLS for Multivariate Linear Modeling QSAR”, Chemometric Methods in Molecular Design, 1994.

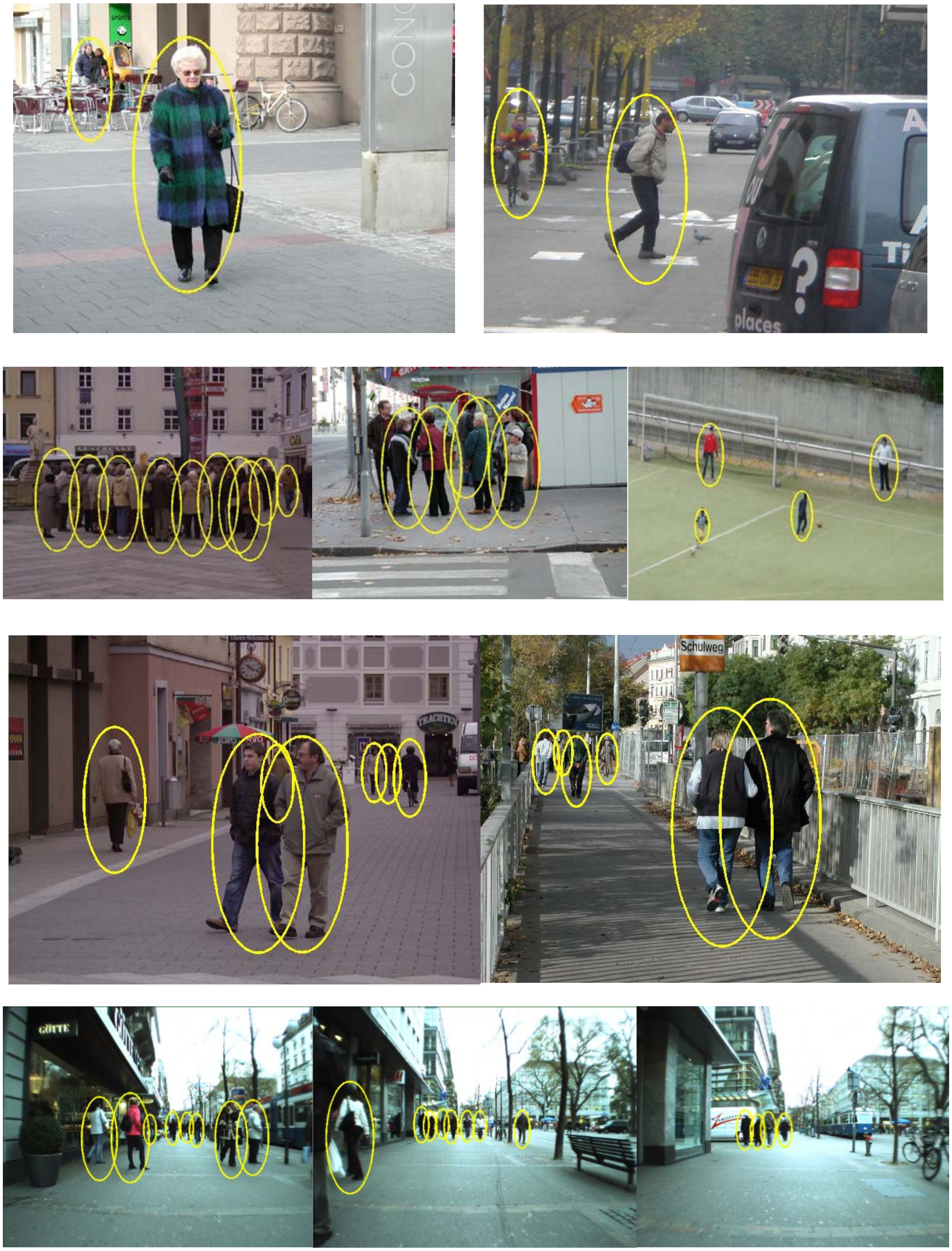


Figure 9. First three rows shows the results obtained from INRIA database and fourth row shows some results from ETH pedestrian database. The performance of our detector in occlusions, cluttered scenes, and pose variations, should be noted.

# A Statistical Approach for Discovering Critical Malicious Patterns in Malware Families

Vida Ghanaei, Costas S. Iliopoulos, and Richard E. Overill

Department of Informatics, King's College London
Email: {vida.ghanaei, c.iliopoulos, richard.overill}@kcl.ac.uk

***Abstract*—In this paper, we present carefully selected critical malicious patterns, which are in common among malware variants in the same malware family, but not other malware families, using statistical information processing. The analysed critical malicious patterns can be an effective training dataset, towards classification of known and unknown malware variants. We present malware variants as a set of hashes, which represent the constituent basic blocks of the malware Control Flow Graph, and classify them into their corresponding malware family. By computing the Distribution Frequency for each basic block residing in all the malware families, the importance of being a possible representative to become a critical malicious pattern for a specific malware family is measured. This value is carefully computed by considering the population of each malware family.**

***Keywords*–Malware; Malicious Patterns; Malicious Shared Code; Classification; Control Flow Graph; Numerical Statistics.**

## I. Introduction

Malware is considered a major computer security problem as it can attach to other computer programs and infect them. Infection is defined as unwanted modification to other program to include a possibly evolved, version of itself [1]. Based on McAfee threats report [2], more than 30 million new malware variants, and over 250 million in total, are recorded in the first quarter of year 2014. Malware spread itself by most common digital methods such as e-mail and instant message applications, and through social engineering techniques. Other means of malware spread methods are World Wide Web (WWW), network-shared file systems, peer to peer file sharing networks, removable media, Internet Relay Chat (IRC), Blootooth or wireless local area networks [3].

In malware research, data collection is not an issue any more. It is easily achievable by setting up configured honeypot servers in the laboratory environment. But, the flow of malicious software variants reaching our networks and computers is so enormous that makes it impossible to process them exclusively. Therefore, it is essential, to identify malicious patterns which appear in malware variants, and to obtain structural understanding of malicious patterns, in order to analyse, classify, and detect malware. However, collection of huge amount of malware variants, which have embodied different obfuscation techniques, and have mutated in various polymorphic and metamorphic forms, demands automated techniques to identify, and present the malicious patterns. While malware variants belonging to the same malware family[1] share a certain amount of malicious code, identification of critical shared code provides knowledge towards classification and detection of unknown malware variants.

In this paper, we present an effective statistical approach to identify, and to render critical malicious patterns in malware families, which are essential elements towards automated classification of known and unknown malware in large amount. We rely on the shared code among different malware variants, which potentially occur in one specific malware family, with considering its possibility of occurring in other malware families, to identify the most critical malicious patterns in every malware family. In this paper, the shared code is studied at basic block level of the control flow related code, and we are able to present the most critical malicious patterns for every malware family. By critical malicious pattern, we mean the most frequent basic blocks, which are present at most in one specific malware family, and comparatively less in other malware families. Also, the shared code among different malware families are not interesting for classification purpose, as certain functions are in-common among all the malicious software variants, and even can be in-common with non-malicious software.

We introduce a novel formalisation methodology which automates the identification of critical malicious patterns for each malware family. It is defined as a statistical approach, which computes the Frequency Distribution Ratio, for each constituent basic block of a malware variant within each malware family. This value is penalised statistically for occurring in other malware families. To our knowledge, our approach in compare to related works, is more consistent as we encounter the distribution frequency of certain basic blocks in each malware family, as well as in between different malware families, to identify critical malicious patterns. Not considering the ratio of the frequency of each basic block in its associated malware family, results in inaccurate identification of critical malicious patterns, which is discussed in details, in Section VIII.

In Section II, the background and related work on different malicious pattern matching techniques are reviewed. In Section III, the notations and definition are given. In Section IV, details of the dataset used in experiments, are presented. In Section V, the formalisation of our approach is defined. In Section VI, an overview on the shared code concept in malware, and how we bypass the obfuscations applied to shared code, is explained. In Section VII, the methodology and implementation process is described. In Section VIII, experimental results of our approach is discussed and compared to related works. Finally in Section IX, our methodology, its results, and the future work are concluded.

## II. Background and Related Work

Different features of malicious software, and pattern matching techniques are studied to classify enormous amount of

[1] Defined in Section III.

malware variants getting submitted to honeypots. The n-gram, defined in section III., computation as a pattern matching techniques, and its application in malware analysis, was first introduced by G. J. Tesauro et al [4], to identify boot sector viruses automatically by applying artificial neural networks. J. Z. Kolter et al[5], computed Information Gain (IG) for each n-gram to select the most relevant n-grams, to heuristically identify and classify malware. They selected top 500 n-grams, and applied different machine learning algorithms to detect and classify malware variants and identified boosted J48 algorithm produces the best detector. Although they showed good detection with areas under the ROC curve around 0.9, they did not consider metamorphic obfuscations and polymorphism.

S. Cesare et al [6], presented a static approach to detect polymorphic malware based on control flow graph[1] classification. They unpacked[1] the polymorphic malware using application level emulation, disassembled the unpacked samples, translated the disassembly into intermediate language, reconstructed and normalised the control flow graph for each procedure based on the intermediate language. Subgraphs of size *K*, and n-grams extracted from the strings representing the graphs, are the two features used to pre-filter potential malware, and used to construct a feature vector. Distance metrics are used to measure similarity between two feature vectors. The presented results shows considerable collisions, and false positives using subgraphs of size *K* compare to using n-grams vector features.

BinDiff[2], which is an add-on plug-in for IDAPro[3], relies on heuristics to generate matches between two malware variants. It generates a signature for every function of the malware executable based on its abstract structure, ignoring the assembly code generated by IDAPro. The signature generated depends on the structure of the normalised flow graph of the function, and consists of number of basic blocks, number of edges, and number of calls to sub-functions. Two functions match, if a signature occurs in both only once. If a match identified, all the calls-to relations between the matched functions is checked for possible matches of all the subset functions, until no further matches found. BinDiff is a pairwise matching tool and generates lots of false matches as it relies on further matches on big portion of the code, rather than the most critical code.

C. Miles et al [7], presented a recent artefact, VirusBattle, which is a malware analysis web-service. VirusBattle reason about malware variants in different levels of abstraction including the code, the statically analysed shared semantics, referred as juice [8], among different variants, and sequence of events a malware takes during execution time to map the similarities and interrelationships. Juice, transforms code semantics computed over an x86 disassembly, by generalising the register names, literal constants, and computing the algebraic constrains between the numerical variables. Therefore, semantically similar code fragments can be identified by comparing their hash values. VirusBattle provides automated PE unpacking web-service as well, which is a generic unpacker[4], and publicly available web-service. In this paper, we use the unpacker provided by the VirusBattle SDK to unpack the malicious samples, as well as the juice which is the generalised presentation of the semantics to avoid code obfuscations.

## III. Definitions and Notations

*a) Malware Taxonomy:* Malware is a general term for malicious software, which refers to virus, worm, root-kit, trojan-horse, dialer, spyware, and key-logger. It is defined based on its mean of distribution, and its dependency on the host to infect.

*b) Malware Family:* Malware variants which are the result of the mutation of each other, and share considerable amount of critical code with one another, are considered to belong to the same malware family. The already known malware variants belonging to the same malware family, can be classified by signature based anti-virus scanners. Variants in the same malware family are meant to show similar behaviour, to target similar files, and to spread the same way. Therefore, each malware family, denoted $f$, is a set of malware variants.

*c) Packed Malware:* Malware which contains encryption routine, compression, or both, is referred to as packed malware. Unpacked malware is the malicious code, without encryption or compression.

*d) Control Flow Graph:* Control Flow Graph (CFG), is a directed graph, is denoted *G = (V, E)*, if $u, v \in V$ be the nodes of the graph, a possible flow of control, from *u* to *v* is represented by $e \in E : u \rightarrow v$. In a CFG, every node is a representation of a basic block, denoted $v$, and the edge is a path between these nodes. A basic block is defined as a sequence of instructions without any jumps or jump targets in between the instructions [9].

A basic block always runs before any other instructions in later positions of the CFG, which means no other instruction runs between two instructions in the same sequence. The directed edge between two basic blocks expresses the jump command in the control flow, which is caused by Control Transfer Instructions (CTI) such as call, conditional jump, and unconditional jump instructions [10].

*e) N-gram Frequency Distribution:* N-gram is a contiguous sequence of *n* items from a given sequence such as assembly statement raw bytes, opcodes and etc. N-gram frequency distribution, is a well-known approach for extracting features from malicious software to develop training dataset for classification purpose [5]. In this article, the hash value computed for every basic block is treated as n-gram sequence.

## IV. Dataset

Our dataset consists of 777 distinguished malware variants, which are spread over 23 different malware families, as shown in Table I., and it contains total of $1,116,798$ basic blocks. Malware families are structured disjoint, to avoid inaccurate computation of critical malicious patterns. If the same malware be a member of different malware families, its constituent basic blocks will be counted towards all the involved malware families and cause false matches. Also, in the classification process of malware variants, a new sample is to be placed in one malware family based on the similarity measurement of its shared code with that malware family as oppose to other malware families. However, each malware family is treated as a multiset of basic blocks. In other words, each basic block

[2] Zynamics and Google, BinDiff. Available: http://www.zynamics.com/bindiff.html

[3] I. Guilfanov, IDAPro, An Advanced Interactive Multi-processor Disassembler, Data Rescue. Available: http://www.datarescue.com

[4] V. Notani and A. Lakhotia, VirusBattle SDK-Unpacker. Available: https://bitbucket.org/srl/virusbattle-sdk/wiki/Home

can occur multiple times in the same malware family, and in other malware families as well. Each basic block is indexed to its associated malware variant to produce more informative outcome.

Malware samples are collected from the VirusSign[5] free on-line service, and the dataset presented in a recent study [11]. All the malware samples existing in the database, are unpacked by the VirusBattle SDK unpacking service[6].

TABLE I. MALWARE FAMILIES EXISTING IN DATASET

| Malware Family | No of Variants | No of Basic Blocks |
|---|---|---|
| Agent | 17 | 12316 |
| Agobot | 17 | 101782 |
| ATRAPS | 9 | 16064 |
| Bancos | 13 | 33831 |
| Cosmu | 8 | 1133 |
| Crypt | 63 | 239821 |
| Dldr | 12 | 82929 |
| Downloader | 9 | 5769 |
| Dropper | 63 | 75082 |
| Fareit | 146 | 64819 |
| Gobot | 14 | 28451 |
| IRCbot | 17 | 11405 |
| Klez | 22 | 69721 |
| Kryptik | 97 | 49696 |
| MyDoom | 40 | 67169 |
| Poebot | 18 | 9285 |
| Sality | 54 | 26379 |
| Spy | 26 | 38511 |
| Symmi | 18 | 23600 |
| ULPM | 65 | 94165 |
| Unruy | 18 | 2407 |
| Vanbot | 16 | 1713 |
| Virut | 27 | 15750 |

## V. FORMALISATION

Considering each malware family as a set of malware variants, and each malware variant be a set of multiple basic blocks. The malware families are disjoint, in other words each malware variant can only be a member of one malware family. However, $v_i$, is the $i$th distinct member of a $v$, which can occur in any malware family and multiple times.

Let $f_k$ be the $k$th distinct member of $f$, and $\tau_{i,k}$ be the number of occurrences of $v_i$ in $f_k$, and $\tau_k = \sum_i \tau_{i,k}$. Therefore, Term Frequency Ratio ($TFR$) for $v_i$ occurring in $f_k$ is defined as shown in Equation 1.

$$TFR_{i,k} = \frac{\tau_{i,k}}{\tau_k} \tag{1}$$

$TFR$ is computed for all the $v_i$, which exist in every malware family, individually. $TFR_{i,k}$ indicates the ratio of how frequently $v_i$ has occurred in $f_k$. Every $v$ is indexed to its associated malware variant. The frequency ratio of each $v_i$ is considered, rather than the frequency of each $v_i$, as the number of malware variants in each malware family is varying and consequently the $\tau_k$. Therefore, it is essential to encounter the the population of malware families in compare to each other, to obtain the correct frequency for each $v_i$.

Here, we compute the importance of each $v_i$, as a possible representative for the critical malicious pattern of each malware family. In order to do so, its TFR value is penalised by subtracting a quantity $\alpha_{i,k}$. $\alpha_{i,k}$ is the sum of $TFR_{i,j}$ of all the occurrences of $v_i$ in all the malware families, $f_j$, where $j \neq k$; except $f_k$. It is computed for every malware family in relation to its population. Therefore $\alpha_{i,k}$ is defined as shown in Equation 2.

$$\alpha_{i,k} = \sum_{j \neq k} TFR_{i,j} \tag{2}$$

Therefore, Term Frequency Distribution ($TFD$) for $v_i$, computes the frequency ratio of $v_i$ in malware family $f_k$, in relation to its distribution in the other malware families, which is $\alpha_{i,k}$, and it is defined as shown in Equation 3.

$$TFD_{i,k} = TFR_{i,k} - \alpha_{i,k} \tag{3}$$

$TFD_{i,k}$ indicates how critical $v_i$ is as a malicious pattern for $f_k$, as $v_i$ is penalised for every time it occurs in $f_j$. The penalising as described, is computed by subtracting the sum of all the corresponding $TFR_{i,k}$ of $v_i$ occurrences in $f_j$. This sum, $\alpha_{i,k}$, indicates how common $v_i$ is in other malware families, $f_j$, rather than $f_k$. Accordingly, $TFD_{i,k}$ shows how specific $v_i$, is to malware family $f_k$, rather than other malware families, $f_j$.

## VI. SHARED BASIC BLOCKS

Malware variants belonging to the same malware family, share certain amount of code which relates them. The syntax of the shared code in every variant varies every time malware mutates, which is due to applied obfuscation techniques to avoid detection by anti-virus scanners. Different syntax presentation for the same semantics, causes many mismatches. To overcome incorrect matches that result from these syntax changes, we use the VirusBattle SDK on-line platform to generate the same syntax for blocks of code, which carry the same semantics. Therefore, malware samples are unpacked using VirusBattle SDK unpacker, and the juice, which is the generalised presentation of the semantics, is generated for every sample using the on-line service provided by the same platform. Here, semantics refers to the semantics of instructions in the basic block, and is developed based on the original disassembled code after unpacking. We use this service as it has shown good results and serves our needs well.

## VII. METHODOLOGY AND IMPLEMENTATION

According to previous studies by C. Miles et al. [7], and Zynamics[2], the CFG has shown the best results for malware similarity measurement purposes. Therefore, we chose to identify shared code among different malware variants by looking into their CFG, and at the basic block level. Basic blocks are preferred to instructions, as instructions become meaningless without contextually relating to other instructions, semantic-wise, and functions can contain many basic blocks and produce incorrect matches, such as the work presented in Zynamics. For every malware sample, the juice corresponding to each

[5] VirusSign. Available: https://www.virustotal.com

[6] V. Notani and A. Lakhotia, VirusBattle SDK-Unpacker, 2014. Available: https://bitbucket.org/srl/virusbattle-sdk/wiki/Home

of its basic blocks is hashed and stored in a text file. Thus, each malware is presented as a multiset of hash values, which resembles its constituent basic blocks. Hashing is applied to accelerate the matching process.

Malware samples are initially scanned, and labeled by Avira[7] anti-virus scanner, and through the Open Malware[8] on-line service, to be pre-classified into their related malware families, and stored accordingly. Each malware sample can be a member of one malware family only, and a warning message is flagged if duplications found. Duplication of malware variants is avoided by computing their hash value. Therefore, each malware family consists of set of malware variants in which, each malware variant is a multiset of $v_i$. Duplicate malware variants are restricted, as the aim of this paper is to identify critical malicious patterns, which can be used as the training dataset for malware classification purpose, and allowing multiple copies of the same malware variant, results in wrong frequency ratio of the shared basic blocks.

The critical malicious patterns for each malware family is described as a list of basic blocks, which occur in malware family $f_k$ the most, and are least likely to occur in other malware families, and computed by TFD for each $v_i \in f_k$. In other words, the importance of basic block $v_i$ is lowered by its occurrence in different malware families, as it is not implicit to a specific malware family, therefore not the best candidate for being categorised as the critical malicious pattern. However, due to the fact that the amount of shared code between different malware families is considerably high, not many distinct basic blocks to one malware family, are ranked as its critical malicious pattern. Nonetheless, our formalisation is defined by considering these fundamentals, by studying the distribution frequency of each basic block in each malware family, as well as in between all the malware families. Hence, the inclusion of basic blocks, which carry less specific functionality to a malware family, as its critical malicious pattern, is avoided. Therefore, the main fundamentals in defining the formalisation encountered are expressed as, each malware family consists of different number of malware variants, as well as each malware variant consists of different number of basic blocks depending on the amount of code involved in its CFG. Also, basic block $v_i$ may occur in different malware families. Thus, without considering the ratio of the frequency of each basic block, rather than its frequency in each malware family, and its distribution frequency in between other malware families, it is impossible to identify the correct critical malicious patterns for each malware family. According to our testing and experiments, these criteria are necessary to be encountered to define an effective formalisation in order to identify Critical malicious patterns for each malware family.

## VIII. Experimental Results and Discussion

The developed command line interface provides the ability to add or delete a malware family, to list already existing malware families, and to compute and update the TFD value for every basic block stored in the database. Also, by making different queries, it is possible to obtain the TFD for each basic block in its associated malware, in its malware family, and in compare to other malware families. It is possible to query a single basic block, and observe in which of the malware families it has occurred, and obtain its TFD value corresponding to each of the malware families. As shown in Listing 1., query for one basic block is made, which shows it is occurred in three malware families, Fareit family, Dropper Family, and the Klez Family. It also displays the number of times it has occurred in each of the malware families, and the total number of malware variants, which is 22. However, in Listing 1. we only show the first two malware variants, due to limited space.

The same way, we can query for one specific malware, and list all of its basic blocks. For each basic block, it shows list of the malware families in which the basic block occurs, the count of times it occurs in each malware family, and the count of malware variants which contain that basic block in one malware family. These queries are developed to understand the distribution of each basic block in its associated malware family, as well as the whole database. Observing these information provides an effective understanding on the shared code among different malware families, which is essential for the classification purpose.

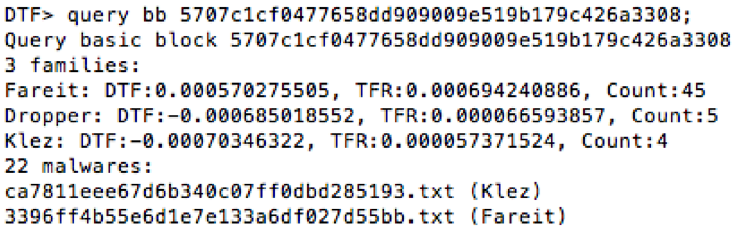
The most critical malicious pattern for each malware family, is supposed to be the constituent basic blocks of that malware family with the highest TFD values. As shown in Listing 2., the top 10 patterns for the Sality family is queried. Sality family is chosen randomly as a sample, and all of the malware families, as shown in Table I., are included in the database, and can be queried. The result displays, the basic blocks based on their TFD value in descending order. Therefore, the result of this query presents a sorted list, in which for each basic block, total number of its occurrences, its TFR value, and its TFD value are retrieved. Furthermore, another query lists the occurrences of $v_i$, in the asked malware family, as well as all the other malware families, $f_j$. This query provides a good understanding on the shared basic blocks between $f_k$, and the other malware families, $f_j$. These shared basic blocks are considered to be strong candidates for the critical malicious pattern of the malware family, as they carry high TFD value. Both of the queries can be made to display any number of the basic blocks, in descending order based on the TFD value, as long as the value is equal or less than $\sum_i v_i$.

The details of the occurrences of $v_i$, in other malware families, $f_j$, as shown in Listing 3. for the first 5 basic blocks of the Sality family, with the highest TFD value, is given. As listed, the 3rd basic block has occurred in other malware families as well, which reveal the importance of the same basic block in other malware families. This basic block has occurred 344 times in Sality family, and 186 times in 11 other malware families, $f_j$. However, as explained before the frequency distribution ratio of a basic block, has the main impact on how critical it can be for a malware family, in terms of critical pattern identification. Further more, the hash value of the basic blocks, can be traced back into its juice, and the actual code for further analysis, as shown in Table II. In this paper, retrieving the actual code associated to each of the hash values, is traced manually as our aim is to identify the malicious patterns. However, it is a straight forward task to automate this process, as the name of the malware families, malware names, and the basic blocks identifier are the computed hash value, for each.

As mentioned in Section VII., different fundamentals regarding the shared code impacts the formalisation of malicious

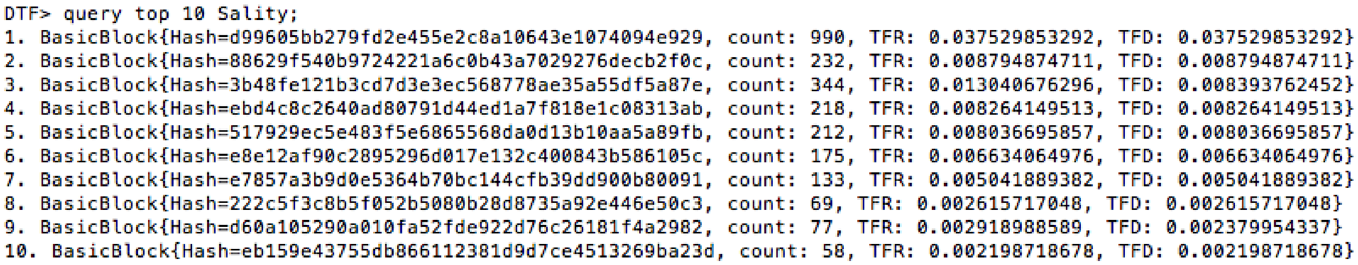
[7] A.O.G. and Co., Avira Antivirus. Available: http://www.avira.com

[8] G.T.I.S. Center, Open Malware, Offensive Computing.



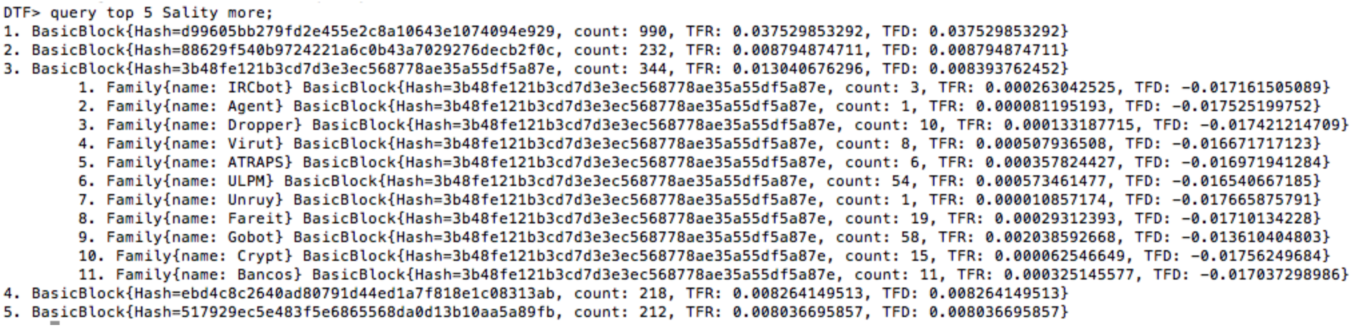
```
DTF> query bb 5707c1cf0477658dd909009e519b179c426a3308;
Query basic block 5707c1cf0477658dd909009e519b179c426a3308
3 families:
Fareit: DTF:0.000570275505, TFR:0.000694240886, Count:45
Dropper: DTF:-0.000685018552, TFR:0.000066593857, Count:5
Klez: DTF:-0.00070346322, TFR:0.000057371524, Count:4
22 malwares:
ca7811eee67d6b340c07ff0dbd285193.txt (Klez)
3396ff4b55e6d1e7e133a6df027d55bb.txt (Fareit)
```

Listing 1. QUERY FOR ONE SPECIFIC BASIC BLOCK IN ALL THE DATABASE



```
DTF> query top 10 Sality;
1. BasicBlock{Hash=d99605bb279fd2e455e2c8a10643e1074094e929, count: 990, TFR: 0.037529853292, TFD: 0.037529853292}
2. BasicBlock{Hash=88629f540b9724221a6c0b43a7029276decb2f0c, count: 232, TFR: 0.008794874711, TFD: 0.008794874711}
3. BasicBlock{Hash=3b48fe121b3cd7d3e3ec568778ae35a55df5a87e, count: 344, TFR: 0.013040676296, TFD: 0.008393762452}
4. BasicBlock{Hash=ebd4c8c2640ad80791d44ed1a7f818e1c08313ab, count: 218, TFR: 0.008264149513, TFD: 0.008264149513}
5. BasicBlock{Hash=517929ec5e483f5e6865568da0d13b10aa5a89fb, count: 212, TFR: 0.008036695857, TFD: 0.008036695857}
6. BasicBlock{Hash=e8e12af90c2895296d017e132c400843b586105c, count: 175, TFR: 0.006634064976, TFD: 0.006634064976}
7. BasicBlock{Hash=e7857a3b9d0e5364b70bc144cfb39dd900b80091, count: 133, TFR: 0.005041889382, TFD: 0.005041889382}
8. BasicBlock{Hash=222c5f3c8b5f052b5080b28d8735a92e446e50c3, count: 69, TFR: 0.002615717048, TFD: 0.002615717048}
9. BasicBlock{Hash=d60a105290a010fa52fde922d76c26181f4a2982, count: 77, TFR: 0.002918988589, TFD: 0.002379954337}
10. BasicBlock{Hash=eb159e43755db866112381d9d7ce4513269ba23d, count: 58, TFR: 0.002198718678, TFD: 0.002198718678}
```

Listing 2. SALITY FAMILY, 10 BASIC BLOCKS WITH THE HIGHEST TFD VALUE



```
DTF> query top 5 Sality more;
1. BasicBlock{Hash=d99605bb279fd2e455e2c8a10643e1074094e929, count: 990, TFR: 0.037529853292, TFD: 0.037529853292}
2. BasicBlock{Hash=88629f540b9724221a6c0b43a7029276decb2f0c, count: 232, TFR: 0.008794874711, TFD: 0.008794874711}
3. BasicBlock{Hash=3b48fe121b3cd7d3e3ec568778ae35a55df5a87e, count: 344, TFR: 0.013040676296, TFD: 0.008393762452}
        1. Family{name: IRCbot} BasicBlock{Hash=3b48fe121b3cd7d3e3ec568778ae35a55df5a87e, count: 3, TFR: 0.000263042525, TFD: -0.017161505089}
        2. Family{name: Agent} BasicBlock{Hash=3b48fe121b3cd7d3e3ec568778ae35a55df5a87e, count: 1, TFR: 0.000081195193, TFD: -0.017525199752}
        3. Family{name: Dropper} BasicBlock{Hash=3b48fe121b3cd7d3e3ec568778ae35a55df5a87e, count: 10, TFR: 0.000133187715, TFD: -0.017421214709}
        4. Family{name: Virut} BasicBlock{Hash=3b48fe121b3cd7d3e3ec568778ae35a55df5a87e, count: 8, TFR: 0.000507936508, TFD: -0.016671717123}
        5. Family{name: ATRAPS} BasicBlock{Hash=3b48fe121b3cd7d3e3ec568778ae35a55df5a87e, count: 6, TFR: 0.000357824427, TFD: -0.016971941284}
        6. Family{name: ULPM} BasicBlock{Hash=3b48fe121b3cd7d3e3ec568778ae35a55df5a87e, count: 54, TFR: 0.000573461477, TFD: -0.016540667185}
        7. Family{name: Unruy} BasicBlock{Hash=3b48fe121b3cd7d3e3ec568778ae35a55df5a87e, count: 1, TFR: 0.000010857174, TFD: -0.017665875791}
        8. Family{name: Fareit} BasicBlock{Hash=3b48fe121b3cd7d3e3ec568778ae35a55df5a87e, count: 19, TFR: 0.00029312393, TFD: -0.01710134228}
        9. Family{name: Gobot} BasicBlock{Hash=3b48fe121b3cd7d3e3ec568778ae35a55df5a87e, count: 58, TFR: 0.002038592668, TFD: -0.013610404803}
        10. Family{name: Crypt} BasicBlock{Hash=3b48fe121b3cd7d3e3ec568778ae35a55df5a87e, count: 15, TFR: 0.000062546649, TFD: -0.01756249684}
        11. Family{name: Bancos} BasicBlock{Hash=3b48fe121b3cd7d3e3ec568778ae35a55df5a87e, count: 11, TFR: 0.000325145577, TFD: -0.017037298986}
4. BasicBlock{Hash=ebd4c8c2640ad80791d44ed1a7f818e1c08313ab, count: 218, TFR: 0.008264149513, TFD: 0.008264149513}
5. BasicBlock{Hash=517929ec5e483f5e6865568da0d13b10aa5a89fb, count: 212, TFR: 0.008036695857, TFD: 0.008036695857}
```

Listing 3. SALITY FAMILY, DETAILS ON THE OCCURRENCES IN THE OTHER MALWARE FAMILIES, OF THE HIGHEST 5 TFD VALUES

TABLE II. SALITY FAMILY, *v*:3b48fe121b3cd7d3e3ec568778ae35a55df5a87e SEMANTIC EQUIVALENT

| Code | $nop$<br>$mov(eax, none(eax-4))$<br>$nop$ |
|---|---|
| Semantics | $eax = none(eax-4)$ |
| Juice | $A = none(A-B)$ |
| Hash value of Juice | 3b48fe121b3cd7d3e3ec568778ae35a55df5a87e |

pattern identification. As part of our experiments, we have implemented the IG algorithm, as applied by I. Santos et al.[12], to learn about the shared code, and the main flaw of this algorithm in the context of shared code which we observed, is not considering the significant of the frequency ratio in between the malware families. However, this flaw may have not affected the work presented by I. Santos et al.[12], as they studied the opcode sequences, and not the basic blocks. In this paper, we aim to identify the critical malicious patterns as basic blocks, as CFG of every malware variant carries information regarding its functionality. Therefore, it provide us with informative pieces of critical malicious codes for each malware family.

The main drawback of our approach is the size of the database. Obviously, more number of malware variants, and malware families, will provide more accurate information regarding the critical malicious patterns. Nonetheless, in this paper we are proofing our proposed concept. However, malicious software includes more information than the shared code, such as strings, the information contained in the import-export tables of the malware, the file type, etc. We contend that our proposed approach is simple, and yet accurate, and effective. The presented approach can be applied to other features of the malicious software, by extracting the feature of interest, computing its hash value, and applying the TFD to obtain the pattern for each malware family. Thus, our presented methodology can be used to retrieve critical information about each malware family, which is essential to understand malicious software. Also, generating an effective general pattern for the numerous variants of each malware family.

The evaluation of our approach is simple, as the formalisation is based on the statistics. The counts and the implementation outcomes are checked manually, and they are accurate. The choices of the presented malware family, and the basic block are in principle arbitrary. Due to the automation the flowchart of the experimental process, is essentially trivial.

## IX. Conclusion and Future Work

In this paper, we presented a statistical approach to generate, and discover critical malicious patterns in malware families. We presented the critical malicious pattern for each malware family, as a list of basic blocks, which represent the most frequent shared code among each malware family by considering its occurrences in other malware families. Here, we present the impact of occurrence of a basic block in different malware families, on its potential to be a candidate of critical malicious pattern. Our developed framework to extract the most frequent shared code, can be applied to any other feature of malware for further analysis. The generated critical malicious patterns, are the initiative for our future work, towards the classification, and detection of known and unknown malware. Also, in the future work, we consider to extend the size of the database.

## Acknowledgment

The authors would like to thank Golnaz Badkoubeh for her insightful comments on the formalisation method, to thank Ehsun Behravesh, Aristide Fattori, Danut Niculae, and Michal Sroka for their generous help in the implementation of the framework, and to thank Arun Lakhotia and Vivek Notani for their support and willing to provide the VirusBattle SDK platform. Also, to thank the anonymous reviewers for the helpful suggestions.

## References

[1] F. Cohen, “Computer Viruses: Theory and Experiments,” Computers & Security 6, 1987, pp. 22–35.

[2] C. Beek et al., “McAfee Labs Threats Report,” McAfee Labs Threats Report, Tech. Rep., Aug. 2014.

[3] O. f. E. Co-operation and Development, Computer Viruses and Other Malicious Software: A Threat to the Internet Economy. OECD, Mar. 2009.

[4] G. J. Tesauro, J. O. Kephart, and G. B. Sorkin, “Neural networks for computer virus recognition,” IEEE Expert, vol. 11, no. 4, Aug. 1996, pp. 5–6.

[5] J. Z. Kolter and M. A. Maloof, “Learning to Detect and Classify Malicious Executables in the Wild,” J. Mach. Learn. Res., vol. 7, Dec. 2006, pp. 2721–2744.

[6] S. Cesare and Y. Xiang, “Malware Variant Detection Using Similarity Search over Sets of Control Flow Graphs,” in Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on, Nov. 2011, pp. 181–189.

[7] C. Miles, A. Lakhotia, C. LeDoux, A. Newsom, and V. Notani, “VirusBattle: State-of-the-art malware analysis for better cyber threat intelligence,” in Resilient Control Systems (ISRCS), 2014 7th International Symposium on. IEEE, 2014, pp. 1–6.

[8] A. Lakhotia, M. D. Preda, and R. Giacobazzi, “Fast location of similar code fragments using semantic ’juice’,” in Proceedings of the 2nd ACM SIGPLAN Program Protection and Reverse Engineering Workshop. ACM, 2013, p. 5.

[9] A. Moser, C. Kruegel, and E. Kirda, “Limits of Static Analysis for Malware Detection,” in Computer Security Applications Conference, 2007. ACSAC, Twenty-Third Annual, Dec. 2007, pp. 421–430.

[10] G. Vigna, “Static Disassembly and Code Analysis,” in Malware Detection, ser. Advances in Information Security, M. Christodorescu, S. Jha, D. Maughan, D. Song, and C. Wang, Eds. Springer US, 2007, vol. 27, pp. 19–41.

[11] S. L. Blond, A. Uritesc, C. Gilbert, Z. L. Chua, P. Saxena, and E. Kirda, “A Look at Targeted Attacks Through the Lense of an NGO,” in 23rd USENIX Security Symposium (USENIX Security 14). San Diego, CA: USENIX Association, 2014, pp. 543–558.

[12] I. Santos et al., “Idea: Opcode-Sequence-Based Malware Detection,” in Engineering Secure Software and Systems, ser. Lecture Notes in Computer Science, F. Massacci, D. Wallach, and N. Zannone, Eds. Springer Berlin / Heidelberg, 2010, vol. 5965, pp. 35–43.

# Systemic A Priori Patterns

Laurent Gasser
Department Tools and Techniques
IRT-SystemX
Palaiseau, France
e-mail: laurent.gasser@irt-systemx.fr

***Abstract.* — Relevant patterns in system architecture have a very broad validity. Identifying or leveraging them across application fields is challenging, though. System architects sufficiently knowledgeable in another field have to establish analogies with their own experience to confirm patterns. To let such discoveries happen by serendipity is inefficient. A systemic framework of a priori transformations guides the investigators mining models. From the repeating discoveries emerges a formal pattern. This method is illustrated by the transfer content pattern. Systemically mining for the content transfer pattern brought up variants only seen in field occurrences.**

***Keywords-systemic framework; pattern; content transfer; model based systems engineering; MBSE; behavior; SysML; activity diagram.***

## I. Introduction

Architecture patterns are valuable assets for complex systems design. As inspiring solutions, they improve the products based on them while reducing design effort. Up to now, the production of patterns is often a lucky emergence from random cross-fertilization of application fields. The subject is worth exploring in a more methodical way. We identify a priori patterns in a systemic way, putting a portrait-robot in hands of model miners. Occurrences of these pattern candidates are searched for in various application fields. The observed cases are then consolidated in a field-proven pattern, as illustrated with the content transfer pattern. A systemic catalogue of patterns is brought to systems architects by a method which also helps establishing their field-proven validity.

We expect that creating such a systemic framework will accelerate the discovery of new patterns in the same way that Mendeleev’s periodic table did for chemical elements.

The very definition of a pattern for systems engineering has been debated at first by Cloutier and Verma [1], being close of reuse, heuristics, and framework among other concepts. Since then, it has gained in acceptance [2][3][4]. Results are still far from the ones in computer science. How can it be that systems engineering fails to be more productive in term of defined patterns?

Unlike computer science, systems engineering deals with systems transforming flows of matter or energy in addition to information. The slow emergence of patterns for systems architects may come from the difficulty to reach the expertise level necessary to formulate patterns in several application fields. Good practices are promoted by engineer's education and professional associations. To leverage them across application fields requires multidisciplinary experts, a not-so-common career profile. Often, patterns are strongly influenced by the modeling language or by the application field paradigm. Establishing analogies across these involves ontological translations at several levels. Consequently, patterns are mainly found by serendipity: patterns emerge instead of being hunted for.

We propose to circumvent the difficulty by reverting the quest: instead of asking whether a field-proven pattern could be useful in another application field, occurrences of systemic a priori patterns are looked for in several application fields.

We are aware that practitioners in other fields may be reluctant to adopt patterns found by our method. The effort to qualify and inscribe them in each application field is left to their respective practitioners.

The rest of this paper is organized as follows. Section II shows a general framework of patterns induced by an a priori systemic categorizing of flows and transformations. Section III reports on several field cases of content transfer. Section IV generalizes the content transfer pattern from these cases, with a discussion on its possible variants. The conclusion and acknowledgement close the paper.

## II. A Priori Pattern Framework

Being implied in several application fields to detect analog models is a precious experience difficult to organize. Collecting patterns that way is putting one’s faith into serendipity. We aim at exploring potential patterns in a more systemic inventory [5]. We were well influenced by German teaching manuals [6][7] or the Schaum’s series world-wide famous among students [8]. They describe a large number of study cases, often presenting alternative solutions to a single problem, which cover most encountered situations during one’s professional life.

The generalized identification of flow nature and the categories over which they transform is explicitly described by Le Moigne [9] in Figure 5.1 via the transformations that may impact a flow of matter, energy, or information over time, space, or form. Le Moigne agglomerates matter and energy in transformations, though. To us, a transformation may be positive, neutral, or negative in its effect. A positive transformation augments, increases, or intensifies the resulting flow. A negative transformation fragments, scatters, or corrupts the resulting flow. Table 1 proposes a function for each of the cases systemically identified a priori.

TABLE I. A PRIORI PROCESSING TRANSFORMATIONS

| Transformation | Matter | Energy | Information |
|---|---|---|---|
| **Time** | + collect | + accumulate | + learn |
| | = preserve, conserve | = store | = memorize |
| | - spoil, decompose | - dissipate | - forget |
| **Space** | + expand, | + diffuse, propagate | + broadcast, disseminate |
| | = transport, anchor | = transfer, distribute | = move |
| | - confine, condense, concentrate | - concentrate | - isolate, hide, segregate, disturb |
| **Form** | + mold, shape | + react, refine | + enrich, augment, decorate |
| | = replicate | = transform, convert | = translate, transcript |
| | - erode | - degrade, disturb | - corrupt, sample, simplify, vulgarize |

TABLE II. A PRIORI TRANSFORMATIONS OF FLOW NATURE

| Transformation | Matter | Energy | Information |
|---|---|---|---|
| **Matter** | + purify, combine | + compress | + detect, sense |
| | = transmute, react | = burn, explode | = measure |
| | - pollute, reduce, lyse | - absorb (shock) | - erase, spoil |
| **Energy** | + centrifuge, distill | + generate, ignite | + refresh |
| | = fuse, melt | = convert, exchange | = detect |
| | - dry | - decay, degenerate, deteriorate | - perturb |
| **Information** | + cook | + optimize | + decode, uncompress |
| | = actuate | = command, control, pilot, solicit | = process, synthesize |
| | - screen, filter, sort | - disperse, overload | - scramble, disturb |

Table 1 reads as follows: the positive (resp. neutral or negative) transformation (in the cell) applies to the flow (column) over a dimension (row). The boilerplate "<function> is a <effect> transformation of <flow> over <dimension>" applied to the upper left item becomes: collect is a positive transformation of matter over time.

A transformation of nature may convert one input flow in an output flow of another nature. The transformation of flow nature may be positive, neutral, or negative in its result.

Table 2 reads as follows: the positive (resp. neutral or negative) transformation (in the cell) of the input flow (in the row) into the output flow (in the column). The boilerplate "<function> is a <effect> transformation of <input flow> into <output flow>" applied to the lower left item becomes: filter is a transformation of information into matter.

The method to exploit these tables starts by selecting one a priori pattern in the tables. It is then looked for in several application fields. Approaching occurrences are collected and compared. A consolidated pattern emerges from the collected cases and is formalized.

## III. ILLUSTRATIVE FIELD CASES

Let us select the middle line of Table 1 as the a priori pattern to be searched for in application fields: a neutral transformation of matter, energy, or information over space.

Industrial systems are full of containers whose content has to be transferred. Would the content be matter, energy, or information, it may flow as a stream (uninterrupted over time) or not (packet by packet). Let us shortly sketch one field case for each combination.

Biomedical devices mix blood with reagents to identify pathogens present in the blood sample [10]. During sample preparation, a precise amount of blood is sucked into the pipette and then flushed into the sample tube. Liquid streams from the blood sample into the pipette, from the pipette into the sample tube: a streaming flow of matter.

A cargo ship enters a harbor to embark or unload fret containers [11][12]. The fret containers are stored on the harbor shore, picked up by a crane which transfers them to and from the cargo ship. Fret containers are taken one by one from an open space (the harbor shore) to another (the cargo ship store-room): a non-streaming flow of matter.

In a thermic power plant producing electricity, the primary circuit extracts the heat from the core to produce high temperature and pressure steam [13][14]. The steam expands then into a turbine coupled to a dynamo (electricity generator). The heat streams from the core source into the steam before being converted into mechanical work (steam expansion): a streaming flow of energy.

Rail transport collects freight cars from their origin to bring them to destination. Since each freight car may have its own destination, they are recombined at a classification yard. Each freight car is uncoupled from its origin train, pushed by a rail tractor through a set of points to reach its destination train and locks to it. To minimize the sorting time, the turnouts are remotely commanded, the rail tractor initiate the motion by a push and let it roll to its destination train. Retarding apparatus along the rails absorb excessive speed by offering resistance to freight car progress. The freight car locks to the destination train without harsh bumping. The tractor impulses kinematic energy to the freight car whose momentum is then controlled to ensure locking to its train: a non-streaming flow of energy.

Television operators broadcast video content as part of their broadcast program or on demand [15]. The video is sent from the television operator to all receivers in an uninterrupted stream of images. Depending on the subscriber contract and order, the content is displayed on television. The operator broadcasts the video in a stream to all receivers: a streaming flow of information.

In large organizations, the information technology department manages the software installed on each computer of the company [16]. When a new release of a software is deployed, a central script copies the new software application to each of the targeted computer. The new software is installed remotely onto each supported computer: a non-streaming flow of information.

Whatever the scale and the nature of the flow, the content of a source container is transferred into a destination container by a transferor.

## IV. THE CONTENT TRANSFER PATTERN

As the examples of the previous section illustrate, transferring content from one container to another is found in many applications varying in size, purpose, and nature of flow. In this section, the content transfer pattern is generalized from the different occurrences that have been cited in Section 3. These have been found by looking after field occurrences of the middle line of Table 1.

The pattern involves at least two containers and the transferor which transfers the content from the source container to the destination container (see Figure 1). All figures are in SysML [17], the modeling language promoted by INCOSE and dedicated to systems architects. The 'Content Transfer Pattern' block is not part of the pattern instantiation: it simply packs the components of the pattern together. The 'Containers' and 'Transferor' blocks are the physical elements of the pattern. In the biomedical device above, the pipette is the transferor from the blood sample (the source container) to the sample tube (its destination).

The transferor gets the content from the source container and transfers it to the destination container. The 'Transferor' provides the 'transfer' function, which is directly taken from Table 1. In the power plant example, the transformation is about space: heat is extracted from the core to be lead into the turbine. A 'Container' provides two auxiliary services: the functions 'host' and 'release' content.

Note that the transferor somehow holds content while passing it through. But, this is not considered as its primary function which is to transfer content or let content be transferred. The transferor is connected to both the source and the destination, as in Figure 2. The connection is a flow interface of 'content' type: information, energy, or matter depending on the flow nature. This artefact is a fret container (matter) in the context of the cargo ship.

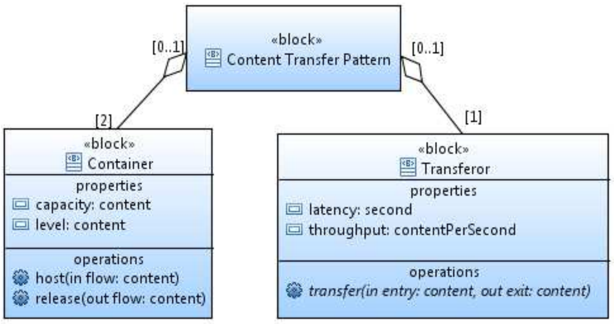


Figure 1. Elements of the Content Transfer Pattern (bdd).

A pattern variant includes a path conducting the content flow fore and aft of the transferor. In another variant, the transferor uses the path to go from one container to the other. The path properties would be its capacity size and length.

By looking at the illustrative field cases, it seems useful to differentiate three specialized transferors according to the propensity of the flow to naturally go from the source to the destination container (see Figure 3). This refinement of the a priori pattern comes from the field cases.

When an adverse potential exists (against the spontaneous transfer from source to destination), the flow from the source has to be pumped up the potential to reach the destination.

When a potential aids to move from the source to the destination, the transferor takes the role of a valve allowing more or less of the flow to pass by.

When no potential difference is noticed, the flow is conveyed from one place to another.

Another point worth mentioning resides in the container behavior. Trivially, a container passively contains its content. Seen as a service provider, the container hosts or releases a content flow. The realized containers always have a limited capacity and a current level of content. A container may be initially full, partially full, or empty. A container is either reversible (a cistern can be loaded and unloaded at will) or not (a non-reloadable battery releases its energy once by irreversible chemical reaction). Three operating modes are defined: empty, full, or containing (see Figure 4).

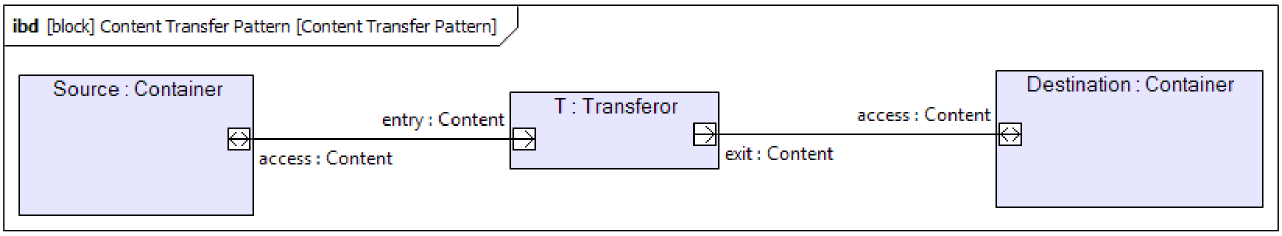


Figure 2. Topology of the Content Transfer Pattern.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-393-3

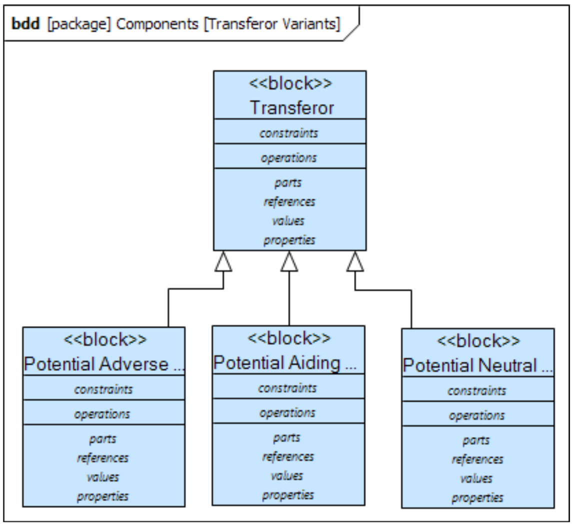


Figure 3. Variants of Transferor

In most cases, the transferor has a unique operating mode, the transfer taking place or not. Reversing the flow in a pump to make it a generator justifies two operating modes of the transferor, though.

The content transfer scenario is rather straightforward (see Figure 5). The two services of the container are used: the ability to accept a content (function "host") and the ability to deliver a part of its content (function "release"). Exceptions occur when the request goes beyond the current content level of the source or exceeds the destination capacity. The transfer content pattern is described in Table 3 using the template proposed in [1].

As is stands, the content transfer pattern can be applied to a wide variety of problems. Electricity distribution has long been a stable situation of potential aiding transferor. When politics encouraged everyone to distribute and produce electricity without centralized planning, the electric flow potential has become versatile. Nowadays, a transferor has to adapt to fluctuant potentials, developing strategies to switch between its operational modes according to the ongoing production and consumption.

A similar evolution has happened with Internet: knowledge is now produced and consumed by visitors to sites like Wikipedia. The rupture is caused by the change in knowledge potential, an important factor in the instantiation of the content transfer pattern.

Daily commuters are experiencing traffic jam in most cities. The storing capacity of the city is not the problem since all commuters finally find place and spend several hours in the city. The transfer content pattern underlines that the destination is hosting an incoming flow with a limited throughput.

Without the content transfer pattern these disruptive cases could not so easily be understood.

## V. CONCLUSION

Usually, collecting architecture patterns from many sources is tedious and progress is fortuitous. In addition, best practices in pattern elicitation recommend that the interest in a pattern candidate be confirmed by many sources. The proposed method elaborates patterns from the systemic a priori Tables 1 and 2. It offers a complementary approach to mining them from experience. The illustrative field cases have been found by a systemic a priori and focused investigation. The variants with respect to flow potential emerged from the field cases. It was not implied by the a priori pattern.

The resulting transfer content pattern gained in applicability scope by the very way it was found. Based on the systemic a priori tables, new patterns (like transforming container: cooking recipes, chemical reactors) can be methodically searched for.

Furthermore, it can be argued that a truly systemic approach starts with a finality from which everything else is derived, model included. The systems architect's action is led by his intent to identify new patterns from the presented a priori framework.

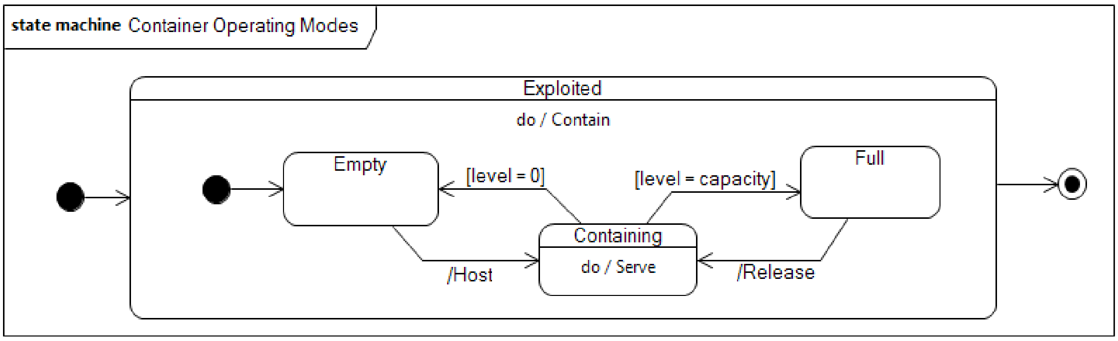


Figure 4. Operating Modes of a Container.

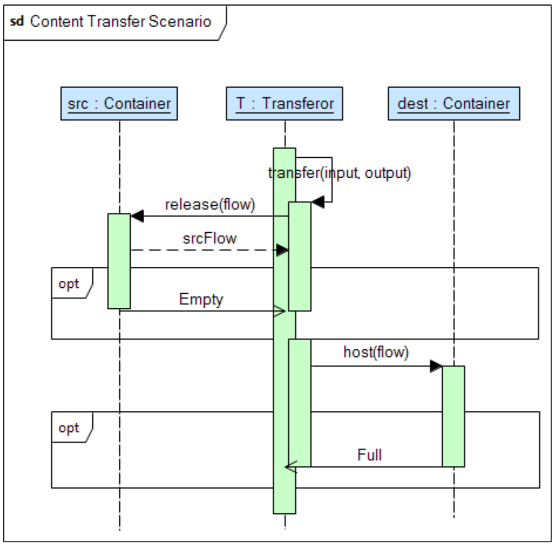


Figure 5. Scenario of Content Transfer.

TABLE III. CONTENT TRANSFER PATTERN

| | |
|---|---|
| **Pattern Name** | Content Transfer |
| **Aliases** | None yet |
| **Keywords** | Container, content, transfer, flow potential |
| **Problem Context** | Most industrial processes include some transfer from one step to another. The nature of the content to be transferred is diverse, with transferring means adapted to the adverse, neutral, or aiding flow potential. |
| **Problem Description** | The way some content is transferred from one container to another depends on the flow potential and the means of transfer. |
| **Forces** | Transferor names are strongly influenced by disciplines and flow potential. The flow control may require processing algorithms specific to the application case. |
| **Pattern Solution** | The services provided by the containers are explicited. The category of transferor depends on the flow potential. Exceptional cases are identified. |
| **Model** | See figures in Section 4. |
| **Interfaces** | The mandatory flow is the content released from the source to be hosted by the destination through the transferor. Additionnal information about the levels of containers as well as control to the transferor may appear. The connections to the containers may be permanent. |
| **Resulting Context** | Depending on the flow potential and the means of transfer, additional flows (like energy or anchoring support to a pump) and components (like pipes) are required. |
| **Example** | Gas filling into a car at the gasoil station.The flow potential is adverse, so the transferor is a pump. The source container is the station cistern, the destination is the car tank. The flow of gas (matter) is streaming. |
| **Known Uses** | See illustratative field cases in Section 3 |
| **Related Patterns** | None yet |
| **References** | This paper. |
| **Pattern Rationale** | The pattern is found from a systemic a priori candidate pattern. |
| **Author** | Laurent Gasser |

## ACKNOWLEDGMENT

We thank IRT-SystemX for approving the publication of this personal research. We also thank the reviewers whose constructive comments helped to improve the presentation of this method.

## REFERENCES

[1] J. R. Cloutier and D. Verma, “Applying the Concept of Patterns to Systems Architecture”, Journal of Systems Engineering, vol. 10, no. 2, 2007, pp. 138–154.

[2] C. Haskins, “Using Patterns to Transition Systems Engineering from a Technological to Social Context”, Journal of Systems Engineering, vol. 11, no. 2, 2011, pp. 147–155.

[3] F. Pfister, V. Chapurlat, M. Huchard, C. Nebut, and J.-L. Wippler, “A Proposed Meta-Model for Formalizing Systems Engineering Knowledge, Based on Functional Architectural Patterns”, Journal of Systems Engineering, vol. 15, no. 3, 2012, pp. 321–332.

[4] T. Stålhane, O. Daramola, and V. Katta, “Pattern in Safety Analysis”, The Fourth International Conferences on Pervasive Patterns and Applications (PATTERNS 2012) IARIA, July 2011, pp. 7-10, ISBN: 978-1-61208-221-9.

[5] A. Hatchuel and B. Weil, “A new approach of innovative design: an introduction to C-K theory”, ICED’03, Stockholm, Sweden 14, 2003, pp. 109-124.

[6] E. Neufert and P. Neufert, “Bauentwurfslehre, 37. Auflage“, Friedr. Vierweg & Sohn, 2002, ISBN: 3-528-98651-4

[7] M. Gressmann, “Fachkunde Fahrradtechnik, 3. Auflage“, Europa-Lehrmittel, 2009, ISBN: 978-3-8085-2293-6

[8] G. Buchanan, “Shaum’s Outline of Theory and Problems of Finite Element Analysis“, McGraw-Hill, 1995, ISBN: 0-07-008714-8

[9] J.-L. Le Moigne, “La théorie du système général (4ème éd.)“, Presses Universitaires de France, 1994, (first edition 1977, at http://www.mcxapc.org/inserts/ouvrages/0609tsgtm.pdf, retrieved: January, 2015).

[10] Biomérieux, Vidia User’s Guides ref 38087-B-2006/07, 2006.

[11] J. W. Böse (ed) and B. Brinkmann, “Handbook of Terminal Planning”, Operations Research/Computer Science Interfaces Series 49, Springer, 2011, ISBN: 978-1-4419-8408-1_2

[12] E. Zehendner, “Operations management at container terminals using advanced information technologies”, Ecole Nationale Supérieure des Mines de Saint-Etienne, 2013, NNT: 2013EMSE0714.

[13] Presse Polytechnique Romande, “Guide de la technique 3, l’énergie”; PPR, 1993, ISBN: 2-88074-221-8

[14] International Energy Agency, “Technology roadmap, Solar therma electricity, 2014 edition”, IEA

[15] H. Cruickshank, M.-P. Howarth, S. Iyengar, and Z. Sun, “A comparison between satellite DVB conditional access and secure IP multicast”, 14th IST Mobile and Wireless Communications Summit, 2005, pp. 19–22.

[16] A. Dearle, “Software deployment, past, present and future”, in 2007 Future of Software Engineering, IEEE Computer Society, 2007, pp. 269–284.

[17] OMG, “OMG Systems Modeling Language, version 1.3”, Object Management Group, June 2012.

# A Review of Audio Features and Statistical Models Exploited for Voice Pattern Design

Ngoc Q. K. Duong

Technicolor
975 avenue des Champs Blancs
35576 Cesson Sévigné, France
Email: quang-khanh-ngoc.duong@technicolor.com

Hien-Thanh Duong

Faculty of Information Technology
Hanoi University of Mining and Geology
Hanoi city, Vietnam
Email: duongthihienthanh@humg.edu.vn

*Abstract*—**Audio fingerprinting, also named as audio hashing, has been well-known as a powerful technique to perform audio identification and synchronization. It basically involves two major steps: fingerprint (voice pattern) design and matching search. While the first step concerns the derivation of a robust and compact audio signature, the second step usually requires knowledge about database and quick-search algorithms. Though this technique offers a wide range of real-world applications, to the best of the authors' knowledge, a comprehensive survey of existing algorithms appeared more than eight years ago. Thus, in this paper, we present a more up-to-date review and, for emphasizing on the audio signal processing aspect, we focus our state-of-the-art survey on the fingerprint design step for which various audio features and their tractable statistical models are discussed.**

***Keywords–Voice pattern; audio identification and synchronization; spectral features; statistical models.***

## I. Introduction

Real-time user interactive applications have emerged nowadays thanks to the increased power of mobile devices and their Internet access speed. Let us consider applications like music recognition [1][2], e.g., people hear a song in a public place and they want to know more about it, or personalized TV entertainment [3][4], e.g., people want to see more service and related content on the Web in addition to the main view from TV; both require a fast and reliable audio identification system in order to match the observed audio signal with its origin stored in a large database. For these purposes, several research directions have been studied, such as audio fingerprinting [5], audio watermarking [6], and timeline insertion [4]. While watermarking and timeline approaches both require to embed signature into the original media content, which is sometimes inconvenient for the considered applications, fingerprinting technique allows directly monitoring the data for identification. Hence, audio fingerprinting has been widely investigated in the literature and already been deployed in many commercialized products [1][7][8][9][10][11]. This technique has recently been exploited for other applications such as media content synchronization [12][13], multiple video clustering [14], repeating object detection [15], and live version identification [15].

A general architecture for an audio fingerprinting system, which can be used for either audio identification or audio synchronization purpose, is depicted in Fig. 1. The fingerprint extraction derives a set of relevant audio features followed by an optional post-processing and feature modeling. Fingerprints of the original audio collection and its corresponding metadata (e.g., audio ID, name, time frame index, etc.) are systematically stored in a database. Then given a short recording from the user side, its feature vectors (i.e fingerprints) are computed in the same way as they were for the original data. Finally, a searching algorithm will find the best match between these fingerprints with those stored in the database so that the recorded audio signal is labeled by the matched metadata.

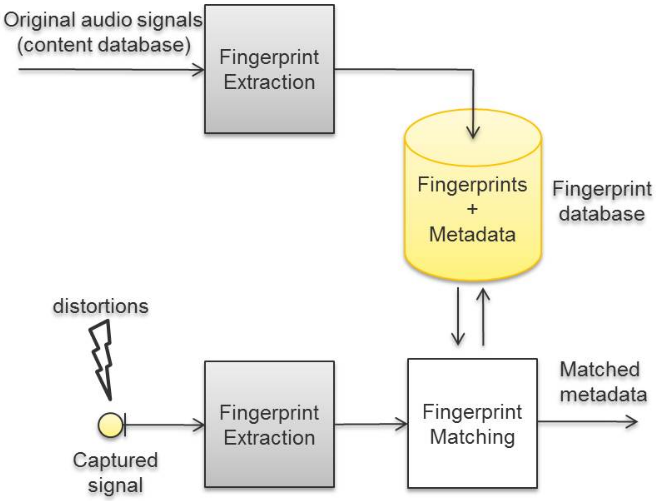


Figure 1: General architecture of an audio fingerprinting system.

In real-world recording, the audio signal often undergoes many kinds of distortion: acoustical reverberation, background noise addition, quantization error, etc. Thus, the derived fingerprints must be robust with respect to these various signal degradations. Beside, the fingerprint size should be as small as possible to save memory resources and to allow real-time matching. The details of general properties of the audio fingerprint was well-discussed in [1][16][17]. In order to fulfil those requirements, audio sample signal is often transformed into Time-Frequency (T-F) domain via the Short Time Fourier transform (STFT) [16] where numerous distinguishable characteristics such as high-level musical attributes, e.g., predominant pitch, harmony structure, or low level spectral features, e.g., mel-frequency cepstrum, spectral centroids, spectral note onsets, etc., are exploited. To further compact the fingerprints, some approaches continue to fit the spectral feature vectors to a statistical model, e.g., Gaussian Mixture Model (GMM) [18], Hidden Markov Model (HMM) [19], so that in the end only the set of model parameters are used as fingerprints.

Though diverse fingerprinting algorithms have been proposed in the literature, the number of review papers remains limited where, to the best of the authors' knowledge, a comprehensive review of fingerprinting algorithms was presented more than eight years ago [5][16], and a more recent survey [20] only focusing on computer vision based approaches (e.g., methods proposed in [21][22]). In this paper, we present a more up-to-date review of the domain, with particular focus concerning the fingerprint extraction block in Fig. 1, where various audio spectral features and their statistical models are summarized systematically. The presentation would particularly benefit new researchers in the domain and engineers in the sense that they would easily follow the described steps to implement different audio fingerprints.

The structure of the rest of the paper is as follows. We first present a general architecture for fingerprint design in Section II, we then review various audio features, which have been extensively exploited in the literature, in Section III. The detail of some statistical feature models is introduced in Section IV. Finally, we conclude in Section V.

## II. General architecture of fingerprint design

Fig. 2 depicts a general workflow of the fingerprint design. The purpose of each block is summarized as follows:

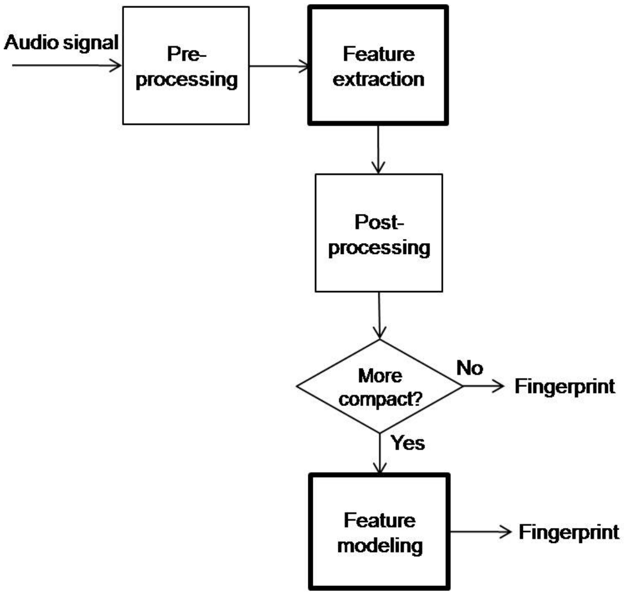


Figure 2: General workflow of the fingerprint design.

- *Pre-processing*: in this step, input audio signal is often first digitalized (if necessary), re-sampled to a target sampling rate, and bandpass filtered. Other types of processing includes decorrelation and amplitude normalization [16]. Then the processed signal is segmented into overlapping time frames where a linear transformation, e.g., Fast Fourier Transform (FFT), Discrete Cosine transform (DCT), or wavelet transform [16], is applied to each frame. At this stage, the input time-domain signal is represented in a feature domain, and the most popular feature domain is time-frequency representation given by the STFT.
- *Feature extraction*: this is a major process since the choice of "which feature is used" will directly affect the performance of the entire fingerprinting system. A great diversity of features have been investigated targeting the reduction of dimensionality as well as the invariance to various distortions. For summary, most approaches first map the linear time-frequency representation given by the STFT to an auditory-motivated frequency scale, i.e., Mel, Bark, Log, or Cent scale, via filterbanks [2][23]. This mapping step greatly reduces the spectrogram size since the number of filterbanks is usually much smaller than the FFT length. Then a feature vector such as Mel-Frequency Cepstral Coefficients (MFCC), spectral centroids of all subbands, etc., are computed for each time frame. In some systems, the first and second derivatives of the feature vectors are also integrated to better track the temporal variation of audio signals [16][18]. Other types of feature that worth mentioning are e.g., time localized frequency peak [24], time-frequency energy peak location [1], or those developed in image processing based approaches such as top-wavelet coefficients computed on the spectral image [22] and multiscale Gabor atoms extracted by Matching Pursuit algorithm [14]. Recently, a general framework for dictionary based feature learning has also been introduced [25].
- *Post-processing*: the feature vectors computed in the previous step are often real-valued and the absolute range depends on the signal power. Therefore when Euclidean distance is used in the matching step, mean substraction and component wise variance normalization are recommended [26][27]. Another popular post-processing is quantization where each entry of the feature vectors is quantized to a binary number in order to gain robustness against distortions and, more importantly, to obtain memory efficiency [2][28][22] [15]. In many existing system, fingerprint is achieved after this step.
- *Feature modeling*: this block is sometimes deployed in order to further compact the fingerprint. In this case, a large number of feature vectors along time frames is fitted to a statistical model so that an input audio signal is well-characterized by the model parameters, which are then stored as a fingerprint [29][18][30]. Popular model includes Gaussian Mixture Model (GMM), Hidden Markov Model (HMM). Other approaches used decomposition techniques, e.g., Non-negative MAtrix Factorization (NMF), to help decreasing data dimension and therefore to reduce the local statistical redundancy of the feature vectors [31][32].

Since the pre-processing and post-processing steps are quite straightforward, in the following of the paper we will present more detail only on the feature extraction and the feature modeling blocks.

## III. Feature extraction

Summarizing numerous types of audio features used for the fingerprint design so far will certainly go beyond the scope of this paper. Thus in this section, we select to present the most popular low level features in the spectral domain only.

### A. MFCC

MFCC is one of the most popular feature considered in speech recognition where the amplitude spectrum of input audio signal is first weighted by triangular filters spaced according to the Mel scale, and DCT is then applied to decorrelate the Mel-spectral vectors. MFCC was shown to be applicable for music signal also in [33]. Examples of fingerprinting algorithms used MFCC feature are found in [33][18]. In [34], MFCC was used also for clustering and synchronizing large scale audio-video sequences recorded by multiple users during an event. Matlab implementations for the computation of MFCC are available [35][36].

### B. Spectral Energy Peak (SEP)

SEP for music identification systems was described in [37][1] where a time-frequency point is considered as a peak if it has higher amplitude than its neighboring points. SEP is argued to be intrinsically robust to even high level background noise and can provide discrimination in sound mixtures [38]. In well-known Shazam's system [1] time-frequency coordinates of the energy peaks was described as sparse landmark points. Then by using pairs of landmark points rather than single points, the fingerprints exploited the spectral structure of sound sources. This landmark feature can also be found in [14] and [39] for multiple video clustering. Ramona *et al.* used start times of the spectral energy peaks, referred to as onsets, for the automatic alignment of audio occurrences in their fingerprinting system [23][40].

### C. Spectral Band Energy (SBE)

Together with spectral peak, SBE has been widely exploited in fingerprinting algorithms. Let us denote by $s(n, f)$ a STFT coefficient of an audio signal at time frame index $n$ and frequency bin index $f$, $1 \leq f \leq M$. Let us also denote by $b$ an auditory-motivated subband index, i.e., in either Mel, Bark, Log, or Cent scale, and $l_b$ and $h_b$ the lower and upper edges of $b$-th subband. SBE is then computed, with normalization, in each time frame and each frequency subband range by

$$F_{n,b}^{\text{SBE}} = \frac{\sum_{f=l_b}^{h_b} |s(n,f)|^2}{\sum_{f=1}^{M} |s(n,f)|^2}. \tag{1}$$

Haitsma et al. proposed a famous fingerprint in [2] where SBEs were first computed in a block containing 257 time frames and 33 Bark-scale frequency subbands, then each $F_{n,b}^{\text{SBE}}$ was quantized to a binary value (either 0 or 1) based on its differences compared to neighboring points. Other fingerprinting algorithms exploiting SBE feature were found for instance in [41][18]. Variances of this subband energy difference feature can be found in more recent approaches [28][21].

### D. Spectral Flatness Measure (SFM)

SFM, also known as Wiener entropy, relates to the tonality aspect of audio signals and it is therefore often used to distinguish different recordings. SFM is computed in each time-frequency subband point $(n, b)$ as

$$F_{n,b}^{\text{SFM}} = \frac{\left(\prod_{f=l_b}^{h_b} |s(n,f)|^2\right)^{\frac{1}{h_b - l_b + 1}}}{\frac{1}{h_b - l_b + 1} \sum_{f=l_b}^{h_b} |s(n,f)|^2}. \tag{2}$$

A high SFM indicates the similarity of signal power over all frequencies while a low SFM means that signal power is concentrated in a relatively small number of frequencies over the full subband.

A similarly feature to SFM, which is also a measure of the tonal-like or noise-like characteristic of audio signal and was exploited as fingerprint, is spectral crest factor (SCF). SCF is computed by

$$F_{n,b}^{\text{SCF}} = \frac{\max_{f \in [l_b, h_b]} \left(|s(n,f)|^2\right)}{\frac{1}{h_b - l_b + 1} \sum_{f=l_b}^{h_b} |s(n,f)|^2}. \tag{3}$$

SFM and SCF were found to be the most promising features for audio matching with common distortions in [42] and were both considered in other fingerprinting algorithms [41][18].

### E. Spectral Centroid (SC)

SC is also a popular measure used in audio signal processing to indicate where the ”center of mass” of a subband spectrum is. It is formulated as

$$F_{n,b}^{\text{SC}} = \frac{\sum_{f=l_b}^{h_b} f.|s(n,f)|^2}{\sum_{f=l_b}^{h_b} |s(n,f)|^2}. \tag{4}$$

SC was argued to be robust over equalization, compression, and noise addition. It was reported in [26] and [18] that SC-based fingerprints offered better audio recognition than MFCC-based fingerprints with 3 to 4 second length audio clips. In our preliminary experiment with speech utterances distorted by reverberation and real-world background noise, we also observed that SC-based fingerprints resulted in higher recognition accuracy than MFCC-, SBR-, and SFM-based fingerprints without post-processing.

Given one of the feature parameters $F_{n,b}$ computed in each time-frequency subband point $(n, b)$ as described above, a $d$-dimensional feature vector $\mathbf{F}_n = [F_{n,1}, ..., F_{n,d}]^T$ is formed to describe the corresponding characteristic of the signal at time frame $n$, where $T$ denotes vector transpose and $d$ is the total number of subbands. When the first and second derivatives of the feature vectors are additionally considered, for better characterizing the temporal variation of audio signal, $\mathbf{F}_n$ will then be a $3d$-dimensional vector [18] before passing to the post-processing block shown in Figure 2.

## IV. Feature modeling

In some systems, in order to further compact the fingerprint the feature vectors $\mathbf{F}_n$ can be adapted to a statistical model. This step allows to reduce the global redundancy of spectral features. As a result, a long sequence of feature vectors $\mathbf{F}_n, n = 1, ..., N$ is characterized by a significantly smaller number of the model parameters while ensuring the discriminative power. In this section we review the use of three popular models, namely gaussian mixture model (GMM), hidden Markov model (HMM), and nonnegative matrix factorization (NMF), for the fingerprint design.

### A. GMM-based fingerprint

GMM has been used to model the spectral shape of audio signals in many different applications ranging from speaker identification [43] to speech enhancement [30], etc. It was also investigated for audio fingerprinting by Ramalingam and Krishnan [18], where spectral feature vectors $\mathbf{F}_n$ are modeled as a multidimensional K-state Gaussian mixture with probability density function (pdf) given by

$$p(\mathbf{F}_n) = \sum_{k=1}^{K} \alpha_k \mathcal{N}_c(\mathbf{F}_n|\mu_k, \mathbf{\Sigma}_k) \tag{5}$$

where $\alpha_k$, which satisfies $\sum_{k=1}^{K} \alpha_k = 1$, $\mu_k$ and $\mathbf{\Sigma}_k$ are the weight, the mean vector and the covariance matrix of the $k$-th state, respectively, and

$$\mathcal{N}_c(\mathbf{F}_n|\mu_k, \mathbf{\Sigma}_k) = \frac{1}{|\pi \mathbf{\Sigma}_k|} e^{-(\mathbf{F}_n-\mu_k)^H \mathbf{\Sigma}_k^{-1} (\mathbf{F}_n-\mu_k)} \tag{6}$$

where $^H$ and $|.|$ denote conjugate transpose and determinant of a matrix, respectively. The model parameters $\theta = \{\alpha_k, \mu_k, \mathbf{\Sigma}_k\}_k$ are then estimated in the maximum likelihood (ML) sense via the expectation-maximization (EM) algorithm, which is well-known as an appropriate choice in this case, with the global log-likelihood defined as

$$\mathcal{L}_{ML} = \sum_{n=1}^{N} \log p(\mathbf{F}_n|\theta). \tag{7}$$

As a result, the parameters are iteratively updated via two EM steps as follow:

- E-step: compute the posterior probability that feature vector $\mathbf{F}_n$ is generated from the $k$-th GMM state
$$\gamma_{nk} = \frac{\alpha_k p(\mathbf{F}_n|\mu_k, \mathbf{\Sigma}_k)}{\sum_{l=1}^{K} \alpha_l p(\mathbf{F}_n|\mu_l, \mathbf{\Sigma}_l)}. \tag{8}$$
- M-step: update the parameters
$$\alpha_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk} \tag{9}$$
$$\mu_k = \frac{\sum_{n=1}^{N} \gamma_{nk} \mathbf{F}_n}{\sum_{n=1}^{N} \gamma_{nk}} \tag{10}$$
$$\mathbf{\Sigma}_k = \frac{\sum_{n=1}^{N} \gamma_{nk} (\mathbf{F}_n - \mu_k)(\mathbf{F}_n - \mu_k)^H}{\sum_{n=1}^{N} \gamma_{nk}}. \tag{11}$$

With GMM, $N$ $d$-dimensional feature vectors $\mathbf{F}_n$ are characterized by $K$ set of GMM parameters $\{\alpha_k, \mu_k, \mathbf{\Sigma}_k\}_{k=1,\dots,K}$ where $K$ is often very small compared to N. However, since GMM does not explicitly model the amplitude variation of sound sources, signals with different amplitude level but similar spectral shape may result in different estimated mean and covariance templates. *To overcome this issue, another version of GMM called spectral Gaussian scaled mixture model (GSMM) could be considered instead. Though GSMM has been used in speech enhancement [30] and audio source separation [44], it has yet been applied in the context of fingerprinting.*

### B. HMM-based fingerprint

HMM is a well-known model in many audio processing applications [45]. When applied for audio fingerprinting, pdf of the observed feature vector $\mathbf{F}_n$ can be written as

$$p(\mathbf{F}_n) = \sum_{q_1,q_2,\dots,q_d} \pi_{q_1} b_{q_1}(F_{n,1}) a_{q_1 q_2} b_{q_2}(F_{n,2}) \\ \dots a_{q_{d-1} q_d} b_{q_d}(F_{n,d}) \tag{12}$$

where $\pi_{q_i}$ denotes the probability that $q_i$ is the initial state, $a_{q_i q_j}$ is state transition probability, and $b_{q_i}(F_{n,i})$ is pdf for a given state.

Given a sequence of observations $\mathbf{F}_n, n = 1, ..., N$ extracted from a labeled audio signal, the model parameters $\theta = \{\pi_{q_i}, a_{q_i q_j}, b_{q_i}\}_{i,j}$ are learned via e.g., EM algorithm (detail formulation can be found in [45]) and stored as a fingerprint. Cano et al. modeled MFCC feature vectors by HMM in their AudioDNA fingerprint system [29]. In [19] HMM-based fingerprint was shown to achieve a high compaction by exploiting structural redundancies on music and to be robust to distortions.

Note that when applying GMM or HMM for the fingerprint design, a captured signal at the user side is considered to be matched with an original signal fingerprinted by the model parameter $\theta$ in the database if its corresponding feature vectors $\widehat{\mathbf{F}}_n$ are most likely generated by $\theta$.

### C. NMF-based fingerprint

NMF is well-known as an efficient decomposition technique which helps reducing data dimension [46]. It has been widely considered in audio and music processing, especially for audio source separation [47][48]. When applying in the context of audio fingerprinting, a $d \times N$ matrix of the feature vectors $\mathbf{V} = [\mathbf{F}_1, ..., \mathbf{F}_N]$ is approximated by

$$\mathbf{V} = \mathbf{W}\mathbf{H} \tag{13}$$

where $\mathbf{W}$ and $\mathbf{H}$ are non-negative matrices of size $d \times Q$ and $Q \times N$, respectively, modeling the spectral characteristics of the signal and its temporal activation, and $Q$ is much smaller than $N$. The model parameters $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{H}\}$ can be estimated by minimizing the following cost function:

$$C(\boldsymbol{\theta}) = \sum_{bn} d_{IS}\left([\mathbf{V}]_{b,n} | [\mathbf{W}\mathbf{H}]_{b,n}\right), \tag{14}$$

where $d_{IS}(x|y) = \frac{x}{y} - \log \frac{x}{y} - 1$ is Itakura-Saito (IS) divergence, and $[\mathbf{A}]_{b,n}$ denotes an entry of matrix $\mathbf{A}$ at $b$-th row and $n$-th column. The resulting multiplicative update (MU) rules for parameter estimation write [49]:

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{W}^T \left( (\mathbf{W}\mathbf{H})^{\cdot -2} \odot \mathbf{V} \right)}{\mathbf{W}^T (\mathbf{W}\mathbf{H})^{\cdot -1}} \tag{15}$$

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{\left( (\mathbf{W}\mathbf{H})^{\cdot -2} \odot \mathbf{V} \right) \mathbf{H}^T}{(\mathbf{W}\mathbf{H})^{\cdot -1} \mathbf{H}^T} \tag{16}$$

where $\odot$ denotes the Hadamard entrywise product, $\mathbf{A}^{\cdot p}$ being the matrix with entries $[\mathbf{A}]_{ij}^{p}$, and the division is entrywise. Fingerprints are then generated compactly from the basis matrix $\mathbf{W}$, which has much smaller size compared to the feature matrix $\mathbf{V}$.

NMF was applied to the spectral subband energy matrix in [32] and to the MFCC matrix in [50]. The resulting fingerprint was shown to better identify audio clips than another decomposition technique namely singular value decomposition (SVD).

## V. Conclusion

In this paper, we presented a review of the existing audio fingerprinting systems which have been developed by numerous researchers during the last decade for a range of practical applications. We described a variety of audio features and reviewed state-of-the-art approaches exploiting them for the fingerprint design. Furthermore, the use of statistical models and decomposition techniques to reduce the global statistical redundancy of feature vectors, and therefore to decrease fingerprint size, was also summarized. As a result, the combination of different presenting features and/or the deployment of a statistical feature model afterward are both applicable to obtain a robust and compact audio signature.

## References

[1] A. L.-C. Wang, “An industrial-strength audio search algorithm,” in Proc. Int. Sym. on Music Information Retrieval (ISMIR), 2003, pp. 1–4.

[2] J. Haitsma and T. Kalker, “A highly robust audio fingerprinting system,” in Proc. Int. Sym. on Music Information Retrieval (ISMIR), 2002.

[3] M. Fink, M. Covell, and S. Baluja, “Social- and interactive-television. applications based on real-time ambient-audio identification,” in Proc. European Interactive TV Conference (Euro-ITV), 2006.

[4] C. Howson, E. Gautier, P. Gilberton, A. Laurent, and Y. Legallais, “Second screen tv synchronization,” in Proc. IEEE Int. Conf. on Consumer Electronics (ICCE), 2011.

[5] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, “A review of algorithms for audio fingerprinting,” in IEEE Workshop on Multimedia Signal Processing, 2002, pp. 169–173.

[6] H.J.Kim, Y.H.Choi, J.W.Seok, and J.W.Hong, “Audio watermarking techniques,” in Intelligent Watermarking Techniques, 2004, ch. 8, pp. 185–218.

[7] R. Macrae, X. Anguera, and N. Oliver, “Muvisync: Realtime music video alignment,” in Proc. IEEE Int. Conf. on Multimedia and Expo (ICME), 2010.

[8] T. website, “http://www.tvplus.com/.”

[9] I.-N. application from Yahoo, “http://www.intonow.com/ci/soundprint.”

[10] M.-S. website, “http://media-sync.tv/.”

[11] C. website, “http://www.civolution.com.”

[12] N. Q. K. Duong, C. Howson, and Y. Legallais, “Fast second screen TV synchronization combining audio fingerprint technique and generalized cross correlation,” in Proc. IEEE International Conference on Consumer Electronics-Berlin (ICCE-Berlin), 2012, pp. 241–244.

[13] N. Q. K. Duong and F. Thudor, “Movie synchronization by audio landmark matching,” in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2013, pp. 3632–3636.

[14] C. V. Cotton and D. P. W. Ellis, “Audio fingerprinting to identify multiple videos of an event,” in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2010, pp. 2386–2389.

[15] Z. Rafii, B. Coover, and J. Han, “An audio fingerprinting system for live version identification using image processing techniques,” in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2014.

[16] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, “A review of audio fingerprinting,” Journal of VLSI Signal Processing, no. 3, 2005, pp. 271–284.

[17] G. Richard, S. Sundaram, and S. Narayanan, “An overview on perceptually motivated audio indexing and classification,” Proceedings of the IEEE, no. 9, 2013, pp. 1939–1954.

[18] A. Ramalingam and S. Krishnan, “Gaussian mixture modeling of short-time fourier transform features for audio fingerprinting,” IEEE Trans. on Information Forensics and Security, no. 4, 2006, pp. 457–463.

[19] E. Batlle, J. Masip, E. Guaus, and P. Cano, “Scalability issues in hmm-based audio fingerprinting,” in Proc. IEEE Int. Conference on Multimedia and Expo, 2004, pp. 735–738.

[20] V. Chandrasekhar, M. Sharifi, and D. A. Ross, “Survey and evaluation of audio fingerprinting schemes for mobile query-by-example applications,” in 12th International Society for Music Information Retrieval Conference (ISMIR), 2011, pp. 801–806.

[21] Y. Ke, D. Hoiem, and R. Sukthankar, “Computer vision for music identification,” in Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR), 2005, pp. 597–604.

[22] S. Baluja and M. Covell, “Audio fingerprinting: Combining computer vision and data stream processing,” in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2007.

[23] M. Ramona and G. Peeters, “Automatic alignment of audio occurrences: application to the verification and synchronization of audio fingerprinting annotation,” in Proc. DAFX, 2011, pp. 429–436.

[24] E. Dupraz and G. Richard, “Robust frequency-based audio fingerprinting,” in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2010, pp. 2091–2094.

[25] M. Moussallam and L. Daudet, “A general framework for dictionary based audio fingerprinting,” in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2014, pp. 3077–3081.

[26] J. S. Seo, M. Jin, S. Lee, D. Jang, S. Lee, and C. D. Yoo, “Audio fingerprinting based on normalized spectral subband centroids,” in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2005, pp. 213–216.

[27] K. Seyerlehner, M. Schedl, P. Knees, and R. Sonnleitner, “A refined block-level feature set for classification, similarity and tag prediction,” in Proc. Music Information Retrieval Evaluation eXchange (MIREX), 2011.

[28] X. Anguera, A. Garzon, and T. Adamek, “Mask: Robust local features for audio fingerprinting,” in Proc. IEEE International Conference on Multimedia and Expo (ICME), 2012, pp. 455–460.

[29] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied, “Robust sound modeling for song detection in broadcast audio,” in Proc. 112th Audio Engineering Society Convention (AES), 2002.

[30] J. Hao, T.-W. Lee, and T. J. Sejnowski, “Speech enhancement using gaussian scale mixture models,” IEEE Trans. on Audio Speech and Language Processing, no. 18, 2010, pp. 1127–1136.

[31] C. J. Burges, J. C. Platt, and S. Jana, “Distortion discriminant analysis for audio fingerprinting,” IEEE Trans. on Audio Speech and Language Processing, no. 3, 2003, pp. 165–174.

[32] J. Deng, W. Wan, X. Yu, and W. Yang, “Audio fingerprinting based on spectral energy structure and nmf,” in Proc. Int. Conf. on Communication Technology (ICCT), 2011, pp. 1103–1106.

[33] B. Logan, “Mel frequency cepstral coefficients for music modeling,” in Proc. Int. Sym. on Music Information Retrieval (ISMIR), 2002.

[34] A. Bagri, F. Thudor, A. Ozerov, and P. Hellier, “A scalable framework for joint clustering and synchronizing multi-camera videos,” in Proc. European Signal Processing Conference (EUSIPCO), 2013, pp. 1–5.

[35] W. 1, “http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html.”

[36] W. 2, “https://engineering.purdue.edu/malcolm/interval/1998-010/.”

[37] C. Yang, “Macs: music audio characteristic sequence indexing for similarity retrieval,” in Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2001, pp. 123–126.

[38] J. Ogle and D. Ellis, “Fingerprinting to identify repeated sound events in long-duration personal audio recordings,” in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2011, pp. 233–236.

[39] N. J. Bryan, P. Smaragdis, and G. J. Mysore, “Clustering and synchronizing multi-camera video via landmark cross-correlation,” in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2012, pp. 2389–2392.

[40] M. Ramona and G. Peeters, “Audio identification based on spectral modeling of bark-bands energy and synchronization through onset detection,” in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2011, pp. 477–480.

[41] E. Allamanche, J. Herre, and O. Hellmuth, “Content-based identification of audio material using mpeg-7 low level description,” in Proc. Int. Sym. on Music Information Retrieval (ISMIR), 2002.

[42] J. Herre, E. Allamanche, and O. Hellmuth, “Robust matching of audio signals using spectral flatness features,” in Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2001, pp. 127–130.

[43] D. Reynolds and R. Rose, “Robust text-independent speaker identification using gaussian mixture speaker models,” IEEE Trans. on Speech and Audio Processing, no. 1, 1995, pp. 72–83.

[44] L. Benaroya, F. Bimbot, and R. Gribonval, “Audio source separation with a single sensor,” IEEE Trans. on Audio, Speech and Language Processing, vol. 14, no. 1, 2006, pp. 191–199.

[45] L. R. Rabiner, “A tutorial on hmm and selected applications in speech recognition,” Proceeding of the IEEE, vol. 17, no. 2, 1989, pp. 257–286.

[46] D. D. Lee and H. S. Seung, “Learning the parts of objects with non-negative matrix factorization,” Nature, vol. 401, 1999, pp. 788–791.

[47] D. El Badawy, N. Q. K. Duong, and A. Ozerov, “On-the-fly audio source separation,” in IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP), 2014, pp. 1–6.

[48] N. Q. K. Duong, A. Ozerov, L. Chevallier, and J. Sirot, “An interactive audio source separation framework based on nonnegative matrix factorization,” in IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2014, pp. 1586–1590.

[49] C. Févotte, N. Bertin, and J.-L. Durrieu, “Nonnegative matrix factorization with the Itakura-Saito divergence. With application to music analysis,” Neural Computation, vol. 21, no. 3, 2009, pp. 793–830.

[50] N. Chen, H.-D. Xiao, and W. Wan, “Audio hash function based on non-negative matrix factorisation of mel-frequency cepstral coefficients,” IET Information Security, no. 1, 2011, pp. 19–25.

# User Experience Patterns from Scientific and Industry Knowledge

## An Inclusive Pattern Approach

Alexander G. Mirnig, Alexander Meschtscherjakov, Nicole Perterer, Alina Krischkowsky,
Daniela Wurhofer, Elke Beck, Arno Laminger, Manfred Tscheligi
Christian Doppler Laboratory for "Contextual Interfaces"
Center for Human-Computer Interaction, University of Salzburg
Salzburg, Austria
{firstname.lastname}@sbg.ac.at

*Abstract*— **Findings from scientific disciplines with close ties to the industry – such as Human-Computer Interaction (HCI) – can be useful for advancing both the scientific discipline itself as well as the associated industry. It is, therefore, an additional challenge to consolidate and convert the scientific knowledge gained into a format of which is applicable and understandable in practice in order to provide meaningful and usable tools for practitioners in their daily work routines. We used patterns to combine research results and industry know-how into solutions for distraction-related design problems in the automotive domain. In this paper, we present our pattern generation process that resulted in the creation of 16 patterns with input from scientists, as well as industrial stakeholders, in several key phases. Thereby, we discuss the advantages of patterns as a means to put scientific knowledge into practice. The contribution of this paper is a pattern generation and validation process and structure tailored towards combining scientific results and industry knowledge, as well as our pattern structure that resulted from this process.**

***Kewords-basics on patterns; design patterns; pattern identification and extraction; validate patterns.***

## I. INTRODUCTION

Patterns are a method to capture proven design solutions to reoccurring problems. They are a structured description of best practices and, as such, highly problem-oriented and reusable [1]. The use of patterns in design can improve the design process (both with regards to time and effort spent) to a considerable degree [2][3][4]. Patterns are also a recognized way of facilitating communication between different stakeholders. Since scientific research in Human-Computer Interaction (HCI) is closely interconnected with the industry, patterns could serve as a tool to communicate scientifically proven solutions to industry stakeholders. In our work, we aimed at generating patterns for HCI researchers and industry stakeholders based on scientific findings and transform them – by directly involving industry practitioners – into solutions that are relevant for and usable by these stakeholders. The underlying research question is how scientific findings may be translated into design patterns usable for practitioners in their daily routines and how such patterns may be generated by including scientific and industry stakeholders.

The outcome of our efforts was a pattern structure that incorporates scientific results and fits industry stakeholder needs, as well as a first set of 16 automotive User Experience (UX) design patterns. We refer to UX design patterns as patterns that tackle user experience issues in their core. In this paper, we present the final pattern structure, as well as the phases of the pattern generation process involving both scientists and industry stakeholders (We use the term 'generation' to delineate our approach from pattern finding methods, which usually focus only on actual implementations, and not theoretical or scientific works). We begin with an overview of current pattern literature in Section II. In Section III, we describe our pattern finding process via the concrete pattern structure example and its development. In section IV, we provide a summary of the overall process, together with a brief discussion on the limitations and potentials of our approach.

## II. RELATED WORK

In order to provide best practices and specific knowledge, the patterns approach has been well established in the domain of HCI [1]. Recently, specific domains in HCI, such as UX research, also deployed patterns to collect and structure their knowledge [3][4].

Köhne [6] (based on Quibeldey-Cirkel [7]) outlines specific steps for generating patterns. The process starts with discovering patterns, so-called *pattern mining*, by identifying whether a solution is valuable to solve a problem. The next step consists of *pattern writing*, where the problem solution is described in a defined structure. This is followed by *shepherding,* in which an expert provides support in improving the patterns content. Thereafter, a *writers workshop* is conducted. In such a workshop, a group of pattern authors discuss a pattern. Based on the feedback from the writers' workshop, the pattern author revises the pattern (*author review*). In a next step, the patterns are made public in a *pattern repository*, which is open to *anonymous peer review*. Finally, the pattern collection is published in a *pattern book* making the final patterns available for a large readership.

Similarly, Biel et al. [8] split the process of defining trust patterns for interaction design into four subtasks. The first task is *identifying a pattern* by analyzing the solutions used by designers. Second, the *pattern gets categorized* in order to make it reusable and accessible for designers. Third, the *pattern is described* following a specific structure. The fourth task is *evaluating the pattern* to prove its quality before it is introduced to a pattern library.

Aside from starting the pattern mining from designers' practical knowledge, patterns can also be harvested from scientific research findings. Martin et al. [9] use patterns to describe findings from ethnographies. For creating their patterns, they started by looking for specific examples in a particular domain in ethnographic studies and then tried to expand the observed phenomena to other domains (similar but different examples). Krischkowsky et al. [10] introduce a step-by-step guidance for HCI researchers for generating patterns from HCI study insights. According to them, the first step is giving novice and expert HCI researchers a *brief overview on the concept of patterns* and, more specifically, Contextual User Experience (CUX) patterns [4] (i.e., patterns to enhance user experience in a particular context). After this, the next step of the guidance concerns the *reflection and selection of relevant UX related results* from empirical studies conducted by the researchers. In a third step, HCI researchers *develop their own CUX patterns*, which are then internally *evaluated by researchers* following a checklist. In the last step, researchers give *feedback on the pattern generation process*.

Following a user centered patterns generation approach, we aimed at including industry designers within a specific domain (in our case automotive user interface design) in the patterns generation process in order to bring the target group as early as possible into the loop. Other approaches often miss to explicitly include industry stakeholders in the patterns generation process. In the following section, we outline and reflect on how we generated patterns. Further, we describe a seven-step approach that describes how we generated an initial set of automotive UX patterns from a scientific knowledge transfer workshop (step 1) to final pattern iteration (step 7). Based on a reflection of our work, we conclude with a novel patterns generation approach consisting of five phases. In addition, this paper presents an according pattern structure for distraction-related design problems in the automotive domain. Both, the patterns generation approach as well as the pattern structure for automotive UX patterns, are the main contributions of this paper.

## III. THE PATTERN GENERATION PROCESS

Within our research activities, the need for pattern guidance occurred within a national project focusing on contextual interface research in the automotive domain. In particular, the following section outlines the process of how we developed a pattern structure that provides insights, information, and guidance on how to design for a positive User Experience (UX) for the driver. This general aim (i.e., designing for a positive UX) was divided into more specific goals related to distinct UX factors (e.g., workload, fun, or trust). As the focus of our work was on the pattern generation process and the pattern structure, we decided to select one specific UX factor and improve the process and the structure by developing patterns for this factor. We chose to generate patterns for reducing workload that is caused by distraction, as this constitutes one of the most prevailing and severe problems in the automotive domain. In the next paragraphs, we outline each phase in the generation process in detail, reflecting on each step individually.

### A. Phase 1: Starting from scientific knowledge

In this first phase, we started from pure scientific knowledge about distraction-related design problems in the automotive domain to create an initial draft set of patterns. This seemed like a logical first step, since we wanted to go from the science to the practice. As we would learn later on, however, a slightly different approach would have been even better. This will be reflected in the conclusion chapter. The first phase can be segmented into four sub-steps, outlined in the following sections.

### B. Step 1: Scientific knowledge transfer workshop

Within the first step, a knowledge transfer workshop, organized by pattern experts and HCI researchers in the automotive domain, was conducted. Hereby, the main goal was to give experts in the automotive domain know-how on pattern generation. This know-how was provided by HCI pattern experts, in order to facilitate the development of an initial draft of patterns. The workshop lasted approximately four hours. Overall, six HCI researchers, all closely familiar with the automotive context, and two HCI pattern experts, who led the workshop, participated in this workshop.

In this initial knowledge transfer session, participants were introduced to patterns in general and the role of patterns in HCI in particular. This included aspects such as the usefulness of patterns as a tool for documentation, collection, communication, and representing knowledge [1]. They were also introduced into the distinguishing characteristics between patterns and guidelines. Thereafter, example patterns from other domains were presented (e.g., [11], [12], [13]). Subsequently, participants where shown the main goals for the development of patterns in the automotive domain (e.g., collect a number of UX related patterns, structured guidance on how to design for a good UX regarding advanced in-car systems). Thereafter, a presentation of the initial pattern structure was given, based on the CUX patterns approach [4]. This approach has already proved its value for collecting and structuring knowledge on UX [3]. The CUX pattern approach was chosen, as it explicitly considers the relation of UX and contextual aspects. In order to provide a better understanding of the CUX pattern approach, an exemplary CUX pattern reflecting on 'increased workload by font size' was shown to the participants. At the end of the workshop, participants were introduced to the entire, initially defined, pattern structure for UX patterns in the automotive domain (see Table 1, not-underlined parts).

### C. Step 2: An Initial set of patterns

After the workshop, the HCI researchers (and pattern experts) received the task to create two patterns within the next 10 days based on literature and/or their own research activities. They received a template with the pattern structure as a guideline for creating a first set of patterns related to a car driver's workload caused by distraction. Furthermore, the HCI researchers were also encouraged to give individual feedback to the pattern experts about issues and problems concerning the generation process, as well as the given structure (i.e., CUX pattern structure template). More details about the identified issues and problems are outlined in the next section.

Within this first generation phase, 16 patterns focusing on workload caused by distraction were developed (i.e., two patterns per person). All patterns were derived on the basis of scientific literature (e.g., research articles or book chapters referenced in the pattern). Also, two pattern experts were involved in this process and generated two patterns each. The generated patterns were about one page each and exclusively dealt with design solutions (e.g., voice interaction, multimodal interfaces, or information presentations) addressing the problem of increased workload due to distraction.

### D. Step 3: First iteration based on participants feedback: identified problems in the generation process, resulted in a refined pattern structure

The first round of pattern generation led to the identification of several issues with the initial pattern structure. During creating their patterns, the HCI researchers listed and forwarded encountered problems to the pattern experts. In a second workshop, the HCI researchers discussed their experiences with the provided pattern structure and the pattern creation process with the pattern experts and collected further problematic issues. The pattern experts then used the feedback for improving the pattern section structure and the related instruction for how to generate patterns based on the provided structure.

The refined pattern structure, as the outcome of the third step, is presented in Table 1. Changes to the section name and instruction are marked with an underline, parts not underlined are those from steps 1 and 2. The proposed pattern structure consists of nine parts: *name* (a description of the solution of the pattern), *UX factor* (the addressed automotive user experience factor), *problem statement* (a very short description of the problem which should be solved by the pattern), *forces* (a more detailed explanation of the problem), *context* (the application context of the pattern), *solution* (the proposed solution of the particular pattern), *examples* (concrete examples of best practices), *keywords* (phrases related to the pattern), and *sources* (origin of the pattern).

Most of the issues brought forward were concerned with what makes the pattern a high-quality pattern and what supports the comprehensibility of the pattern. More specifically, the HCI researchers had difficulties with *achieving the aim of a pattern to provide best practices*. The HCI researchers experienced it as challenging to judge if the provided solutions are the "gold standard". They also felt uneasy if "old" literature can serve as basis for pattern creation. Therefore, it would be more realistic to speak of providing existing knowledge to the best of one's judgment, i.e., preferably using the newest knowledge for underpinning a specific pattern and using as many potential evidences (studies, norms, etc.) as possible. Our patterns suggest solutions for specific UX demands in the car area based on existing knowledge (e.g., studies, best practices).

Another difficulty is related to *deciding on the abstraction level of a pattern.* The HCI researchers were unsure whether they should create very general patterns (global patterns) versus very specific patterns (sub-patterns, local patterns). They finally agreed on providing patterns that are abstract enough to make generalizations, while providing practical solutions at the same time. Thus, both elements (i.e., generalization as well as a concrete example) should be provided.

*Identifying the stakeholders of the patterns* was also an issue. It was unclear to the HCI researchers whom they should address with the patterns; whether the future users of the created patterns are designers (expert or novice), domain-specific users (e.g., industrial manufacturers), researchers, or developers.

TABLE I. INITIAL AND REFINED PATTERN STRUCTURE (ITERATION CHANGES UNDERLINED)

| **Instructions on Each Pattern Section** | | |
|---|---|---|
| **#** | ***Section Name*** | ***Instruction on Each Section*** |
| 1 | Name | *The name of the pattern should shortly describe the solution suggested by the pattern (2-3 words would be best).* |
| 2 | UX Factor | *List the UX factor(s) addressed by the pattern.* |
| 3 | Problem Statement | *As short as possible - the best would be to describe the problem in one sentence.* |
| 4 | Forces | *Should be a detailed description and further explanation of the problem.* |
| 5 | Context | *In general, our patterns should focus on the driver. Describe the detailed context in which the pattern can be applied in this section.* |
| 6 | Solution | *1) Can range from rather general suggestions to very concrete suggestions for a specific application area (e.g., " Presenting High-Priority Warnings").*<br>*2) A successful solution is based on existing knowledge (e.g., state of the art solutions, empirical studies, guidelines, etc.).*<br>*3) More than one solution is no problem but even better than only one.*<br>*4) There can also be a general solution and more specific "sub-solutions".* |
| 7 | Examples | *Concrete examples underpinned by pictures, standard values (e.g., angle, size) etc. Examples should not provide a solution (this is done in the solution part) but rather underpin and visualize the solution presented above.* |
| 8 | Keywords | *Describe main topics addressed by the pattern in order to enable structured search.* |
| 9 | Sources | *Origin of the pattern (cf. the different ways to generate patterns)* |

The HCI researchers also experienced difficulties in *creating a pattern name;* should the pattern name be formulated as solution or problem? It was eventually decided to opt for a solution orientation of the pattern name and modified the pattern instruction accordingly. Moreover, *using technical terms in the pattern name* lead to comprehensibility problems among the HCI researchers. A pattern needs to be easy to understand and quickly assessed. Consequently, very specific technical terms should not be used in the pattern name and, if they occur in the description of the pattern, they need to be explained.

Furthermore, the first round of pattern generation revealed that the HCI researchers deployed *different ways to generate their patterns*, which are based on existing state of the art knowledge/experience in the field, on own empirical studies, on literature (desktop research of empirical studies), as well as on existing structured knowledge. Therefore, the section on sources (#9) was supplemented with different ways to generate patterns.

### E. Step 4: Participants iterate patterns based on refined structure

Finally, the HCI researchers task was to iterate their initially created patterns based on the refined pattern structure. Each researcher transformed the existing pattern he/she originally constructed into the new pattern structure. Parts were reformulated, where necessary, and other parts were added.

### F. Step 5: Industry stakeholder pattern structure evaluation workshop

To further iterate and finalize the pattern structure, we involved the industry stakeholders in a workshop with the aim of evaluating the current pattern structure on the basis of two representative patterns. The workshop was conducted at our facility with five participants (one female and four male) from our industrial partner from the automotive domain. The participants' age ranged from 20 to 45 years, job experience from 7 months to 20 years. Their professional background was software developers, engineers, and designers. After a 10-minute general introduction to patterns and our pattern structure, participants received printouts of one of our automotive UX patterns with the instruction to read through it attentively (duration: 10 minutes). After that, they had to fill in a questionnaire regarding the quality and understandability of the pattern. Participants then received another pattern printout and were again given 10 minutes to read it thoroughly. This was done to ensure that the participants had a means of comparison and also to reduce bias regarding the quality (or the lack thereof) of the pattern structure based on only one pattern. After these preparations, the main part of the workshop, a discussion session (total duration: 1.5 hours), began. The moderated discussion was audio recorded and later transcribed for further analysis. During the course of the discussion, participants could voice concerns they had encountered when reading the individual patterns, together with suggestions for improvements to the pattern structure, as well as the existing automotive UX patterns in particular.

In the following section, we will outline the most important outcomes of the workshop, in reference to the iterated structure shown in Table 1. Participants were confused by the separation of *problem* and *forces*, stating that they did not understand why those were two separate categories and that they found the term 'forces' itself difficult to understand. Furthermore, participants found that they had to read quite far into the patterns before they knew what the patterns were exactly about. Generally, the participants desired an "abstract" for each pattern, containing scope, context, and possibly an outlook on the solution in a very compact format. In addition, the pattern should be re-structured, so that the most important information (at the very least: *name*, *keywords*, and *problem*) is at the very beginning of the pattern. Or, as one participant put it, "*If using a pattern collection is more cumbersome than using Google and produces lesser results, then there is little reason to use that pattern collection.*"

The writing style and vocabulary used in both patterns was perceived as very unusual by the participants and more "scientific" than what they were used to. More specifically, they were not used to citing sources for every claim and the rather high number of technical terms used in each pattern. While they found the scientific writing style an overall pleasing quality that should be kept, they suggested a minimal citation style (numbers only, full references only at the very end of the pattern collection). The issues identified in the workshop were then further discussed and transformed into concrete instructions for another pattern structure overhaul.

### G. Step 6: Final pattern structure iteration

Based on the feedback gained from this workshop, the pattern structure underwent a final iteration, which would then become the basis for all further patterns (see Table 2). Similar to the pattern structure shown in Table 1, the final pattern structure consists of nine elements. The *name* of the pattern should again focus on the provided solution. The *intent* should include the main category of the pattern, a short problem statement, and briefly outline the context in which the pattern should be used. It replaces the problem statement (3) and the context (6) of the initial structure presented in Table 1. The new element *topics* is a structured list of keywords describing the problem scope. The element *problem* replaced the forces (4) section. The new element *scenario* gives a detailed description of the problem in a scenario like style. The *solution* again describes the solution to the problem. Within the final structure, we provide a structured approach how to present the solution. *Examples*, as before, should show best practices of the pattern. *Keywords*, again, should help to find related patterns. Finally, *sources* link to the origin of the pattern. The element UX factor (2) from the initial pattern structure was omitted at all.

The new structure focuses on informing the reader as concisely as possible about whether the pattern is relevant for them. *Name*, *intent*, and *topics* are standardized and kept brief so that only a minimal amount of time is needed to read and process them. *Context* and *forces* are combined into the new Scenario-category, since the stakeholders had a hard time differentiating between them and found the distinction to be inconsequential in practice.

TABLE II. FINAL PATTERN STRUCTURE

| Instructions on Each Pattern Section | | |
|---|---|---|
| **#** | ***Section Name*** | ***Instruction on Each Section*** |
| 1 | Name | *The name of the pattern should shortly describe the solution suggested by the pattern (2-3 words would be best).* |
| 2 | Intent | *Short statement in three parts:*<br>*a) Main category of pattern (e.g., visual information presentation)*<br>*b) Short issue/problem statement (e.g., effective display position)*<br>*c) Short context preview (e.g., while driving)* |
| 3 | Topics | *Max. 8 Keywords describing problem scope: 1) who is affected (driver, co-driver, etc); 2) which modalities are addressed (visual, haptic, acoustic)* |
| 4 | Problem | *Should be a detailed description and further explanation of the problem.* |
| 5 | Scenario | *Provide a detailed example of a case, in which the problem occurs* |
| 6 | Solution | • *First, provide a general (either high level or one that is applicable in the most cases) solution.*<br>• *Then provide alternative solutions, together with delineating criteria to determine, when such alternative solutions apply.*<br>• *Whenever possible, reuse (modified) figures, illustrations, etc. from other patterns, for a more consistent style and easier combination of pattern solutions.*<br>• *A successful solution is based on existing knowledge (e.g., state of the art solutions, empirical studies, guidelines, etc).*<br>• *More than one solution is no problem but even better than only one.* |
| 7 | *Examples* | *Concrete examples underpinned by pictures, standard values (e.g., angle, size) etc. Examples should not provide a solution (this is done in the solution part) but rather underpin and visualize the solution presented above.* |
| 8 | Keywords | *Describe main topics addressed by the pattern and related patterns in order to enable structured search.* |
| 9 | Sources | *Origin of the pattern, related literature, related patterns (if they are not part of the same pattern collection), norms and guidelines, other references. Citations format: Numbers and endnotes, to distract the reader as little as possible.* |

### *H. Step 7: Final pattern iteration*

The entire set of 16 patterns was then revised, based on the above-mentioned structure (see Table 2). *Scenario*, *Solution* and *Examples* specifically underwent an adaptation according to the stakeholders' requirements. If possible, solutions were also represented graphically or illustrations from cited publications were added. Concrete examples (state of the art) from recent production vehicles illustrated, if appropriate, the examples section. In general, care was taken to present the information in every pattern in a compact form, easily comprehensible and practicable. This final iteration was completed as a team effort by two scientists.

### *I. Validating the patterns*

For the final validation of the iterated pattern set, we conducted a second workshop at our facility with seven participants (4 employees from our industrial partner and 3 researchers; 6 male and 1 female). Age ranged from 21 to 48 years, job experience from one month to eight years. Regarding their professional background, they were software developers, engineers, designers, and HCI experts. Part of the participants from the first workshop also participated in the second one. To have a good mix of informed and fresh views, we involved two stakeholders who had already participated in the previous workshop, and two who were completely new to the topic. The overall goal of the second workshop was to assess the quality of the first UX pattern set, as well as to iterate the pattern set based on the industry stakeholders' feedback.

In this workshop, the full iterated pattern set was presented to the participants and evaluated on a peer-pattern basis. Each of the 16 existing patterns was rated individually by each participant. To avoid serial positions effects and similar forms of bias, the patterns were presented to participants in different orders. After a 10-minute general introduction to patterns and explanations of the iterated UX pattern structure from the first workshop, a researcher explained the purpose and the agenda of the one-day workshop to the participants.

Then, each participant read thoroughly and rated each pattern based on the following slightly modified quality criteria checklist [14] that consists of four quality criteria ($c_1$, $c_2$, $c_3$, $c_4$). The first quality criterion ($c_1$) states that all parts of a pattern description should be reasonable to the pattern users. This means they should have a meaningful *name*, a clear formulated *problem* statement, and enough background information for the provided *scenario*, concrete *solutions*, as well as give plausible *examples*. The second quality criterion ($c_2$) addresses five aspects: (1) *completeness,* i.e., necessary information is given in the pattern; (2) *clarity of the language,* i.e., the style of the pattern is well-readable; (3) *problem-centricity,* i.e., the scenario, solutions, and examples are coherent and clearly related to the problem description; (4) *good balance* between concreteness and abstractness; and (5) *helpfulness,* i.e., the presented patterns support stakeholders to develop better interactive systems. The third criterion ($c_3$) requested an *individual overall assessment of* *the patterns* from a more general perspective. $C_4$ states that the whole collection of patterns captures *relevant knowledge* about User Experience and provides a *suitable common basis* for designers, developers, and researchers. $C_4$ applies to the whole collection and not to individual patterns. It was, therefore, excluded from the questionnaire and instead discussed in plenum at the end of the workshop for a qualitative, *overall assessment* of the pattern set quality and applicability.

After the rating of the patterns, a moderated discussion session took place. During the discussion, participants could voice all concerns they had encountered when reading the 16 existing patterns, together with suggestions for future improvements to the existing UX patterns. In order to trigger the discussion, two questions of criterion ($c_4$) from the quality framework were asked to the participants: Do you think that the presented patterns support the communication of designers, developers and researches by providing common basis? Do you think the presented patterns capture relevant knowledge about user experience? The discussion session was audio-recorded and later on transcribed for further analysis.

Due to the low number of participants, the questionnaire results were analyzed in descriptive form. The results of the first quality criterion ($c_1$), rated on a scale from 1 (absolutely agree) to 5 (do not agree at all), show that the pattern set had a meaningful name (M=1.86, SD=1.08), a clear stated problem (M=1.48, SD=0.80), and enough background information of the stated scenario (M=2.03, SD=1.02). The two last categories of $c_1$, i.e., the solution (M=2.69, SD= 1.15) and the examples (M=2.60, SD=1.16), were rated as neutral.

The questionnaire responses of the second quality criterion ($c_2$) indicated a very positive overall picture with mean values all in a positive spectrum (lowest was 1.55) and the most negative responses being neutral ones (2.78). The responses were also rated on a scale from 1 (absolutely agree) to 5 (do not agree at all). Lowest mean values were identified clarity of the language used in the pattern (M= 1.55, SD=0.73) and the problem-centricity (M=2.16, SD=0.88).

In total, the participants perceived only one pattern as implausible ($c_3$). As mentioned by the participants during the discussion ($c_4$), the presented patterns support the communication of designers, developers and researchers, provide a common basis, and capture relevant knowledge about user experience.

One recurring problem, which had sporadically been voiced during the previous workshop as well, was the relevance of the problem statements in the discussion. The participants felt that the problems stated in some patterns were only partly relevant for them and while they appreciated the solutions, they would often have desired to be part of the problem statement beforehand. This led us to modify our pattern generation approach to involve the industry stakeholders already during the very first step in the pattern generation process. The list of design problems that patterns are then generated for should, together with a rating regarding relevance and importance, come from the industry stakeholders themselves. Ideally, this should happen with guidance and assistance from researchers. Contextual inquiries or brainstorming with subsequent problem rating sessions with the industry stakeholders are both suitable methods to achieve this.

## IV. Conclusion, Limitations and Future Work

In this paper, we have described a seven-step approach how to generate (automotive) UX patterns. It started with a scientific knowledge transfer workshop (step 1), which led to an initial set of patterns (step 2). A first iteration based on participants' feedback and the identification of problems in the generation process resulted in a refined pattern structure (step 3). An iteration of the patterns led to a refined pattern structure (step 4), with which we conducted a pattern structure evaluation workshop with industry stakeholders (step 5). Another pattern structure iteration (step 6) led to a final pattern iteration (step 7).

By involving industry practitioners directly in the pattern generation process, we were able to translate scientifically proven results into proven solutions for industry stakeholders. As mentioned earlier, it might have been beneficial to include industry stakeholders before the first patterns were generated. We experienced that not all of the patterns we initially produced were urgent problems for our practitioners from the industry. They mentioned that an approach where they could identify problems of high priority to them would be better. Nonetheless, the insights we gained have resulted in a pattern structure suitable for industry stakeholders' needs in the automotive domain. The structure focuses on clarity and brevity and should, with slight modifications, be adaptable for other industry domains as well. Furthermore, we have documented our pattern generation process, together with both scientists and industry stakeholders. A high level overview of the process can be seen in Figure 1. It includes a first phase with industry focus, in which industry problems are identified, and where patterns

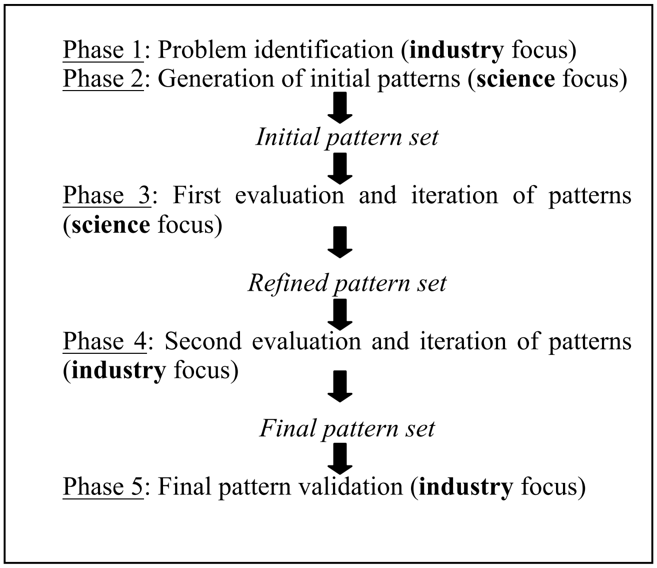

Phase 1: Problem identification (**industry** focus)
Phase 2: Generation of initial patterns (**science** focus)
*Initial pattern set*
Phase 3: First evaluation and iteration of patterns (**science** focus)
*Refined pattern set*
Phase 4: Second evaluation and iteration of patterns (**industry** focus)
*Final pattern set*
Phase 5: Final pattern validation (**industry** focus)

Figure 1. Final inclusive patterns generation process.

might be a beneficial way of helping to solve these problems. In phase 2, we suggest generating an initial set of patterns. Phase 3 includes evaluation and iteration through a scientific lens. Phase 4 includes evaluation and iteration with a focus from industry and, in phase 5, patterns are validated.

Apart from the patterns generation process, this paper presents a structure for automotive user experience patterns. It consists of nine elements (*name, intent, topics, problem*, *scenario, solution, examples, keywords*, and *sources)*, which proved to be a useful way to structure UX patterns in the automotive domain.

The approach described in this paper is a departure from the common practice of documenting already working solutions, to a way to convert (proven) scientific results to working problem solutions. The evaluation of the described approach was based mainly on feedback of practitioners from the HCI car domain. Furthermore, we did not compare the quality of our patterns' problem solutions to those of other HCI patterns in our research. While the positive assessment of the overall process and its results (the patterns) provides a positive outlook, further evaluations (and possible iterations) are certainly needed to fully validate it as a reusable standard procedure in the community.

Overall, the pattern generation process and structure we gained will be used for generating additional UX patterns for the automotive domain. More specifically, we intend to also cover the factors *perceived safety* and *joy of use* and generate patterns for these. We have already begun the generation process by identifying common design problems related to these factors in a workshop together with the industry stakeholders. In the future, we intend to implement the full pattern collection as an online database based on the pattern framework proposed in [15]. We will continue using our inclusive pattern generation process to translate scientifically proven results into proven solutions for industry stakeholders and encourage others to employ and further refine our proposed method.

## ACKNOWLEDGMENTS

The financial support by the Austrian Federal Ministry of Science, Research and Economy and the National Foundation for Research, Technology and Development and AUDIO MOBIL Elektronik GmbH is gratefully acknowledged (Christian Doppler Laboratory for "Contextual Interfaces").

## REFERENCES

[1] C. Alexander, "The Timeless Way of Building," New York: Oxford University Press, 1979.

[2] A. Dearden and J. Finlay, "Pattern Languages in HCI: A Critical Review," HCI, Volume 21, 2006, pp. 49-102.

[3] M. Obrist, D. Wurhofer, E.Beck, A. Karahasanovic, and M. Tscheligi, "User Experience (UX) Patterns for Audio-Visual Networked Applications: Inspirations for Design," Proc. Nordic Conference on Human-Computer Interaction (NordiCHI 2010), Reykjavik, Iceland, 2010, pp. 343-352.

[4] M. Obrist, D. Wurhofer, E. Beck, and M. Tscheligi, "CUX Patterns Approach: Towards Contextual User Experience Patterns," Proceedings of the Second International Conferences on Pervasive Patterns and Applications (PATTERNS 2010), Lisbon, Portugal, 2010, pp. 60-65, retrieved: January 2015.

[5] T. Erickson, "Lingua francas for design: sacred places and pattern languages," Proceedings of the 3rd conference on Designing interactive systems (DIS '00), ACM, New York, NY, USA, 2000, pp. 357-368.

[6] S. Köhne, "A Didactical Aproach towards Blended Learning: Conception and Application of Educational Patterns", Dissertation, Universty of Hohenheim, Hohenheim, Germany, 2005 [Orig. title: Didaktischer Ansatz für das Blended Learning: Konzeption und Anwendung von Educational Patterns].

[7] K. Quibeldey-Cirkel. Design Patterns in Object Oriented Software Engineering. Springer, 1999 [Orig. title: Entwurfsmuster: Design Patterns in der objektorientierten Softwaretechnik].

[8] B. Biel, T. Grill, and V. Gruhn, "Patterns of trust in ubiquitous environments," Proc. 6th International Conference on Advances in Mobile Computing and Multimedia (MoMM 2008), Gabriele Kotsis, David Taniar, Eric Pardede, and Ismail Khalil (Eds.). ACM, New York, NY, USA, 2008, pp. 391-396. DOI=10.1145/1497185.1497269, retrieved: January 2015.

[9] D. Martin, T. Rodden, M. Rouncefield, I.Sommerville, and S. Viller, "Finding Patterns in the Fieldwork," Proceedings of the Seventh European Conference on Computer-Supported Cooperative Work, Bonn, Germany, 2001, pp. 39-58.

[10] A. Krischkowsky, D. Wurhofer, N. Perterer, and M. Tscheligi, "Developing Patterns Step-by-Step: A Pattern Generation Guidance for HCI Researchers," The Fifth International Conferences on Pervasive Patterns and Applications (PATTERNS 2013) IARIA, Valencia, Spain, 2013, pp. 66-72, retrieved: January 2015.

[11] M. van Welie and B. Klaassen, "Evaluation Museum Websites using Design Patterns" Internal Report, IR-IMSE-001, Faculty of Computer Science, Vrije Universiteit Amsterdam, The Netherlands, 2004.

[12] J. Borchers, "A pattern approach to interaction design", Wiley series in software design patterns, John Wiley & Sons, NY, USA, 2001.

[13] J. Tidwell, J., "Common Ground: A Pattern Language for Human-Computer Interface Design," 1999, http://www.mit.edu/~jtidwell/interaction_patterns.html, retrieved: January, 2015.

[14] D. Wurhofer, M. Obrist, E. Beck, and M. Tscheligi, "A Quality Criteria Framework for Pattern Validation," International Journal On Advances in Software, Volume 3, Issue 1-2, 2010, pp. 252-264.

[15] A. Mirnig and M. Tscheligi, "Building a General Pattern Framework via Set Theory: Towards a Universal Pattern Approach," The Sixth International Conference on Pervasive Patterns and Applications (PATTERNS 2014) IARIA, Venice, Italy, May 2014, pp. 8-11, retrieved: January 2015.

# A Novel Pattern Matching Approach for Fingerprint-based Authentication

Moudhi AL-Jamea, Tanver Athar, Costas S. Iliopoulos
Solon P. Pissis, M. Sohel Rahman

Department of Informatics
King's College London
London, UK
e-mail:{mudhi.aljamea,tanver.athar,costas.iliopoulos,solon.pissis,sohel.rahman}@kcl.ac.uk

*Abstract*—**In Biometrics, fingerprint is still the most reliable and used technique to identify individuals. This paper proposes a new fingerprint matching technique, which matches the fingerprint information by using algorithms for approximate circular string matching. The minutiae information is transformed into string information by using a series of circles, which intercepts the minutiae and that information into a string. This string fingerprint information is then matched against a database by using approximate string matching techniques.**

***Keywords–Biometrics, fingerprints, matching, verification, orientation field.***

## I. Introduction

Recently, the need for automatic person identification has increased more and more in our daily activities, in general, and in the world of business and industry, in particular. To this end, the use of biometrics has become ubiquitous [1], [2]. Biometrics refers to metrics related to human characteristics and traits. Since biometric identifiers are unique to individuals, automatic person identification systems based on biometrics offer more reliable means of identification than the classical knowledge-based schemes such as password and personal identification number (PIN) and token based schemes such as magnetic card, passport and driving license. Among all the various forms of biometrics including face, hand and finger geometry, eye, voice and speech and fingerprint [3], the fingerprint-based identification is the most reliable and popular personal identification method.

Fingerprints offer an infallible means of personal identification and has been used for person authentication since long. Possibly, the earliest cataloguing of fingerprints dates back to 1891 when the fingerprints of criminals were collected in Argentina [4]. Now, it is used not only by police for law enforcement, but also in commercial applications, such as access control and financial transactions; and in recent times in mobile phones and computers.

In terms of applications, there are two kinds of fingerprint recognition systems, namely, verification and identification. In the former, the input is a query fingerprint with an identity (ID) and the system verifies whether the ID is consistent with the fingerprint and then outputs either a positive or a negative answer depending on the result. On the contrary, in identification, the input is only a query fingerprint and the system computes a list of fingerprints from the database that resemble the query fingerprint. Therefore, the output is a short (and possibly empty) list of fingerprints.

The majority of research in recent times has focused only on the fingerprint authentication, but not on the rotation of fingerprints. The majority of the state-of-the-art assumes that the fingerprint is aligned in the same direction as that of the stored fingerprint images. This is an important aspect of fingerprint matching, which various techniques have ignored, and only very few, in the literature [5], have considered. With the introduction of fingerprint detection in mobile devices, the rotation aspect of the fingerprint detection is an important area of research.

### A. Our Contribution

In this paper, we revisit the fingerprint recognition problem that is the basis of any fingerprint based identification system. Despite a plethora of fingerprint matching algorithms there is still room for improvement [6]. Interestingly enough, in spite of similarities between the two domains, there has not been much work at the intersection of algorithms on strings and the study of fingerprint recognition. To the best of our knowledge, here we make a first attempt to employ string matching techniques to solve fingerprint recognition problem efficiently and accurately. Converting the fingerprint image into string results in a small string. Matching this string against other fingerprint images stored as strings can be done in time linear with respect to the total length of the strings. In our approach, we have formulated an algorithm to detect and verify a fingerprint regardless of its position and rotation in wide scanning surface area in a simple and efficient way.

### B. Road Map

The organization of the rest of this paper is as follows. In Section II, we present some background related to fingerprints. Section III presents a very brief literature review. We present our approach in Section V after discussing some preliminaries in Section IV. Finally, we briefly conclude in Section VI.

## II. Background

Fingerprint pattern can be simply defined as the combination of ridges and grooves on the surface of a fingertip. The inside surfaces of the fingers contain minute ridges of skin with furrows between each ridge. The ridges and furrows run in parallel lines and curves to each other forming complicated patterns. The basic fingerprint (FP) patterns are whorl, loop, and arch [7]. However, the most common and widely used classification method is based on Henry's classification [8] [9] which contain 8 classes: Plain Arch, Tented Arch, Left Slant

Loop, Right Slant Loop, Plain Whorl, Double-Loop Whorl Central-Pocket Whorl, and Accidental Whorl (see Figure 1).

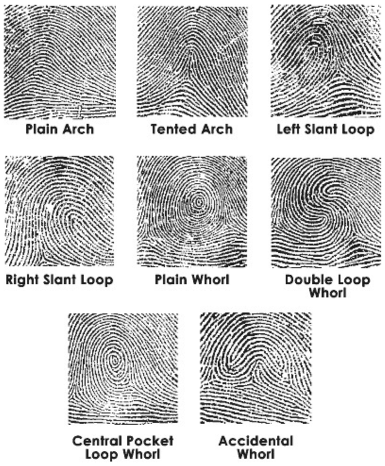


Figure 1. Classification of fingerprint patterns

Each fingerprint is highly stable and unique. This uniqueness is determined by global features like valleys and ridges, and by local features like ridge endings and ridge bifurcations, which are called minutiae. According to recent studies, the probability of two individuals having the same fingerprint is less than one in a billion [10].

Fingerprinting has been used historically to identify individuals using the so-called ink-technique [11], where the fingers are dabbed with ink to get an imprint on paper cards which are then scanned to produce the digital image. In this off-line fingerprint acquisition technique, the fingerprints are matched by using the scanned images produced above. This method is still very important and popular especially in the forensics field, where fingerprints are captured from crime scenes. However, this type of off-line methods are not feasible for biometric systems [12]. The other approach is of-course to scan and match fingerprints in real time.

## III. Related works

Fingerprint recognition has been the centre of studies for a long time and as a result, many algorithms/approaches have been proposed to improve the accuracy and performance of fingerprint recognition systems. In the fingerprint recognition literature, a large body of work has been done based on the minutiae of fingerprints [13]–[16]. These works consider various issues including, but not limited to, compensating for some of the non-linear deformations and real word distortion in the fingerprint image. As a trade off with accuracy, the issue of memory and processor intensive computation has also been discussed in some of these works.

The minutiae-based matching are the most popular approach due to the popular belief that minutiae are the most discriminating and reliable features [17]. However, this approach faces some serious challenges related to the large distortions caused by matching fingerprints with different rotation (see Figure 2). As a result, researchers have also used other features for fingerprint matching. For example, the algorithm in [18] works on a sequence of points in the angle-curvature domain after transforming the fingerprint images into these points. A filter-based algorithm using a bank of Gabor filters to capture both local and global details in a fingerprint as a compact fixed-length finger code is presented in [19]. The combinations of different kind of features have also been studied in the literature [20], [21]. There exist various other works in the literature proposing different techniques for fingerprint detection based on different feature sets of fingerprints [22], [23], [15]. Due to brevity we do not discuss these works in this paper. Interested readers are referred to a very recent review by Unar et al. [2] and references therein.

Note that, in addition to a large body of scientific literature, a number of commercial and propitiatory systems are also in existence. In the related industry, such systems are popularly termed as Automatic Fingerprint Identification System (AFIS). One issue with the AFIS available in the market relates to the sensor used to capture the fingerprint image. In particular, the unrealistic assumption of the most biometric systems that the fingerprint images to be compared are obtained using the same sensor, restricts their ability to match or compare biometric data originating from different sensors [24]. Another major challenge of commercially available AFISs is to increase the speed of the matching process without substantially compromising accuracy in the application context of identification, especially, when the database is large [6]. This is why the quest for even better fingerprint recognition algorithms is still on particularly in the high-performance computing context [6].

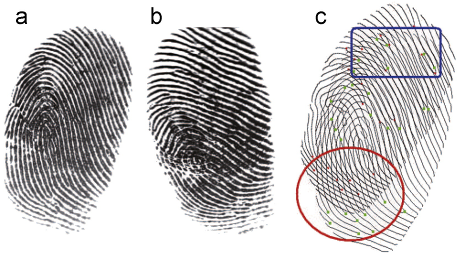


Figure 2. An example of large distortion from FVC2004 DB1 [25]

## IV. Preliminaries

In order to provide an overview of our results and algorithms, we begin with a few definitions. We think of a string $x$ of length $n$ as an array $x[0..n-1]$, where every $x[i]$, $0 \leq i < n$, is a letter drawn from some fixed alphabet $\Sigma$ of size $\sigma = |\Sigma|$. The empty string of length 0 is denoted by $\varepsilon$. A string $x$ is a factor of a string $y$ if there exist two strings $u$ and $v$, such that $y = uxv$. Let the strings $x$,$y$,$u$, and $v$, such that $y = uxv$. If $u = \varepsilon$, then $x$ is a prefix of $y$. If $v = \varepsilon$, then $x$ is a suffix of $y$. In the string $x = aceedf$, $ac$ is a prefix, $ee$ is a factor and $df$ is suffix.

Let $x$ be a non-empty string of length $n$ and $y$ be a string. We say that there exists an occurrence of $x$ in $y$, or, more simply, that $x$ occurs in $y$, when $x$ is a factor of $y$. Every occurrence of $x$ can be characterised by a position in $y$. Thus, we say that $x$ occurs at the starting position $i$ in $y$ when $y[i..i+n-1] = x$.

A circular string of length $n$ can be viewed as a traditional linear string, which has the left- and right-most symbols wrapped around and stuck together in some way [26]. Under this notion, the same circular string can be seen as $n$ different linear strings, which would all be considered equivalent. Given a string $x$ of length $n$, we denote by $x^i = x[i..n-1]x[0..i-1]$, $0 < i < n$, the $i$-th rotation of $x$ and $x^0 = x$. Consider, for instance, the string $x = x^0 = abababbc$; this string has the following rotations: $x^1 = bababbca$, $x^2 = ababbcab$, $x^3 = babbcaba$, $x^4 = abbcabab$, $x^5 = bbcababa$, $x^6 = bcababab$, $x^7 = cabababb$. Here we consider the problem of finding occurrences of a pattern $x$ of length $m$ with circular structure in a text $t$ of length $n$ with linear structure. This is the problem of circular string matching.

The problem of exact circular string matching has been considered in [27], where an $O(n)$-time algorithm was presented. The approach presented in [27] consists of preprocessing $x$ by constructing a suffix automaton of the string $xx$, by noting that every rotation of $x$ is a factor of $xx$. Then, by feeding $t$ into the automaton, the lengths of the longest factors of $xx$ occurring in $t$ can be found by the links followed in the automaton in time $\mathcal{O}(n)$. In [28], an average-case optimal algorithm for exact circular string matching was presented and it was also shown that the average-case lower bound for single string matching of $\Omega(n \log_\sigma m/m)$ also holds for circular string matching. Very recently, in [29], the authors presented two fast average-case algorithms based on word-level parallelism. The first algorithm requires average-case time $O(n \log_\sigma m/w)$, where $w$ is the number of bits in the computer word. The second one is based on a mixture of word-level parallelism and $q$-grams. The authors showed that with the addition of $q$-grams, and by setting $q = \Theta(\log_\sigma m)$, an average-case optimal time of $O(n \log_\sigma m/m)$ is achieved.

The Approximate Circular String Matching via Filtering (ACSMF) algorithm [30] is used here in order to identify the orientation of the fingerprint. The basic principle of algorithm ACSMF is the partitioning scheme that splits the concatenation of the circular pattern string into $2d + 4$ fragments, where $d$ is the maximum edit distance allowed. The Aho-Corasick automaton [31] is then used to search for the fragments against the text. Once a fragment is identified, the fragment is extended on both left and right directions to determine a valid match.

*Theorem 1 ( [30]):* Given a pattern $x$ of length $m$ drawn from alphabet $\Sigma$, $\sigma = |\Sigma|$, a text $t$ of length $n > m$ drawn from $\Sigma$, and an integer threshold $d = \mathcal{O}(m/\log_\sigma m)$, algorithm ACSMF requires average-case time $\mathcal{O}(n)$.

## V. Our Approach

In this section we present our main contribution, i.e., a novel pattern matching approach to solve the fingerprint recognition problem. As has been discussed above, two main difficulties related to the fingerprint recognition problem are lack of a fixed orientation and the presence of errors in the scanned image due to various reasons (e.g., the presence of dust, oil and other impurities on the finger and on the scanning surface). We therefore employ a two-stage algorithm. We start with a brief overview of our algorithm below.

### A. Algorithmic Overview

Our algorithm consists of two distinct stages, namely, the *Orientation Identification* stage and the *Matching and Verification* stage.

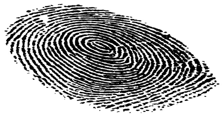


Figure 3. Left-oriented fingerprint

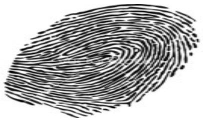


Figure 4. Right-oriented fingerprint

*1) Stage 1 – Orientation Identification:* When scanning a fingerprint, the user can place the finger on the scanning device at different angles. It could be aligned to left (see Figure 3) or right (see Figure 4). In fact, the position of the finger can be placed anywhere on the scanning surface. The scanning surface usually is somewhat larger compared to the fingerprint surface area. Hence, the first challenge is to exactly pinpoint the location and area of the fingerprint impression on the scanning surface.

The second challenge of course is to identify the orientation of the fingerprint. Without identifying the proper orientation, we can not properly compare it with the fingerprint(s) in the database and the recognition will no be possible. The task of this stage (i.e., Stage 1) is to identify and locate the fingerprint with its correct orientation.

*2) Stage 2 – Verification and Matching:* Like all other fingerprint recognition systems a database is maintained with fingerprint information against which the input fingerprint will be matched. In the database, we will store a black and white image. Once the orientation of the input fingerprint has been identified, we can easily reorient the fingerprint impression (according to the standard format stored in the database) and then the matching algorithm runs. Since finger print can contain dust, fudges, etc., the scanned information may contain errors which means that an exact match with the existing data is highly unlikely. So, in this stage (i.e., Stage 2) we perform an error tolerant matching in an effort to recognize the input fingerprint against the database of the system.

### B. Details of Stage 1: Orientation Identification

In this stage we employ a novel approach based on circular templates as follows. Let us use $f_i$ to denote the image of the input fingerprint. Let us assume that we know the appropriate center point, $p$ of $f_i$. We then can convert $f_i$ to a representation consisting of multiple circular bit streams by extracting circular segments of the image. This is achieved by constructing $k$ concentric circles $C_j$ of radius $r_j, 1 \leq j \leq k$, with center at point $p$. For each circle we obtain minutiae features of the image by storing 1 wherever the edge of a circle intersects with a ridge and a 0 if it intersects with a furrow [see Figure 5]. So, in this way, for $f_i$, we get $k$ concentric circles, which can be transformed into $k$ circular binary strings [see Figure 6]. Clearly, this procedure can be easily applied on a fingerprint data stored on the database. In what follows, we will use $Y_j, 1 \leq j \leq k$ to denote the $k$ circular strings obtained after applying the above procedure on a fingerprint data stored in the database. In what follows, we may slightly abuse the notation and say the $Y_j$ corresponds to the circle of radius $r_j$.

Now to identify the location and orientation of the input fingerprint we generalize the above approach to extract the minutiae feature and apply the approximate circular string matching algorithm of [30] as described below (please refer

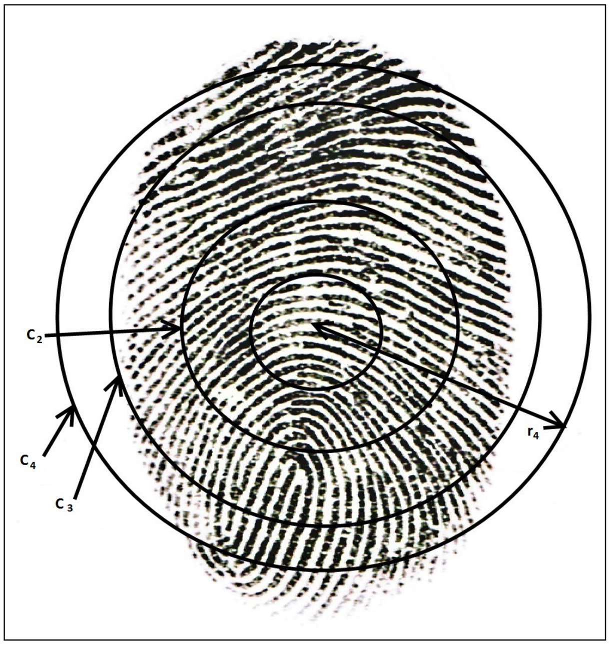


Figure 5. Fingerprint with scan circles

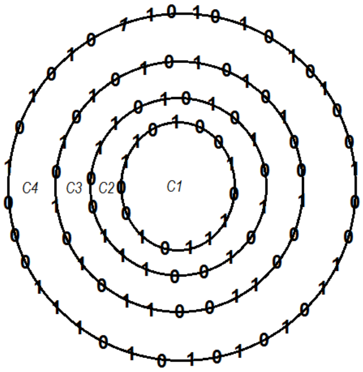


Figure 6. Intersection of a circle with the fingerprint

to Figure 7). What we do is as follows. For the input fingerprint, we cannot assume a particular center point to draw the concentric circle which is actually the main reason for difficulty in the process. So, instead, we take reference points at regular intervals across rows and columns of the entire frame of the image (i.e., the input scanning area) and at each point $p_\ell$, concentric circles $C_{j\ell}$ of radius $r_j$ are constructed. Like before, $k$ is the number of circles at each reference point $p_\ell$. So, from the above procedure, for each point $p_\ell$ we get $k$ circular strings $X_{j\ell}, 1 \leq j \leq k$.

At this point the problem comes down to identifying the best match across the set of same radius circles. To do this we make use of the Approximate Circular String Matching via Filtering (ACSMF) algorithm, presented in [30], which is accurate and extremely fast in practice. To do this we take a particular $X_{j\ell}$, construct $X_{j\ell}.X_{j\ell}$ (to ensure that all conjugates of $X_{j\ell}$ are considered) and apply algorithm ACSMF on $X_{j\ell}.X_{j\ell}$ and $Y_j$. In other words, we try to match the circular string $Y_j$ (corresponding to the circle of radius $r_j$) to all circular strings

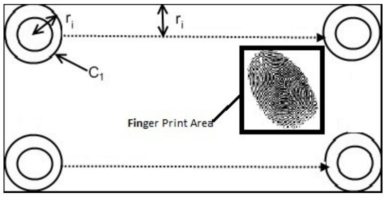


Figure 7. Identifying the orientation and surface area of the fingerprint impression

$X_{j\ell}$ (corresponding to the circle of radius $r_j$) generated at each point $p_\ell$. Thus we can identify the best matched circular string, i.e., the best matched circles and thereby locate and identify the fingerprint impression with the correct orientation. Once the orientation has been identified, we can apply standard techniques to reorient the image to match with the image from the database in the next stage.

### C. Details of Stage 2: Verification and Matching

Once Stage 1 of the algorithm is complete, we can assume that we have two images of the same size and orientation which we need to match and verify. We call this a verification process because in Stage 1 as well we have done a sort of matching already. However, we need to be certain and hence we proceed with the current stage as follows. Each image can now be seen as a two dimensional matrix of zero/one values, which can be easily converted to a (one dimensional) binary string. Now it simply comes down to pattern matching between two strings of the same length. However, note that, here as well we need to consider possibilities of errors. So, we simply compute the edit distance between the two binary strings and if the distance is within the tolerance level, we consider the fingerprint to be recognized. Otherwise, the authentication fails.

### D. Accuracy and Speed

We have two parameters that determine the accuracy of our approach. In Stage 1, the accuracy depends on the number of concentric circles, $k$. The larger the value of $k$, the higher the accuracy of pinpointing the location with the correct orientation. However, as $k$ increases the computational requirement and time also increases. In Stage 2, we have another parameter $d$ which is the tolerance level, i.e., the (edit) distance allowed between the two strings.

At this point a brief discussion on the response time of our algorithm is in order. Note that, the bulk of the computational processing in our approach is required in Stage 1, where we apply algorithm ACSMF to identify the best matched circles. As has been shown in [30], on average, ACSMF works in linear time in the size of the input and is practically extremely fast. The size of the circles and hence the corresponding circular strings are very small and can be assumed to be constant for all practical purposes. As a result the running time of Stage 1 would be extremely fast. Again, since the size of the fingerprint image is very small, any efficient approximate string matching algorithm in Stage 2 would give us a very quick result. Overall, this promises us an excellent turn-around time.

### E. Two Modes of Fingerprint Recognition System

As has been mentioned before, in terms of applications, there are two kinds of fingerprint recognition systems. So far we have only considered the mode where the input is a query fingerprint with an identity (ID) and the system verifies whether the ID is consistent with the fingerprint (i.e., verification mode). Here, the output is an answer of *Yes* or *No* and we need only match against one fingerprint from the database (i.e., the finger print coupled with the ID). To handle the other mode (identification mode), we need to match the query fingerprint against a list of fingerprints in the database. This can be done using an extension of algorithm ACSMF, namely Approximate Circular Dictionary Matching via Filtering algorithm (ACDMF) [32]. We omit the details here due to space constraints. Both ACSMF and ACDMF implementations are available at [33].

## VI. Conclusion

This paper has proposed a new pattern matching based approach for quick and accurate recognition of fingerprints. One overlooked feature in fingerprint matching is that the rotation of the fingerprint is assumed to be in sync with the stored image; in this paper we have tackled this issue. The novel element of this paper is the process of using a series of circles to transform minutiae information into string information consisting of 0s and 1s, and then using the approximate circular string matching algorithm to identify the orientation. This technique is expected to improve the performance and the accuracy of the fingerprint verification system. The proposed approach is currently under implementation on different smart phone platforms.

## Acknowledgement

M. Sohel Rahman is supported by a Commonwealth Academic Fellowship funded by the UK government and is currently on a sabbatical leave from Bangladesh University of Engineering and Technology (BUET).

## References

[1] S. Sebastian, “Literature survey on automated person identification techniques,” International Journal of Computer Science and Mobile Computing, vol. 2, no. 5, May 2013, pp. 232–237.

[2] J. Unar, W. C. Seng, and A. Abbasi, “A review of biometric technology along with trends and prospects,” Pattern Recognition, vol. 47, no. 8, 2014, pp. 2673 – 2688.

[3] P. Szor, The art of computer virus research and defense. Addison-Wesley Professional, 2005.

[4] National Criminal Justice Reference Service, The Fingerprint Sourcebook, A. McRoberts, Ed. CreateSpace Independent Publishing Platform, 2014.

[5] A. Agarwal, A. K. Sharma, and S. Khandelwal, “Study of rotation oriented fingerprint authentication,” International Journal of Emerging Engineering Research and Technology, vol. 2, no. 7, October 2014, pp. 211–214.

[6] P. Gutierrez, M. Lastra, F. Herrera, and J. Benitez, “A high performance fingerprint matching system for large databases based on gpu,” IEEE Transactions on Information Forensics and Security, vol. 9, no. 1, 2014, pp. 62–71.

[7] K. H. Q. Zhang and H. Yan, “Fingerprint classification based on extraction and analysis of singularities and pseudoridges,” in Fingerprint Classification Based on Extraction and Analysis of Singularities and Pseudoridges, ser. VIP 2001. Sydney, Australia: VIP, 2001.

[8] E. R. Henry, Classification and Uses of Finger Prints. Routledge, 1900.

[9] H. C. Lee, R. Ramotowski, and R. E. Gaensslen, Eds., Advances in Fingerprint Technology, Second Edition. CRC Press, 2002.

[10] S. Sebastian, “Literature survey on automated person identification techniques,” International Journal of Computer Science and Mobile Computing, vol. 2, no. 5, 2013, pp. 232–237.

[11] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition. Springer-Verlag, 2009.

[12] Griaule Biometrics. Online and offline acquisition. [Online]. Available: http://www.griaulebiometrics.com/en-us/book/ [retrieved: Nov., 2014]

[13] X. Tan and B. Bhanu, “Fingerprint matching by genetic algorithms,” Pattern Recognition, vol. 39, no. 3, 2006, pp. 465–477.

[14] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle, “An identity-authentication system using fingerprints,” Proceedings of the IEEE, vol. 85, no. 9, 1997, pp. 1365–1388.

[15] Z. M. Kovacs-Vajna, “A fingerprint verification system based on triangular matching and dynamic time warping,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 22, no. 11, 2000, pp. 1266–1276.

[16] X. Tan and B. Bhanu, “Robust fingerprint identification,” in International Conference on Image Processing 2002, vol. 1. IEEE, 2002, pp. I–277.

[17] C. Kai, Y. Xin, C. Xinjian, Z. Yali, L. Jimin, and T. Jie, “A novel ant colony optimization algorithm for large-distorted fingerprint matching,” Pattern Recognition, vol. 45, no. 1, 2012, pp. 151–161.

[18] A. A. Saleh and R. R. Adhami, “Curvature-based matching approach for automatic fingerprint identification,” in System Theory, 2001. Proceedings of the 33rd Southeastern Symposium on. IEEE, 2001, pp. 171–175.

[19] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, “Filterbank-based fingerprint matching,” Image Processing, IEEE Transactions on, vol. 9, no. 5, 2000, pp. 846–859.

[20] A. Jain, A. Ross, and S. Prabhakar, “Fingerprint matching using minutiae and texture features,” in Image Processing, 2001. Proceedings. 2001 International Conference on, vol. 3. IEEE, 2001, pp. 282–285.

[21] A. V. Ceguerra and I. Koprinska, “Integrating local and global features in automatic fingerprint verification,” in Pattern Recognition, 2002. Proceedings. 16th International Conference on, vol. 3. IEEE, 2002, pp. 347–350.

[22] A. K. Jain, S. Prabhakar, and L. Hong, “A multichannel approach to fingerprint classification,” IEEE Transactions on Pattern Analysis and Machine Intelligence , vol. 21, no. 4, 1999, pp. 348–359.

[23] M. R. Girgis, A. A. Sewisy, and R. F. Mansour, “A robust method for partial deformed fingerprints verification using genetic algorithm,” Expert Systems with Applications, vol. 36, no. 2, 2009, pp. 2008–2016.

[24] A. Ross and A. Jain, “Biometric sensor interoperability: A case study in fingerprints,” in Biometric Authentication. Springer, 2004, pp. 134–145.

[25] C. Xinjian, T. Jie, Y. Xin, and Z. Yangyang, “An algorithm for distorted fingerprint matching based on local triangle feature set,” Information Forensics and Security, IEEE Transactions on, vol. 1, no. 2, 2006, pp. 169–177.

[26] B. Smyth, Computing Patterns in Strings. Pearson Addison-Wesley, 2003.

[27] M. Lothaire, Applied Combinatorics on Words. Cambridge University Press, 2005.

[28] K. Fredriksson and S. Grabowski, “Average-optimal string matching,” Journal of Discrete Algorithms, vol. 7, no. 4, 2009, pp. 579–594.

[29] K.-H. Chen, G.-S. Huang, and R. C.-T. Lee, “Bit-Parallel Algorithms for Exact Circular String Matching,” Computer Journal, 2013.

[30] C. Barton, C. S. Iliopoulos, and S. P. Pissis, “Fast algorithms for approximate circular string matching,” Algorithms for Molecular Biology, vol. 9, no. 9, 2014.

[31] A. V. Aho and M. J. Corasick, “Efficient string matching: an aid to bibliographic search,” Communication ACM, vol. 18, 1975, pp. 333–340.

[32] C. Barton, C. S. Iliopoulos, S. P. Pissis, and F. Vayani, “Accurate and efficient methods to improve multiple circular sequence alignment,” submitted.

[33] S. P. Pissis. ACSMF and ACDMF implementation. [Online]. Available: http://github.com/solonas13/bear/ [retrieved: Nov., 2014]

# Fast Road Vanishing Point Detection Based on Modified Adaptive Soft Voting

Xue Fan, Cheng Deng, Yawar Rehman, and Hyunchul Shin
Department of Electronics and Communication Engineering
Hanyang University, Ansan, Korea
e-mail: {fanxue, dengcheng, yawar}@digital.hanyang.ac.kr, shin@hanyang.ac.kr

***Abstract*— Detecting the vanishing point from a single image is a challenging task since the information contained in the input image which can be used to detect the location of vanishing point is very limited. In this paper, we propose a framework for vanishing point detection based on the Modified Adaptive Soft Voting (MASV) scheme. First of all, the input image is convolved with the generalized Laplacian of Gaussian (gLoG) filters, which are used to estimate the texture orientation map. Then, MASV scheme is used to get accurate voting map. Finally, peak identification is performed on the voting map to locate the vanishing point. In addition, a scaling method for voting map computation is proposed to further accelerate the vanishing point detection algorithm. Through experiments, we show that the proposed algorithm is about 10 times faster and outperforms by 4.64% on an average than the complete-map based gLoG, which is the state-of-the-art method.**

***Keywords-vanishing point detection; gLoG; MASV; voting map.***

## I. INTRODUCTION

Automatic driver assistant systems and driverless vehicles have been the focus of attention for many computer vision researchers over the last twenty years [1]. There are numerous researchers who devoted themselves for developing Autonomous Vehicle Navigation Systems (AVNS) in either structured [2][3], urban environments [4] or unstructured roads [5]. Road detection is one of the crucial parts of the AVNS. Considering the continuously changing background, traffic conditions, and road types, robust road detection using a monocular vision system is a challenging problem. Many vison-based road detection methods have been proposed. Among all road detection methods, vanishing point constrained road detection schemes have given promising results in detecting both off-road areas [5][6] and urban roads [7][8].

The current method of road vanishing point detection can be generalized into three main categories: edge-based methods [2][3], prior-based methods [9][10], and texture-based method [5][11][12].

In edge-based vanishing point detection methods, two or more dominant straight lines (segments), which correspond to road borders or markings, are detected by Hough transform or random sampling and consensus, and the vanishing point is detected as the intersection of these straight lines (segments). For structured roads with well-paved lanes, edge-based approaches can be used for real-time application due to their accuracy and computation efficiency. While for unstructured roads without edges or contrasting local characteristics, their performance is really limited.

Recently, some prior based techniques have been proposed. They intend to integrate contextual 3D information with low-level features in order to improve the detection performance. Such weak contextual cues include a 3D scene layout, 3D road stages, temporal road cues, and so on. It is really difficult to apply these prior based methods in real-time and practical situations. Besides, all these methods assume that the road is structured and well-paved. In order to overcome this limitation texture-based approach is proposed [5][12][13], which can accurately detect vanishing point of both well-paved roads and unstructured off-roads. Texture-based vanishing-point detection methods apply a bank of oriented filters to compute the texture orientation of each pixel. Then each pixel votes for the candidate vanishing point through a pre-defined voting scheme. Either the local soft voting scheme proposed by Kong et al. [5], or the global voting scheme proposed in [6][11] is time-consuming and cannot meet the requirement of real-time applications.

In this paper, we propose a texture-based method for detecting vanishing point from a single image. Specially, the contributions of this paper are as follows. First, a scaling method for fast voting map computation is proposed. We first down-sample the input image, and the voting map is computed for the down-sampled image. Then, the final voting map is up-sampled for vanishing point detection. As a result, the detection time is significantly reduced. Second, we propose MASV scheme for the purpose of accurate and effective voting map generation. Instead of using exponential function based soft-voting in [12], we use Gaussian function for weighted voting [14], which is more adaptive and robust. In addition, we define a voting radius $R_v$ to limit the voting region of each pixel and to reduce redundant voting.

The rest of this paper is organized as follows. A brief review of the related work is presented in Section II. The proposed MASV scheme based vanishing point detection is explained in Section III. Experimental results are shown in Section IV. Finally, the conclusions are summarized in Section V.

## II. RELATED WORK

A vanishing point is a point in perspective images to which parallel lines converge. It plays a leading role in road detection.

Most of the existing edge-based vanishing point detection algorithms rely on three steps [2][3]. The first step performs edge detection on the input image in order to extract the most dominant edges such as road borders or lane markings. The next step is to determine whether there are any line segments in the image. Once all the line segments are identified, a

ISBN: 978-1-61208-393-3

voting procedure is applied to find the intersections of the lines.

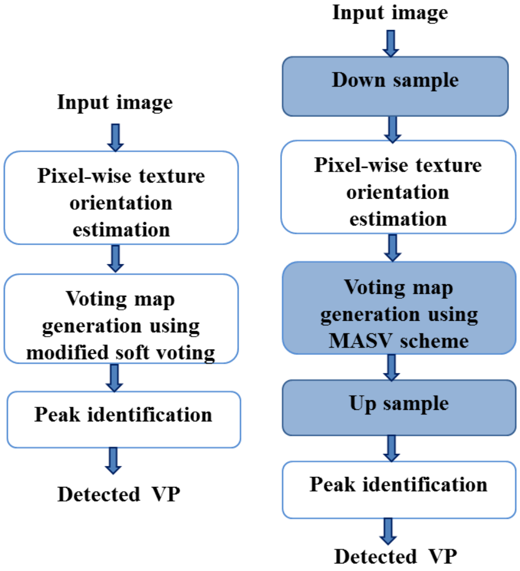
Wu et al. [10] proposed a global perspective structure matching (GPSM) scheme based on an image retrieval technique to identify the best candidate images in an image database, and then use the pre-labeled vanishing points of the best candidate images as the initial estimation of input image’s vanishing point, and finally, a probabilistic model of vanishing point is used to refine the location of vanishing point. For these prior-based methods, not only a large scale image or video database is necessary in order to make these prior-based methods robust to various imaging conditions, road types and scenarios, but the training algorithm is also very important, and not to mention laborious manual label works for the training stage.

Texture-based methods are proposed to overcome the drawback of edge detection based and prior-based vanishing point detection methods. Firstly, a bank of oriented filters is applied, such as Gabor filter banks [13] and steerable filter banks [15], to estimate the dominant texture orientation of each pixel and generate the texture orientation map. Then, pixel-based voting is performed to obtain the voting map. Each pixel votes for the candidate vanishing point through a pre-defined voting scheme. Finally, vanishing point is detected by using peak point identification.

## III. Proposed Vanishing Point detection approach

In this part, our improved method will be explained in detail. Figure 1 shows the workflow comparison of our proposed method and the complete-map based gLoG vanishing point detection method [12]. The shaded blocks show the new contributions of our method.

### A. *Generalized Laplacian of Gaussian (gLoG) Filter*


Input image
Pixel-wise texture orientation estimation
Voting map generation using modified soft voting
Peak identification
Detected VP
Input image
Down sample
Pixel-wise texture orientation estimation
Voting map generation using MASV scheme
Up sample
Peak identification
Detected VP

(a) The gLoG method (b) Our method

Figure 1. The framework of our proposed method and the complete-map based gLoG method.

The standard 2-D Gaussian function and the generalized 2-D Gaussian function [12] are defined as in (1) and (2) respectively.

$$G_s(x,y,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{x^2+y^2}{2\sigma^2}) \quad (1)$$

$$G_g(x,y) = A \cdot exp(-(ax^2+2bxy+cy^2)) \quad (2)$$

where $A$ is the normalization factor, the coefficients a, b, and c explicitly control the shape and orientation of kernel $\mathrm{G}_g(\mathrm{x},\mathrm{y})$ by means of $\theta, \sigma_x$, and $\sigma_y$.

$$a = \frac{cos^2\theta}{2{\sigma_x}^2} + \frac{sin^2\theta}{2{\sigma_y}^2} \quad (3)$$

$$b = -\frac{sin2\theta}{4{\sigma_x}^2} + \frac{sin2\theta}{4{\sigma_y}^2} \quad (4)$$

$$c = \frac{sin^2\theta}{2{\sigma_x}^2} + \frac{cos^2\theta}{2{\sigma_y}^2} \quad (5)$$

Then the gLoG filter can be presented as follows:

$$\nabla^2 G_g(x,y) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2} \quad (6)$$

Figure 2 shows the gLoG kernels with different shapes (controlled by $\sigma_x$ and $\sigma_y$) and orientations (controlled by $\theta$). The first row in Figure 2 shows the 2-D circular LoG filters. Compared with the circular LoG kernels, we can see that the proposed gLoG filters can be readily applicable to estimate the orientations of local image textures.

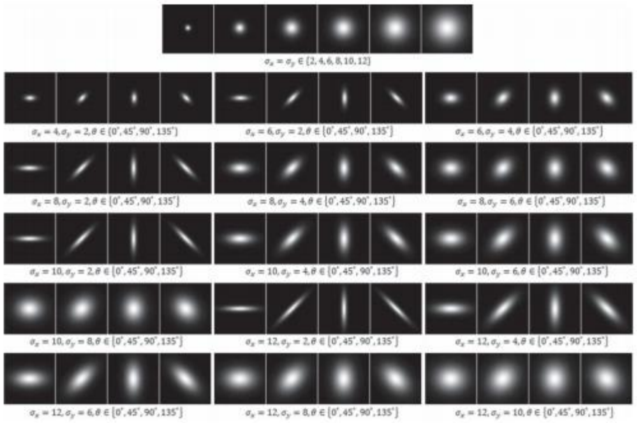

$\sigma_x = \sigma_y \in \{2,4,6,8,10,12\}$
$\sigma_x = 4, \sigma_y = 2, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$
$\sigma_x = 6, \sigma_y = 2, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$
$\sigma_x = 6, \sigma_y = 4, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$
$\sigma_x = 8, \sigma_y = 2, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$
$\sigma_x = 8, \sigma_y = 4, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$
$\sigma_x = 8, \sigma_y = 6, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$
$\sigma_x = 10, \sigma_y = 2, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$
$\sigma_x = 10, \sigma_y = 4, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$
$\sigma_x = 10, \sigma_y = 6, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$
$\sigma_x = 10, \sigma_y = 8, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$
$\sigma_x = 12, \sigma_y = 2, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$
$\sigma_x = 12, \sigma_y = 4, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$
$\sigma_x = 12, \sigma_y = 6, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$
$\sigma_x = 12, \sigma_y = 8, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$
$\sigma_x = 12, \sigma_y = 10, \theta \in \{0^\circ,45^\circ,90^\circ,135^\circ\}$

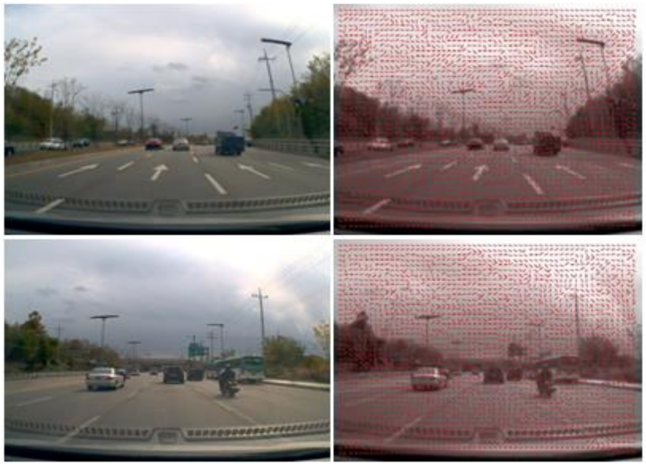
Figure 2. Generalized LoG filters [12].

### B. *Pixel-Wise Texture Orientation Estimation*

In order to estimate texture orientations, we need to generate a set of gLoG filters. As mentioned above, the gLoG filters are determined by $\sigma_x$, $\sigma_y$ and $\theta$. According to the experimental results in [12], we set ${\sigma_x}^{max}$ =16, ${\sigma_y}^{min}$ =4, the number of orientations $n_\theta$=12 ($n_\theta = \frac{180^\circ}{\theta}$). Because $\sigma_x$ and $\sigma_y$ are interchangeable, we set $\sigma_y$ smaller than $\sigma_x$.

ISBN: 978-1-61208-393-3

After generating a set of gLoG kernels by using different combinations of $\{\sigma_x, \sigma_y,\ \theta\}$, we divide the produced kernels into $n_\theta$ groups, where each groups only contains the kernels with the same orientation. Then, the test image is convolved with every kernel in each group. Each pixel's orientation is determined by the group that can produce the maximum convolution response. Figure 3 shows the estimated texture orientation map. In Figure 3(a), the image is overlaid with texture orientation bars at evenly sample locations.



(a) Input image (b) Texture orientation map

Figure 3. Visualization of estimated texture orientation map.

### C. Voting Map Generation Using MASV Scheme

After estimating the texture orientation map at each pixel of the image, one can make these pixels vote to obtain the voting map. Then, vanishing point will be detected by using peak identification in the voting map.

Below is the voting scheme proposed in [12], which has achieved very good detection result.

$$Vote(p,v) = \begin{cases} exp\left(-d(p,v) * \frac{|\gamma|}{l}\right), if\ |\gamma| \le \delta \\ \quad\quad 0, \quad\quad otherwize \end{cases} \quad (7)$$

where $d(p,v)$ is the Euclidean distance between pixel $p$ and $v$, $l$ is the normalization factor, which is set to the diagonal length of the input image. $\delta$ is set to the angular resolution $\frac{180^{\circ}}{n_\theta}$. In (7) pixel $p$ whose vector is $\overline{V_p}$ can vote for any pixels above $p$ as long as the angle between the direction *(pv)* and the vector $\overline{V_p}$ is below the threshold value $\delta$, which means that each pixel in the image will vote for the sky region. It's really time consuming and will also introduce a lot of noise.

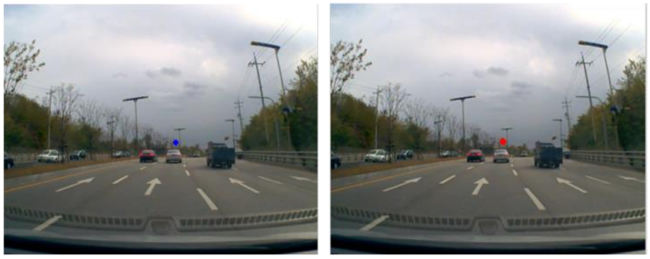
The complete-map based gLoG method uses exponential function for weighted voting. Using this voting scheme, the votes decrease rapidly as the distance increases, and pixels will mainly vote for the locations nearby themselves. Whereas vanish point is the converging point of the parallel lines in perspective images, and at least more than half of the pixels which belong to the road regions are far away from the vanishing point. It means that most of the pixels belonging to the road regions cannot generate effective votes to the ground truth vanishing point. Compared to exponential function, Gaussian function is more adaptive. As the distance increases, the weighted votes decrease slowly. Hence, we propose a modified adaptive voting scheme to further improve the voting accuracy, as shown in (8).

$$Vote(p,v) = \begin{cases} exp\left(-\frac{(d(p,v)*|\gamma|)^2}{2\sigma^2 * l}\right), if\ |\gamma| \le \delta\ and\ d(p,v) < R_v \\ \quad\quad\quad 0, \quad\quad otherwize \end{cases} \quad (8)$$

where $\mathrm{R_v}$ is the radius of the voting region, $l$ is the normalization factor, and $\sigma$ is experimentally set to 20.

### D. Speed Up by Using Image Scaling

Although the accuracy of vanishing-point detection is very promising based on the pixel-wise texture orientation estimation and voting, whereas, it is really time-consuming during the voting stage. The complexity of the voting stage [14] is $\mathrm{O}(\mathrm{w}^2 * \mathrm{h}(\mathrm{h}+1)/2)$, assuming that the dimension of the input image is $\mathrm{w} * \mathrm{h}$. We can see that the complexity of the voting stage is mainly determined by the dimension of the input image. If the image is down sampled, its computation time will be significantly reduced. Hence, in this paper, a scaling method is used to reduce the computation time.



(a) No scaling (b) 1/4 scaling

Figure 4. Detected vanishing point with different scaling factors of the voting map.

The method is very simple, we first down sample the input image, and then compute the voting map of the down sampled image. The final voting map is up sampled by up sampling the voting map. Considering that the bilinear interpolation makes a good trade-off between computation time and image quality, we use the popular bilinear interpolation method for image scaling. The scale factor is experimentally set to 1/4. The results are shown in Figure 4. As we see, almost no difference can be found for different scales in the detection of vanishing point.

## IV. EXPERIMENTAL RESULTS

There are several methods reported in [12]. The one based on the complete texture orientation map (complete-map based gLoG) can achieve the best result among all. In our paper, fair comparison is performed with the complete-map based gLoG [12], which is considered as the state-of-the-art method.

The experiments contain the comparison of vanishing point detection accuracy and the computation time. The dataset in this experiment consists of 1000 images taken from local roads with various types of background, illuminations, and traffic conditions. For every image, the ground truth of vanishing point is manually labeled through the method suggested in [12].

In order to measure the accuracy of vanishing point detection methods, we use the normalized Euclidean distance as suggested in [16][17], where the Euclidean distance between the estimated vanishing point and the ground truth is normalized by the length of diagonal of the input images as follows:

$$NormDist = \frac{\|V_e(x,y) - V_g(x,y)\|}{DiagImage} \quad (8)$$

where $V_e(x,y)$ and $V_g(x,y)$ are the estimated vanishing point and the ground-truth of vanishing point respectively, DiagImage is the diagonal length of the given image, which is used for normalization. If NormDist value is close to 0, it means that the detected vanishing point is close to the ground truth; otherwise, it may correspond to incorrectly estimated vanishing point.

### A. Vanishing Point Detecting Accuracy Comparsion

Figure 5 shows the comparison of vanishing point detection accuracy between our method and the complete-map based gLoG. The *x* axis is the *NormDist* distance between the detected vanishing point and the ground truth, and y axis is the detection accuracy.

We will consider the detected vanishing point is accurate when the *NormDist* distance between the detected vanishing point and the ground truth is smaller than a certain *NormDist* value, as suggested in references [5]][12][16][17]. It can be seen that our method outperforms by 4.64% on average under the same *NormDist* distance. Figure 6 shows some examples of detected vanishing points by our method and the reference method.

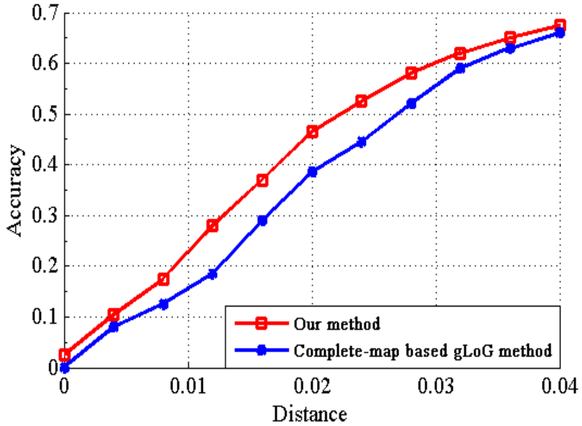


Figure 5. Results comparison of our method and the complete-map based gLoG method.

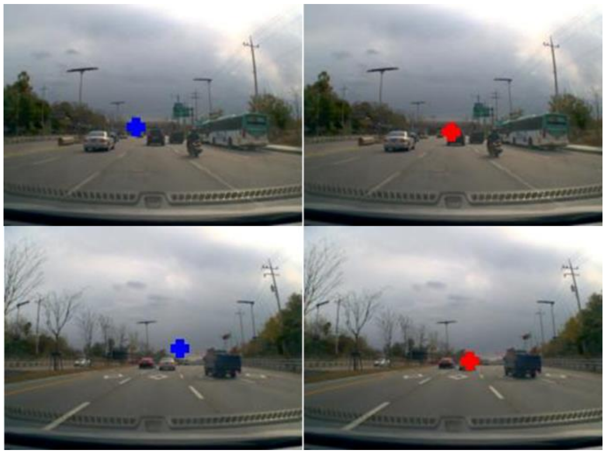


Figure 6. Sample detection results of our method (right) and the complete-map based gLoG method (left).

### B. Computation Time Comparison

The experiments are conducted in Matlab with a 3.4 GHz Intel i7 Processor. Table 1 shows the computation time comparison of our method and the complete-map based gLoG method [12].

TABLE I. COMPARISON OF CPU TIME

| *Image Size* | *Complete-map based gLoG* | *Our Method* | *CPU time reduced* |
|---|---|---|---|
| 360*480 | 594.5 s | 46.8 s | 92.1% |

We can see that our method is approximately 12 times faster than the complete-map based gLoG method.

## V. CONSLUSION

In this paper, we have proposed a new framework for vanishing point detection. By using image scaling and MASV, our approach shows better results both in detection accuracy and CPU time than those of complete-map based gLoG. We believe that our method is useful for fast and robust vanishing point detection.

## ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MOE) (No. NRF-2013R1A1A2004421).

## REFERENCES

[1] J. C. McCall and M. M. Trivedi, “Video Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation,” IEEE Transactions on Intelligent Transportation Systems, vol. 7, no.1, Mar. 2006, pp. 20-37.

[2] Y. Wang, E. K. Teoh, and D. Shen, “Lane Detection and Tracking Using B-snake,” Image and Vision Computing, vol. 22, Apr. 2004, pp. 269-280.

[3] T. Suttorp and T. Bucher, “Robust Vanishing Point Estimation for Driver Assistance,” Proc. of the IEEE Intelligent Transportation Systems Conference, 2006, pp. 1550-1555.

[4] Y. He, H. Wang, and B. Zhang, “Color-based Road Detection in Urban Traffic Scenes,” IEEE Trans. on Intelligent Transportation Systems, vol. 5, no. 4, Dec. 2004, pp. 309-318.

[5] H. Kong, J. Audibert, and J. Ponce, “General Road Detection from A Single Image,” IEEE Trans. on Image Processing, vol. 19, no. 8, Aug. 2010, pp. 2211-2220.

[6] C. Rasmussen, “Grouping Dominant Orientations for Ill-structured Road Following,” Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2004, pp. I-470–I-477.

[7] Q. Wu, W. Zhang, and B. V. Kumar, “Example-based Clear Path Detection Assisted by Vanishing Point Estimation,” Proc. of the IEEE International Conference on Robotics and Automation, 2011, pp. 1615–1620.

[8] N. Hautiere, J. Tarel, and D. Aubert, “Mitigation of Visibility Loss for Advanced Camera-based Driver Assistance,” IEEE Transaction Intelligent Transportation Systems, vol. 11, no. 2, Jun. 2011, pp. 474–484.

[9] J.M. Alvarez, T. Gevers, and A.M. Lopez, “3D Scene Priors for Road Detection,” Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 57-64.

[10] Q. Wu, T. Chen, and B. Kumar, “Prior-based Vanishing Point Estimation Through Global Perspective Structure Matching,” Proc. of IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), 2010, pp. 2110-2113.

[11] P. Moghadam, J. A. Starzyk, and W. S. Wijesoma, “Fast Vanishing Point Detection in Unstructured Environments,” IEEE Transactions on Image Processing, vol.21, no.1, Jan. 2012, pp. 425-430.

[12] H. Kong, S. Sarma, and F. Tang, “Generalizing Laplacian of Gaussian Filters for Vanishing-Point Detection,” IEEE Transactions on Intelligent Transportation Systems, vol.14, no.1, Mar. 2013, pp. 408-418.

[13] H. Kong, J.Y. Audibert, and J. Ponce, “Vanishing Point Detection for Road Detection,” Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 96-103.

[14] H. Guo, “A Simple Algorithm for Fitting a Gaussian Function [DSP Tips and Tricks],” IEEE Transactions on Signal Processing Magazine, vol.28, no.5, Sept. 2011, pp. 134-137.

[15] M. Nieto and L. Salgado, “Real-Time Vanishing Point Estimation in Road Sequences Using Adaptive Steerable Filter Banks,” Proc. of International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS), 2007, pp. 840–848.

[16] P. Moghadam, J.A. Starzyk, and W.S Wijesoma, "Fast Vanishing-Point Detection in Unstructured Environments," IEEE Transactions on Image Processing, vol.21, no.1, Jan. 2012, pp. 425-430.

[17] W. Yang, X. Luo, B. Fang, and Y. Y.Tang, "Fast and accurate vanishing point detection in complex scenes," IEEE International Conference on Intelligent Transportation Systems (ITSC), Oct. 2014, pp. 93-98.

# A Rule for Combination of Spatial Clustering Methods

Danielly C. S. C. Holmes, Ronei M. Moraes, Rodrigo P. T. Vianna

Graduate Program in Decision Models and Health
Federal University of Paraiba
João Pessoa, Brazil
Email: daniellycristina9@gmail.com, ronei@de.ufpb.br, rodrigopissoa@gmail.com

***Abstract*—In the area of spatial analysis, spatial clustering methods use georeferencing information in order to identify significant and non-significant spatial clusters of the phenomenon in study in a specific geographical region. Several methods are available in the literature, such as scan statistic, Getis-Ord statistics, and the Besag and Newell method. In practical applications, all those methods are not able to produce results which can capture the real event with good accuracy. In this paper, we propose using the a combining classifier technique in order to provide better results for spatial clustering methods, using the majority voting rule for that combination. A study case was presented using epidemiological data of dengue fever from state of Paraba, Brazil, in the year of 2011 and the final results allowed to identify the priority and non-priority areas in the region of interest.**

***Keywords–Majority vote rule; Spatial clustering methods; Statistical significance.***

## I. INTRODUCTION

A classifier is defined as a function, whose domain is an attribute space in $R^n$ and its co-domain is a set of class labels $\Omega$ with $K$ elements, where $\Omega = \{w_1, ..., w_K\}$ [16][18]. Classification has been an area for research in the pattern recognition and machine learning communities [7]. The classification process can be performed using supervised classifiers and unsupervised classifiers. The supervised classifiers require a previous knowledge of the problem, which is translated by a training database that contains labeled samples. The unsupervised classifiers are performed using a database of unlabeled samples, i.e., samples for which their class are unknown. So, there is no previous knowledge about the real class labels [13].

In the scientific literature, several cases can be found, in which combining multiple classifiers provided an improvement of the results with respect to each individual classifier performance. So, that combination makes them more efficient [5][9][10][19][21][22].

Combining classifiers can be done using three architectures: in sequential (or linear) way, in parallel or hierarchically [24]. In order to provide the final decision, an architecture should be chosen, as well as a combination scheme of classifiers, which is called combiner [2]. One of these schemes is the static combiner, which performs combination using a predefined rule and no training is required over that architecture [24]. The architecture chosen and the combination scheme (including the combination rule) allow to create a new classifier [2].

In the past years, the number successfully applications combining classifiers is increasing in many areas, as, for instance, image classification, writing and character recognition; among others [5][17]. Several schemes can be found in the literature to combine classifiers, as voting (and its variations), sum, mean, median, product, minimum, maximum, adaptive weighting [2], logistic regression [5], Dempster-Shapher theory and mixture of experts [22]; among others [24].

In the area of spatial analysis, spatial clustering methods use georeferencing information in order to identify significant and non-significant spatial clusters of the phenomenon in study in a specific geographical region. Several methods are available in the literature, such as scan statistic [1][15], Getis-Ord statistics [3] and the Besag and Newell method [4][6]. In practical applications, all those methods are not able to produce results which can capture the real event with good accuracy. Each method works with different methodologies and provides different results with respect to the others. In addition to these issues, as there is no reference information about the real clusters, it is not possible to check the similarity between the results produced by one method and the true result. Thus, it is possible to use just indirect forms of evaluation, as for instance, maps of relative risk, in studies in public health [25]. These problems have do not have a good solution yet.

The first problem is quite similar to the classification problem which is solved by using combining classifiers. So, in this paper, we propose using the combining classifier technique in order to provide better results for spatial clustering methods.

This paper is organized as follows: the next section presents some theoretical aspects of spatial clustering methods and combining rules. Section 3 brings the new methodology proposed. On Section 4, the results obtained, as well as their analysis, are presented, followed by some considerations in Section 5.

## II. THEORETICAL ASPECTS

In this section, some theoretical aspects are presented. Three methodologies of the spatial clustering are shown: Scan statistic, Getis-Ord statistic, and the Besag and Newell method. All these methods are used in order to identify significant and non-significant regions (binary information) in a geographic area of interest. It means, high values or small ones, which are statistically different to the others in a sub-region will be assigned on map with different signal of the others, which are not significant. The significant regions on the map are named spatial clusters. The main goal is to identify significant areas, visualize and describe the spatial patterns [12].

Statistical functions provide measure for the spatial associations and evaluate the statistical significance for it. They

ISBN: 978-1-61208-393-3

are divided into: global, local, and focused statistics. Global statistics identify the spatial structure which can be cluster autocorrelation, but not identifying the location of cluster or quantify the spatial dependency. Local statistics quantify spatial autocorrelation and clustering within the small areas in the geographical region of interest, i. e. they search for regions which are significantly different from the area where they are inserted. The focused statistics quantify clustering on certain specific location, which is named *focus*. In Spatial Epidemiology, these kind of tests can relate information about incidence of a disease and possible sources of contamination in the same region [12]. According to Knox [23], local statistics are generic tests and focused statistics are focused tests. In this paper, we use only generic tests.

### A. Scan Statistic

The scan statistic does a survey of the study region, looking for the most likely significant events. The survey occurs in the following way: for each sub-region, a centroid $\xi_i$ is associated, it contains a random variable $X$, which denotes the numbers of individuals that have the disease along with the population size on that sub-region. This method is based on circles positioned over each centroid, in which the radius $r$ can be the greatest measure that involves a new neighboring centroid and within it a percentage of the population [15]; in other words, multiple circles are generated with different radius and different geographical localizations [11]. This process is finalized when all the centroids have been tested. The hypotheses are:

$H_0$: There is no spatial cluster in the geographical region.

$H_1$: There is at least one spatial cluster in the geographical region.

The hypotheses are tested by the Monte Carlo simulation [1]. As the circles are increaseding, a likelihood test is performed, in which we verify if the study region is a conglomerate. The test is based on the maximum likelihood method [18], assuming some probabilistic distributions, and the evaluation is done using the Monte Carlo simulation [1]. The Monte Carlo simulation is used to test if the clusters are statistically significant. The hypotheses test via Monte Carlo are generated simultaneously from simulated data multiple times under the null hypothesis and the p-value is $\frac{r}{(R+1)}$, in which the $R$ is the number of occasional data repetition of the simulated data and $r$ is the classification of the statistical test [1].

### B. Getis-Ord Statistic

The Getis-Ord statistic measures the spatial association between the spatial dependencies functions. It performs the distance measurements only with the positive observations and with data that have a non-normal distribution [3].

The Getis-Ord statistics is estimated by groups of neighbors of the critical distance d of each area $i$. The critical distance is formed by a proximity matrix $W$, in which the elements are formed in function of the critical distance $w_{ij}(d)$. With that, two statistical functions were proposed: the Getis-Ord statistic evaluates the significance of the statistic generated. It is said to be significant if the p-value is lower than the adopted significance [3]. The global statistic $G(d)$ is equal to the traditional measures of spatial agglomeration with just one value $G(d)$. The global statistic is given by:

$$G(d) = \frac{\sum_i \sum_j w_{ij} x_i x_j}{\sum_i \sum_j x_i x_j} \tag{1}$$

in which $x_i \in X$ is a value observed in the position $i$ and $w_{ij}(d)$ an element of the proximity matrix. The level of significance is defined as the probability of rejecting the null hypothesis (existence of spatial autocorrelation), if it is true. The p-value confronted with the adopted significance defining the significance of the Getis-Ord index generated. The analysis is based on the value of the index and its significance: the positive and significant value of $G(d)$ indicates spatial agglomeration of high values, the negative and significant values of $G(d)$ indicate spatial agglomeration of small values [3].

The local statistic $G_i$ and $G_i^*$ are measures of the spatial association for each area and they measure the association in each spatial unit for each observation $i$, in which $G_i$ and $G_i^*$, shows the position which is surrounded by high or low values for the variable. The $G_i(d)$ equation for each observation $i$ and distance $d$ is shown in the following way [3]:

$$G_i(d) = \frac{\sum_{j, j \neq i} w_{ij} x_j}{\sum_j x_j}, \tag{2}$$

in which all positions $j$, except those ones where $j = i$, can be in the sum. This index is equal to the ratio of the sum of the values in the neighbouring positions by the sum of the values in the whole data series. However, in the statistic of $G_i^*$, all values of $j$, including those ones where $j = i$ are included in the sum [3].

$$G_i(d) = \frac{\sum_j w_{ij} x_j}{\sum_j x_j} \tag{3}$$

TABLE I. INTERPRETATION OF THE LOCAL INDEX SIGNIFICANCE.

| *Significance* | *Statistic* | *p-value* |
|---|---|---|
| Negative*** | Negative | p<0.005 |
| Negative** | Negative | 0,005<p<0,025 |
| Negative* | Negative | 0,025<p<0,05 |
| Negative | Negative | p>0,05 |
| Positive | Positive | p>0,05 |
| Positive* | Positive | 0,025<p<0,05 |
| Positive** | Positive | 0,005<p<0,025 |
| Positive*** | Positive | p<0,005 |

The Getis-Ord local index is interpreted as follows: the positive and significant standardized values (p-value less than 5%) means a spatial agglomeration with high values. The significant negative standardized statistical values (p-value less than 5%) indicates a spatial agglomeration with low values. According to Table 1, the interpretation is given in the following way: The smaller p-value implies the higher agglomeration and it does not matter whether is a positive or negative spatial agglomeration [3].

### C. Besag and Newell Method

The Besag and Newell method [4] produces circular spatial clusters. The process is: a radius is determined in such a way that it contains a circle with at least $p$ cases in its interior. The method starts with the circle radius equal zero. When the circle achieves $p$ cases, the process stops; if that does not occur, the radius is increased, including a new centroid. The procedure is executed until at least $p$ cases are found or when the number of centroids is finished.

Let $C$ be the total number of cases in the study region and $Y$ the total population exposed to the risk in the region. Let $C_j(i)$ and $M_j(i)$ be the number of cases and the population of the $j$ areas closer to the centroid. The statistic of the test is based on the random variable $A$, defined as the minimum of areas next to the centroid [6]. So, we have:

$$A = \min_j \{C_j(i)\} \geq p \tag{4}$$

in each centroid is verified the existence of a spatial cluster. The cluster is said significant if the p-value is less the adopted significance. From the value $a$ observed for $A$, the level of significance of the test is defined by $P(A \leq a)$, which tests the null hypothesis (absence of spatial clusters). The significance, denoted by $p_k(i)$ is calculated by the following equation [6]:

$$\begin{aligned} p_k(i) = P(A_i \leq a_i) = 1 - \sum_{j=1}^{k-1} \frac{(M_j(i)C/M)^j}{j!} \times \\ \times\, exp(M_j(i)C/M) \end{aligned} \tag{5}$$

in which $M_j$ is the population observed in the area $j$.

### D. Classifiers Combining Rule

As previously mentioned, there are several rules for combining multiple classifiers. In this section, two methods are presented and it is shown that they are equal when applied in the binary case [26].

The *majority voting* is the most popular rule for combining classifiers [8]. The majority voting rule defines the winner class as that one which obtained more than half of the total number of votes. If there is no class in this condition, then $x \in X$ did not receive a label (it works as a rejection option). Let $\Delta_{ji} \in 0, 1$ be the vote for the class $j$ which was signed by the classifier $i$. Let $H$ be a decision function which sign the final class for $x$, then:

$$H(x) = \begin{cases} j, & \text{i}f \sum_{i=1}^{D} \Delta_{ji}(x) = \\ & = \sum_{j=1}^{K} \sum_{i=1}^{D} \Delta_{ki}(x) \\ \text{r}ejection, & \text{o}therwhise. \end{cases} \tag{6}$$

where $K$ is the number of classes in $\Omega$ and $D$ is the number of classifiers [8][14].

Another kind of voting rule for combining classifiers is the *plurality voting*. In this case, the winner class is that one which receives the largest number of votes, i. e., it is not necessary achieve to get more than 50% of classifiers votes. Its equation is given by:

$$H(x) = j, \qquad if \sum_{i=1}^{D} \Delta_{ji} = \max_k \sum_{i=1}^{D} \Delta_{ki} \tag{7}$$

According to Zhou [26], in the case of binary classification, the *majority voting* and the *plurality voting* produce the same results.

## III. Methodology

As mentioned before, spatial clustering methods use georeferencing information in order to identify significant and non-significant spatial clusters of a phenomenon in study in a geographical region of interest. All methods available in the literature are not able to produce results which can capture the real event with good accuracy and it is possible to use just indirect forms of evaluation of their results. In the applications using public health data, maps of relative risk can be used for this purpose. The measure of relative risk is defined as the probability of an individual to have a disease in a determined time divided by the cumulative incidence of the area of the interest [20].

The final result provided by a spatial clustering method is a georeferencing list of significant centroids, i.e., a database which contains the pair $(centroid, label)$. The methodology consists of applying an impair number of spatial clustering methods on the same area and data. From those applications, we obtained a number of georeferencing lists. Finally, over them is applied a voting rule in order to obtain the final class for each centroid in the region of interest.

In this paper, we applied that methodology using the three spatial clustering methods presented above, on the same area and data. From those applications, we obtained three georeferencing lists and we applied the majority voting on them. It is worth noting that the problem is a binary classification, then majority voting and plurality voting produce the same results.

## IV. Results

The methodology designed and presented in the previous section was applied on epidemiological data of dengue fever from state of Paraba, Brazil, in the year of 2011. As dengue fever is a tropical disease, it is recurrent health problem in all country. Due to financial restrictions, it is important for the health authorities to know the areas in which the relative number of cases is significant larger than others, as well as areas where the relative number of cases is significantly lower than others. The first areas can be called priority areas and the second areas can be named protection areas.

Applications results for the three spatial clustering methods (Scan statistic, Getis-Ord statistic, and the Besag and Newell method) are presented in Figures 1-3. The relative risk map is presented in Figure 4 and the final decision map, obtained by the combination of those three spatial clustering methods using majority voting is presented in Figure 5. In the results below, we show the comparison of the decision map (result of the combination of spatial clustering methods from the majority vote rule) with all spatial clustering methods.

In the comparison of the final decision map (Figure 5) with the Scan statistic map (Figure 1), all the 16 cities on the final decision map (Figure 5) are in the scan statistic map. The Scan statistic map identified 53 cities with significant values and from those, 16 are present in the final decision map.

In the comparison of the final decision map (Figure 5) with the Getis and Ord map (Figure 2), of the 16 cities on the decision map, 10 are on the Getis-Ord map. According to the Getis-Ord map, 10 cities are in spatial clusterings of negative values and the rest of them are not significant.

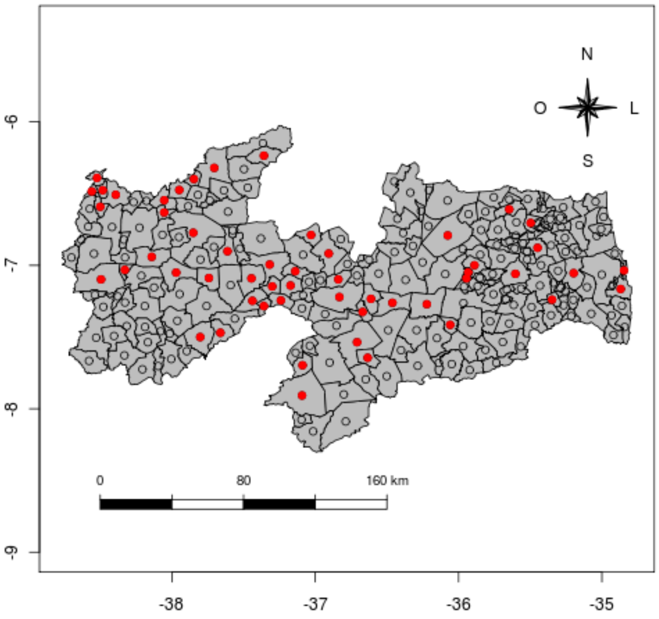


Figure 1. Scan satistic map of dengue fever for the state of Paraba in 2011.

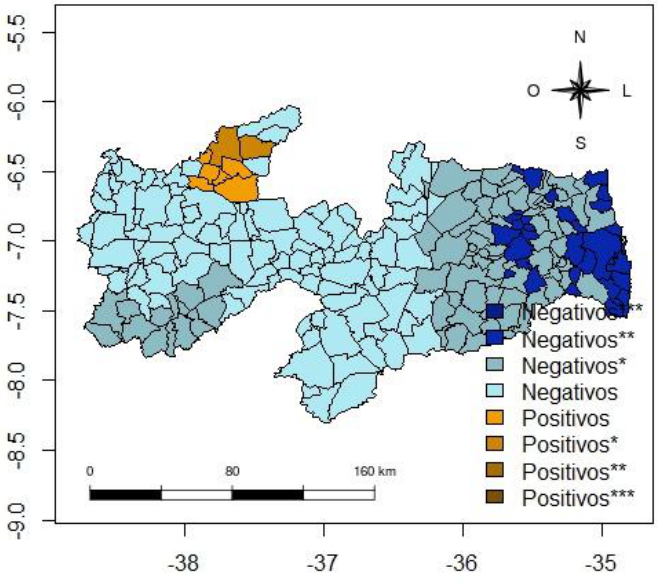


Figure 2. Getis and Ord map of dengue fever for the state of Paraba in 2011.

Comparing the final decision map (Figure 5) with the Besag and Newell map (Figure 3), it was observed that only 7 of them are on the Besag and Newell map. On the other hand, all significant cities on the Besag and Newell map are present on the final decision map.

Comparing the dengue final decision map (Figure 5) with the dengue risk map (Figure 4), it was verified that the cities on the final decision map present risk above 1.25, but not all cites with risk above 1.25 in the relative risk map are present on the final decision map. Finally, the result allows us to state that the methodology identified the cities with high relative risk of dengue fever in the state of Paraba in the year of 2011.

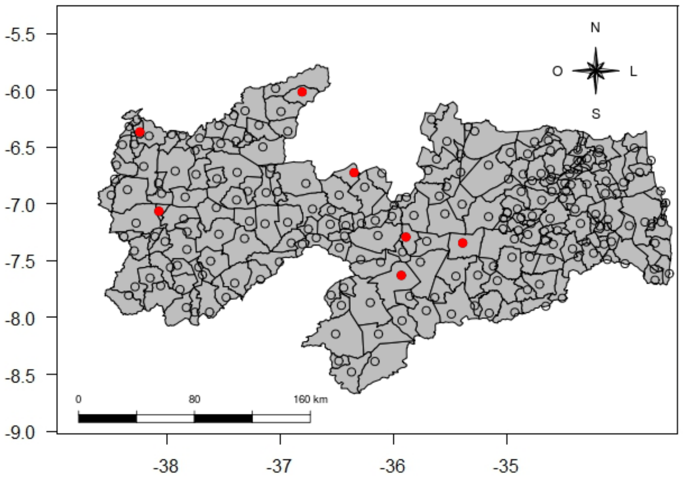


Figure 3. Besag e Newell map of dengue fever for the state of Paraba in 2011.

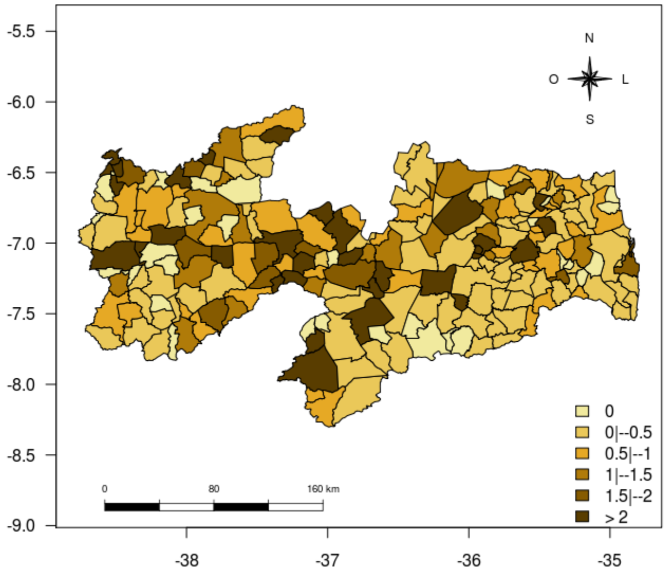


Figure 4. Map of the relative risk of dengue fever for the state of Paraba in 2011.

## V. CONCLUSIONS

In this paper, we presented a new methodology for the combination of spatial clustering methods. We also presented a rule for building that combination based on majority voting. A study case was presented using epidemiological data of dengue fever from state of Paraba, Brazil, in the year of 2011.

Based on the results achieved, it is possible to affirm that the combination of spatial clustering methods using the majority voting rule presented coherent results and those results were better than each individual classifier. At the end, the methodology identified the priority and non-priority regions for dengue fever in the state of Paraba, Brazil.

## VI. ACKNOWLEDGMENTS

This project is partially supported by CAPES. It is also partially supported by grant 310470/2012-9 of the National Coun-

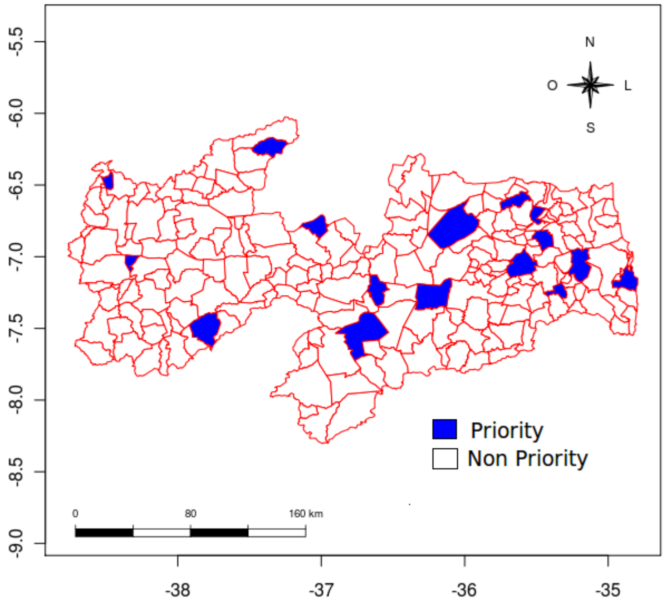


Figure 5. Decision map of the combination of spatial clustering methods for dengue fever in the state of Paraba in 2011.

cil for Scientific and Technological Development (CNPq) and is related to the National Institute of Science and Technology “Medicine Assisted by Scientific Computing”(181813/2010-6) also supported by CNPq.

## References

[1] A. Abrams, K. Kleiman, and M. Kulldorff, “Gumbel based p-value approximations for spatial scan statistics”, International Journal of Health Geographics, vol. 9, December 2010, pp. 1-12.

[2] A. Jain, R. Duin, J. Mao, and S. Member, “Statistical Pattern Recognition: A Review”, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 22, January 2000, pp. 4-37.

[3] L. Anselin, “Spatial data analysis with GIS: an introduction to application in the social sciences,” Fundamental Research in Geographic Information and Analysis, University of California, August 1992.

[4] J. Besag and J. Newell, “The detection of clusters in rare diseases,” Journal of the Royal Statistical Society, vol. 154, June 1991, pp. 143-155.

[5] F. Breve, M. Ponti, and N. Mascarenhas, “Combining Methods to Stabilize and Increase Performance of Neural Network-Based Classifiers,” in Proceding of the XVIII Brazilian Symposium on Computer Graphics and Image Processing, 2005.

[6] M. A. Costa and R. M. Assunção, “A fair comparison between the spatial scan and the Besag-Newell disease clustering test,” Environmental and Ecological Statistics, vol. 12, iss: 3, September 2005, pp. 301-319.

[7] T. Damoulas and A. M. Girolami, “Combining Feature Spaces for Classification,” Pattern Recognition, vol. 42, April 2009, pp. 2671-2683.

[8] R. P. W. Duin and D. M. J. Tax, “Experiments with Classifier Combining Rules,” in MCS ’00 Proceedings of the First International Workshop on Multiple Classifier Systems, vol.1857, June 2000, pp. 16-29.

[9] T. K. Ho, J. J. Hull, and S. N. Srihari, “Combination of Structural Classifiers,” in Proceedings of 1990 IAPR Workshop on Syntactic and Structural Pattern Recognition, 1990.

[10] J. J. Hull, A. Commike, and T. K. Ho, “Multiple Algorithms for Handwritten Character Recognition,” International Workshop on Frontiers in Handwriting Recognition, Montreal, Canada, April 2-3, 1990.

[11] H. Izakian and W. Pedrycz, “A new PSO-optimized geomatry of spatial and spatio-temporal scan statistics for disease outbreak detection,”, Swarn and Evolutionary Computation, vol.4, June 2012, pp. 1-11.

[12] G. M. Jacquez, The Handbook of Geographic Information Science. Blackwell Publishing Ltd, 2008.

[13] J. Kersten, “Simulateneous Feature Selection and Gaussian Mixture Model Estimation for Supervised Classification Problems,” Pattern Recognition, vol. 47, August 2014, pp. 2582-2595.

[14] J. Kittler, M. Hatef, R. P.W. Duin, and J. Mata, “On Combining Classifiers”, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 20, March 1998, pp. 226-239.

[15] M. Kulldorff and N. Nagarwalla, “Spatial Disease Clusters: Detection and Inference,” Statistics in Medicine, vol.14, 1995, pp. 799-810.

[16] L. I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms. John Wiley & Sons, Inc., Hodoken, New Jersey, 2004.

[17] Y. D. Lan and L. Gao, “ A new model combining multiple classifiers based on neural network,” Fourth International Conference on Emerging Intelligent Data and Web Tecnologies, September 2013, pp. 154-159.

[18] C. M. Bishop, Pattern Recognition and Machine Learning. Spring, London, 2006.

[19] C. Nadal, R. Legault, and C. Suen, “Complementary Algorithms for the Recognition of Totally Unstrained Handwritten Numerals,” in Proceedings of 10th International Conference on Pattern Recognition, vol. 1, June 1990, pp. 443-449.

[20] K. Rothman, Modern Epidemiology. The United States of America, Little, Brown and Company, 1986.

[21] C. Suen, C. Nadal, R. Mai, and L. Lam, “Recognition of Totally Unconstrained Handwritten Numerals Based on the Concept of Multiple Experts,” in Proceedings of International Workshop on Frontiers in Handwriting Recognition, Montreal, Canada, 1990.

[22] X. G. L. Tong, “Learning Opponent’s Indifference Curve in Multi-issue Negotiation Combining Rule Based Reasoning with Bayesian Classifier”, in in Proceedings of 6th International Conference on Natural Computation, vol. 6, August 2010, pp. 2916-2920.

[23] E. G. Knox, “Detection of clusters”. In: Methodology of enquiries into disease clustering. P. Elliot (ed.) Small Area Health Statistics Unit. London, 1989, pp. 17-20.

[24] R. M. Moraes and L. S. Machado, “A New Class of Assessment Methodologies in Medical Training Based on Combining Classifiers”, in Proceedings of XII Safety, Health and Environment World Congress (SHEWC’2012). 22-25 July, 2012, pp. 91-95.

[25] R. M. Moraes, J. A. Nogueira, and A. C. A. Sousa, “A New Architecture for a Spatio-Temporal Decision Support System for Epidemiological Purposes”, in Proceedings of the 11th International FLINS Conference on Decision Making and Soft Computing (FLINS2014), 17-20 Agosto, 2014, Brazil, pp. 17-23.

[26] Z-H. Zhou, “Ensemble Methods: Foundations and Algorithms”. CRC Press, Boca Raton, 2012.

# The Effect of 2nd-Order Shape Prediction on Tracking Non-Rigid Objects

Kenji Nishida
and Takumi Kobayashi

National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba, JAPAN
Email: kenji.nishida@aist.go.jp takumi.kobayashi@aist.go.jp

Jun Fujiki

Department of Applied Mathematics,
Fukuoka University
Fukuoka, JAPAN
Email: fujiki@fukuoka-u.ac.jp

***Abstract*—**

**For the tracking of non-rigid objects, we have previously proposed a shape-based predict-and-track algorithm. The method was built upon the similarity between the predicted and actual object shapes. The object shape was predicted from the movement of feature points, which were approximated by a second-order Taylor expansion. Approximate first-order movements, the so-called optical flows, were simultaneously exploited by chamfer matching of edgelets. In this paper, the effect of second-order shape prediction is quantitatively analyzed by tracking a non-rigid skier object. The method exhibits superior shape prediction performance compared to a simple linear prediction method.**

***Keywords*–*Tracking non-rigid objects, Chamfer distance, Shape prediction.***

## I. INTRODUCTION

Visual object tracking is one of the most popular techniques in the field of computer vision. Recently, tracking algorithms for non-rigid (deformable) objects have been used in many application fields [1], [2]. In sports scenes, especially those of team sports such as football, there are many similar objects, which increase the difficulty of tracking. Therefore, we consider both the movement and form (shape) of these objects to be discriminative for tracking.

We have already proposed a shape-based predict-and-track algorithm [3], which was evaluated by tracking a skier to determine the effect of shape prediction performance. In this paper, we quantitatively evaluate the shape prediction performance of a second-order shape prediction algorithm against linear (first-order) prediction. The performance is measured by the similarity between the predicted and actual shapes of the tracked object.

The remainder of this paper is organized as follows. In Section II, we describe our shape prediction algorithm and the tracking procedure that uses the chamfer distance as a similarity measure. The experimental results are presented in Section III. Finally, we present our conclusions and ideas for future work in Section IV.

## II. SHAPE-BASED PREDICT-AND-TRACK ALGORITHM

In this section, we describe an algorithm for tracking by shape prediction [3]. The algorithm consists of two components, shape prediction and tracking by shape similarity.

### A. *Notation*

The following notation is used throughout this paper.

- $X$ denotes the center of the object,
- $O(X)$ denotes the object image centered at position $X$,
- $E(X)$ denotes the binary edge image for the object at position $X$,
- $\hat{O}$ and $\hat{E}$ denote the predicted image and edge image of the object, respectively,
- $x$ denotes the positions of the feature points for object $X$,
- $x'$ denotes the differential of $x$, i.e., $x' = \frac{dx}{dt}$,
- $x''$ denotes $\frac{d^2x}{dt^2}$,
- $\tilde{x}$ denotes the subset of feature points in the object that constitute the outline edge, $\tilde{x} \in E(X)$,
- $\hat{x}$ denotes the predicted position at the next frame for $\tilde{x}$,
- $l(x)$ denotes the edgelet for position $x$.

### B. *Shape Prediction*

The object shape is represented by the collection of feature points $x$, and the deformation of the object is predicted by exploiting the movement of the feature points. Sim and Sundaraj proposed a motion tracking algorithm using optical flow [4], and this can be considered as the first-order approximation of the movement. For our tracking algorithm, we adopt a shape prediction method based on the second-order approximation of the feature points' movement [3].

Let $x_t$ be the 2-D position of the feature points that constitute the object image $O$ at time $t$. The position of the points at $t+1$ can be estimated using a Taylor expansion. Up to the second-order, this is

$$x_{t+1} = x_t + x'_t + \frac{1}{2}x''_t, \qquad (1)$$

where $x'$ is the so-called optical flow, which is practically computed as the difference in the pixel position:

$$x'_t = x_t - x_{t-1}. \qquad (2)$$

Similarly, $x''$ denotes the second-order differential of $x$, which is calculated as

$$\begin{aligned} x''_t &= x'_t - x'_{t-1} \\ &= x_t - x_{t-1} - (x_{t-1} - x_{t-2}) \\ &= x_t - 2x_{t-1} + x_{t-2}. \qquad (3) \end{aligned}$$

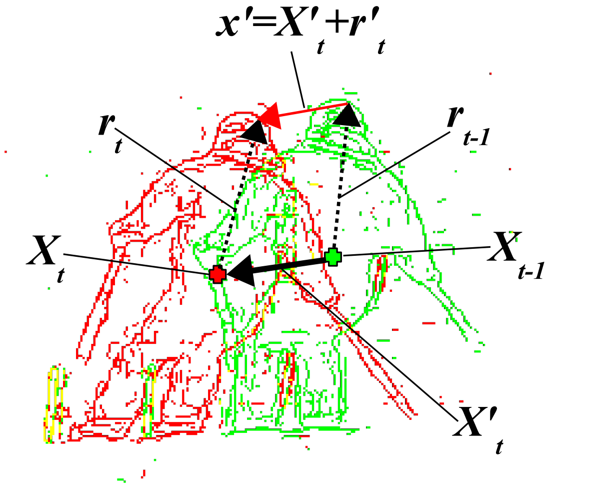
Therefore, the appearance of the object at $t+1$ can be predicted based on the optical flows computed from three consecutive video frames. Suppose that the shape of the object is determined by the outline edge image $E$, predicted from the feature point movements in previous video frames. The algorithm for detecting the feature point movements is described in Section II-D.

### *C. Estimation of Object Translation*

The movement of the feature points comprises both the object translation (global movement of the center of the object) and the movement of the pixels relative to the center of the object, which is described by

$$x'_t = X'_t + r'_t, \tag{4}$$

where $X$ denotes the position of the object's center, and $r$ denotes the position of the pixels relative to $X$. Figure 1 shows the movement of feature point $x'$, the movement of the object's center $X'$, and the relative movement $r'$.



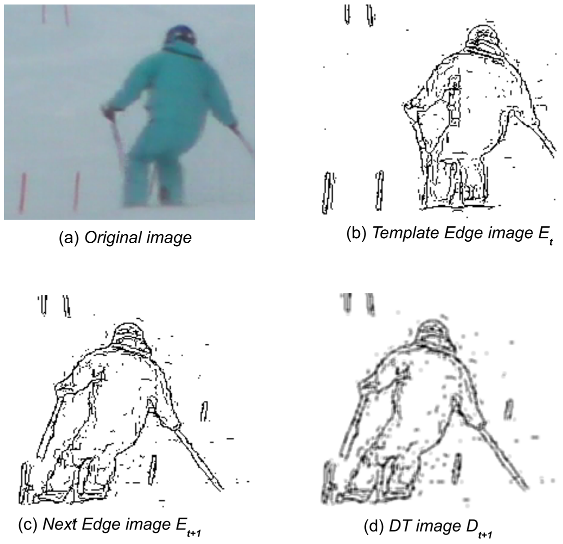
Green: Edge image for $t-1$, Red: Edge image for $t$

Figure 1. Edge image and object movement.

The relative movement $r'$ is derived from the object deformation, and thus makes a significant contribution to the prediction of the object's shape. Because relative movement obeys the physical constraints of the body parts of the object, its second-order prediction is effective. In contrast, the second-order movement contributes less to the object translation $X$, because such global movement contains the ego-motion of the camera as well as the real movement of the object. Therefore, the purpose of our tracking algorithm is to determine the next object position $X_{t+1}$ based on the similarity between the predicted and actual object shapes, which is computed globally.

The similarity between the predicted edge image $\hat{E}_{t+1}$ and actual edge image $E_{t+1}$ is measured using the chamfer system [5]. This system measures the similarity of two edge images using a distance transform (DT) methodology [6].

Let us consider the problem of measuring the similarity between template edge image $E_t$ (Figure 2(b)) and a successive edge image $E_{t+1}$ (Figure 2(c)). We apply the DT to obtain an image $D_{t+1}$ (Figure 2(d)), in which each pixel value $d_{t+1}$ denotes the distance to the nearest feature pixel in $E_{t+1}$. The chamfer distance $D_{chamfer}$ is defined as



(a) *Original image*

(b) *Template Edge image $E_t$*

(c) *Next Edge image $E_{t+1}$*

(d) *DT image $D_{t+1}$*

Figure 2. Chamfer system.

$$D_{chamfer}(E_t, E_{t+1}) = \frac{1}{|E_t|} \sum_{e \in E_t} d_{t+1}(e), \tag{5}$$

where $|E_t|$ denotes the number of feature points in $E_t$ and $e$ denotes a feature point of $E_t$.

The translation of the object can be estimated by finding the position of the predicted edge image $\hat{E}_{t+1}$ that minimizes $D_{chamfer}$ between $\hat{E}_{t+1}$ and the actual edge image $E_{t+1}$:

$$X_{t+1} = \arg\min_{E_{t+1}} D_{chamfer}(\hat{E}_{t+1}, E_{t+1}). \tag{6}$$

Figure 3 illustrates the tracking procedure. First, the optical flow $x'_t$ and its approximate derivative $x''_t$ are computed from preceding video frames at $t-2$, $t-1$, and $t$. The object shape at $t+1$, denoted by $\hat{E}_{t+1}$, is then predicted using $x'$ and $x''$. The object position is determined by locating $\hat{E}_{t+1}$ at the position of minimum chamfer distance to the actual shape at $t+1$, $E_{t+1}$. Finally, the optical flow for the next video frame $x'_{t+1}$ is recomputed using actual edge images $E_t$ and $E_{t+1}$.

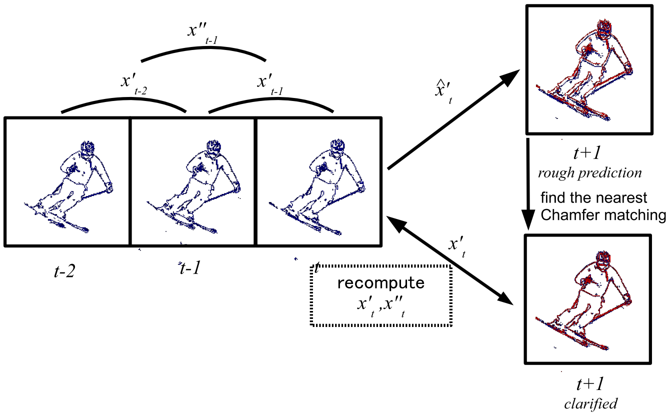
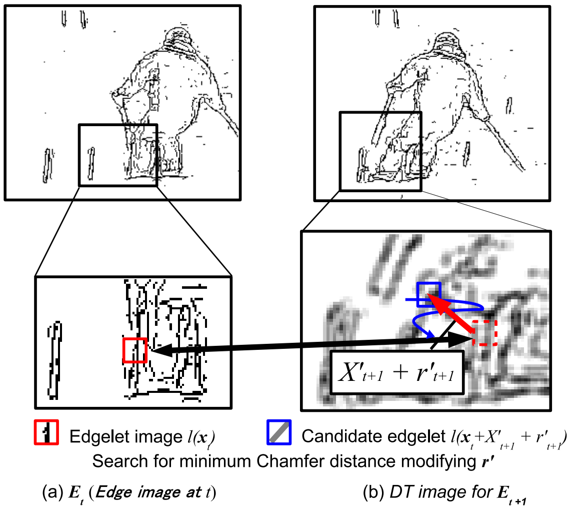


Figure 3. Tracking procedure.

### D. Detection of Feature Point Movements

After the object translation $X'_{t+1}$ has been determined, the movement of the feature points $x'_{t+1}$ is detected from the actual object images $O(X_t)$ and $O(X_{t+1})$.

The feature point movements $x'_{t+1}$ are directly computed based on the actual edge image at $t+1$ by tracking small parts of the edge (edgelets). We also employed the chamfer system to detect the movement of the edgelets. A template edgelet image $l(\tilde{x}_t)$ extracted from $E_t$ is compared against the candidate edgelet $l(\tilde{x}_t + \hat{x}'_{t+1})$ in the next edge image $E_{t+1}$. By minimizing the chamfer distance between the two, we obtain the feature point movement (Figure 4):


$X'_{t+1} + r'_{t+1}$
Edgelet image $l(x_t)$
Candidate edgelet $l(x_t+X'_{t+1} + r'_{t+1})$
Search for minimum Chamfer distance modifying $r'$

(a) $E_t$ (*Edge image at t*)

(b) *DT image for* $E_{t+1}$

Figure 4. Edgelet tracker.

$$\hat{x}'_{t+1} = \arg\min_{\hat{x}'_{t+1}} D_{chamfer}(l(\tilde{x}_t), l(\tilde{x}_t + \hat{x}'_{t+1})). \tag{7}$$

As the detected movements $\hat{x}'_{t+1}$ may contain noise, we apply a smoothing process by averaging the relative movements in the neighboring region:

$$x'_{t+1} = \frac{1}{N} \sum_{\hat{x}'_{t+1} \in \delta_{t+1}} \hat{x}'_{t+1}, \tag{8}$$

where $N$ denotes the number of detected movements $\hat{x}'_{t+1}$ in the neighborhood $\delta$ of $\tilde{x}_t$.

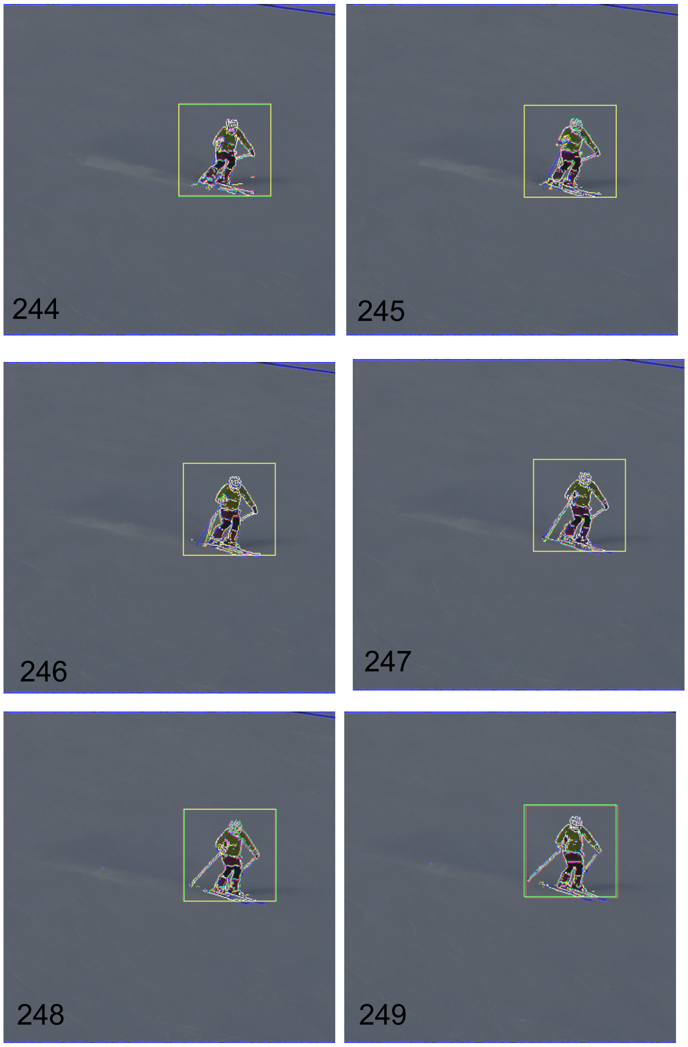
## III. Evaluation of Shape Prediction Performance

The algorithm described above was applied to a video sequence of a skier, captured by a hand-held camera, and the effect of camera ego-motion on the shape prediction performance was examined. The second-order shape prediction was compared with linear shape prediction, which is formulated by

$$x_{t+1} = x_t + x'_t. \tag{9}$$

The prediction performance was evaluated by the chamfer distance between the actual image $E_{t+1}$ and the predicted image $\hat{E}_{t+1}$.

In the skiing sequence captured by a hand-held camera, the skier was manually "tracked" so as to remain close to the center of the image frame. Thus, the object tends to exhibit only a small translation in the image frame. However, the object sometimes suffers from a large degree of translation due to manual mis-tracking of the camera. Figure 5 shows the tracking results. The blue pixels represents the predicted object shape, the green ones represent the translated predicted shape to determine the object position using (6), and the red ones represents the reconstructed object shape, as calculated by (1).



Blue: Ground Truth; Green: Linear Prediction;

Red: Second-order Prediction.

Figure 5. Tracking result.

Figure 6 shows the chamfer distance to the ground truth, calculated over frames 230–300. The results show that the second-order prediction attained better precision than the linear prediction in 40 out of 70 frames. The second-order prediction is superior during frames 244–249, whereas the linear prediction is preferable from frames 238–240.

Figure 7 shows the object translation from frames 244–248, indicating the direction change at around frame 246. Figure 8 shows the object translation from frames 238–240, when the translation direction did not change.

These results indicate that the second-order shape prediction method works well when the direction in which the object

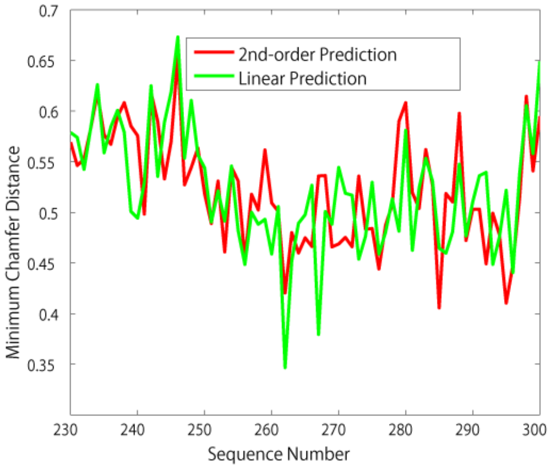
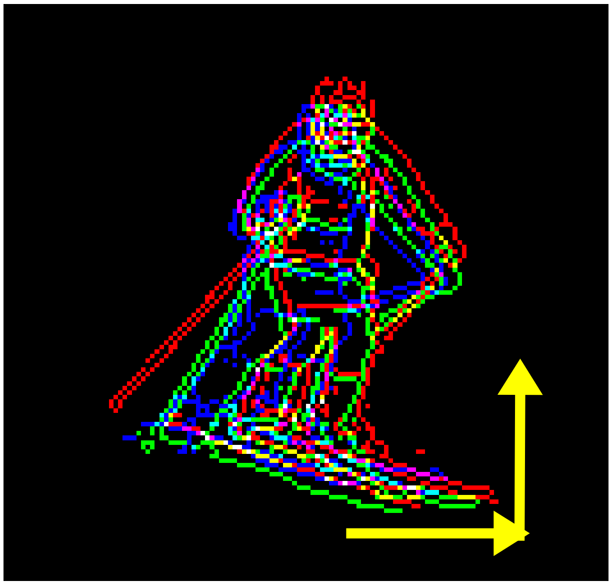


Figure 6. Shape precision according to chamfer distance.



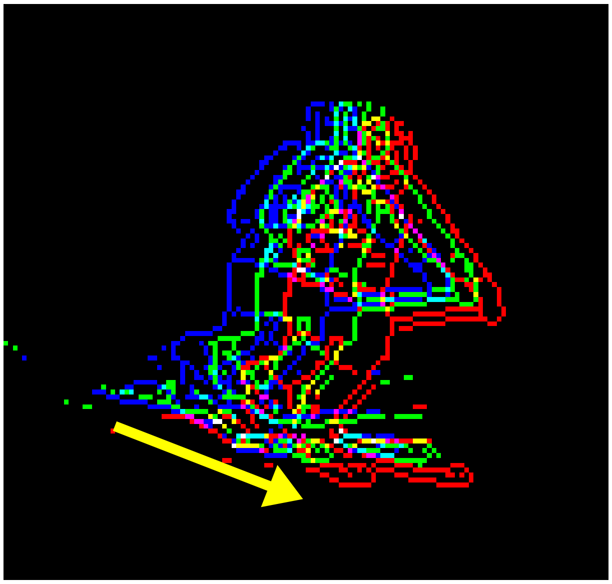
Blue: 244; Green: 246; Red: 248.
Yellow arrow: Object translation.

Figure 7. Object translation from frames 244–248.

must be translated changes.

## IV. Conclusions

We have evaluated the performance of a second-order shape prediction algorithm. Though the performance is generally similar to that of a linear model, our method outperformed the linear approach when the direction of object movement changed. This evaluation result indicates that the proposed second-order model is robust to objects under high acceleration. In future work, we will integrate the two models, allowing us to switch which model is applied for prediction, and apply the method to various types of objects besides skiers.



Blue: 238; Green: 239; Red: 240.
Yellow arrow: Object translation.

Figure 8. Object translation from frames 238–240.

## Acknowledgment

The authors would like to thank Dr. Akaho, group leader of Mathematical Neuroinformatics Group, for his valuable comments and suggestions. This work was supported by JSPS KAKENHI Grant Number 26330217.

## References

[1] G.Sundaramoorthi, A.Mennucci, S.Soatto, and A.Yezzi, “A New Geometric Metric in the Space of Curves, and Applications to Tracking Deforming Objects by Prediction and Filtering”, in SIAM j. of Imaging Science, Vol.4, No.1, 2010, pp.109-145.

[2] M. Godec, P.M Roth, and H.Bischof, “Hough-based Tracking on Non-rigid Objects”, J. of Computer Vision and Image Understanding, Vol.117, No.10, 2013, pp.1245-1256.

[3] K. Nishida, T. Kobayashi, and J. Fujiki, “Tracking by Shape with Deforming Prediction for Non-Rigid Objects”, in proc. International Conference on Pattern Recognition Applications and Methods (ICPRAM) , 2014, pp. 581-587.

[4] K.F. Sim, and K. Sundaraj,. “Human Motion Tracking of Athlete Using Optical Flow & Artificial Markers”, in Proc. International Conference on Intelligent and Automation Systems (ICIAS), 2010, pp. 1-4.

[5] D.M. Gavrila, “Pedestrian Detection from a Moving Vehicle”, in Proc. European Conference on Computer Vision (ECCV), 2009, pp. 37-49.

[6] D.Huttenlocher, G.Klanderman, and W.J.Rucklidge, “Comparing Images using the Hausdorff Distance”, in IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 15, No. 9, 1993, pp. 850-863.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-393-3

# Assessment of Fuzzy Gaussian Naive Bayes for Classification Tasks

Jodavid A. Ferreira, Elaine A. M. G. Soares, Liliane S. Machado and Ronei M. Moraes

Laboratory of Statistics Applied to Image Processing and Geoprocessing
Federal University of Paraiba
João Pessoa, Brazil
Email: jodavid.arts@gmail.com, elaine.soares@ci.ufpb.br,
liliane@di.ufpb.br, ronei@de.ufpb.br

***Abstract*—Statistical methods have been used in order to classify data from random samples. In general, if we know the statistical distribution of data, we can use specific classifiers designed for that distribution and expect good results. This work assesses the accuracy of a Fuzzy Gaussian Naive Bayes (FGNB) classifier for tasks using data from five different statistical distributions: Negative Binomial, Logistic, Log-Normal, Weibull and Gamma. The FGNB classifier was recently proposed as a fuzzy extension of Gaussian Naive Bayes for training assessment in virtual environments. Results of assessment are provided and show different accuracy according to the statistical distribution of data.**

***Keywords–Fuzzy Gaussian Naive Bayes Classifier, Classification, Accuracy Assessment.***

## I. Introduction

Statistical methods have been widely used in order to classify data from random samples [1]. In general, if we know the statistical distribution of data, we can use specific classifiers designed for that distribution and we can expected good results from that use [2]. Classifiers based on Gaussian distribution were exhaustively studied in the literature [3][4] and applied in several kinds of problems [5][6][7]. Some of their variations are known as Classical Bayes Rule [8] and Gaussian Naive Bayes [9].

However, in several kind of applications, it is not possible to affirm the sample data were measured with accuracy. In these cases, the imprecision on data should be incorporated in the classification method. Nowadays, a possible approach for this modelling is using fuzzy sets proposed by Lofti A. Zadeh [10]. Several classification methods based on fuzzy sets can be found in the literature and some of them are based on probability measures of fuzzy events [11]. Among them, the Fuzzy Gaussian Naive Bayes (FGNB) method was proposed by Moraes and Machado [12] and has been applied to classification and training assessment problems [13][14][15] achieving good results.

The main question about the classifiers based on Gaussian distribution is related to some classification problems in which data did not follow Gaussian distribution. So, it is interesting to know the limitations when those methods are used. This paper aims to verify if the FGNB method has good performance when classifying data given by the Logistic, Gamma, Weibull, Log-Normal and Negative Binomial distributions. For each statistical distribution were used data dimensions from 1 to 4.

The FGNB classifier was proposed recently; then it is necessary to know its accuracy. A preliminary performance analysis from these authors using FGNB classifier and other statistical distributions was performed [15]. In this paper, we enlarge the ¿ range of distributions used to verify the accuracy of the method. The results of those comparisons are analysed with respect to the better statistical distribution of data to be used for better FGNB performance, according to each dimension of data.

Section 2 presents the Fuzzy Gaussian Naive Bayes (FGNB) classification method which was used for data classification. In Section 3, the methodological part is described: the data used and how the samples were generated. In Section 4, the classification results for the 5 distributions statistics are detailed. The conclusion of the study, highlighting the distribution that was not well sorted, is in Section 5.

## II. FUZZY GAUSSIAN NAIVE BAYES (FGNB)

Formally, let the classes of performance in space of decision be $\Omega = \{1, ..., M\}$ where $M$ is the total number of classes. Let $X$ be a vector of training data, according to sample data D, where $X$ is a vector with $n$ distinct features, **i.e.,** $X = \{X_1, X_2, , X_n\}$ and $w_i$, $i \in \Omega$ is the class in space of decision for the vector $X$. So, the probability of the class $w_i$, given the vector $X$, can be estimated using the Bayes Theorem:

$$P(w_i|X) = \frac{P(X|w_i)P(w_i)}{P(X)} = \frac{[P(X_1, X_2, \ldots, X_n|w_i)P(w_i)]}{P(X)} \tag{1}$$

Let us assume a naive hypothesis, in which each feature $X_k$ is conditionally independent of every other feature $X_l$ , for all $k \neq l \leq n$. This hypothesis, though sometimes it is not exactly realistic, enables an easier calculation of (1). An advantage of that assumption is the robustness acquired by classifier that now can classify data for which it was not trained for [16]. So, unless a scale factor S, which depends on $X_1, X_2, \ldots, X_n$, the equation (1) can be expressed by:

$$P(w_i|X_1, X_2, \ldots, X_n) = \frac{P(w_i)}{S} \prod_{k=1}^{n} P(X_k|w_i) \tag{2}$$

A possible approach is to assume Gaussian distribution for $X$ and compute its parameters from $D$, i.e., mean vector and covariance matrix [17]. From equation (2) it is possible to use the logarithm function in order to simplify the exponential function in the Gaussian distribution formula and, consequently, to reduce computational complexity replacing multiplications by additions:

$$g(w_i, X_1, X_2, \ldots, X_n) = log[P(w_i|X_1, X_2, \ldots, X_n)] = \quad (3)$$
$$= log\frac{P(w_i)}{S} + \textstyle\sum_{k=1}^{n} log[P(X_k|w_i)]$$

where $g$ is the classification function.

At this point, it is assumed that random variables $X_1, X_2, \ldots, X_n$ are also fuzzy variables because we are going to use their membership functions $\mu_{wi}(X_k)$ for this calculus [18]. Then, based on probability of a fuzzy event [11], the **FGNB** is done by [12]:

$$g_f(w_i, X_1, X_2, \ldots, X_n) = log[P(w_i|X_1, X_2, \ldots, X_n)] =$$
$$log\frac{P(w_i)}{S_f} + \textstyle\sum_{k=1}^{n} log\mu_{wi}(X_k)P(X_k|w_i) \quad (4)$$

where $g_f$ is the new classification function and $S_f$is a new scale factor.

The necessary parameters to compute $P(X_k|w_i)$ and $\mu_{wi}(X_k)$ should be learned from sample data $D$. The better estimation for class of the vector $X$ can be obtained from the highest values of the classification function $g_f$. However, as $S_f$ is a scale factor, it is not necessary to compute it for this maximization process. Then:

$$X \in w_i logP(w_i) + \sum_{k=1}^{n} log[\mu_{wi}(X_k)P(X_k|w_i)] > \quad (5)$$
$$logP(w_j) + \textstyle\sum_{k=1}^{n} log[\mu_{wj}(X_k)P(X_k|w_j)]$$

is the classification rule for FGNB.

## III. ASSESSMENT METHODOLOGY

Several studies show that assessment methods present better results when they are applied with data from a particular statistical distribution. In general, each method can achieve better results when data follow some specific statistical distributions [14]. In a previous work, Moraes [14] studied the FGNB method for classification tasks using six different statistical distributions: Binomial, Continuous and Discrete Uniform, Exponential, Gaussian and Poisson [15]. However, since the FGNB is a recent method, its performance is not clear with this other five statistical distributions: Negative Binomial, Logistic, Log-Normal, Gamma and Weibull. In this paper, we use the Monte Carlo simulation [19] to investigate the behaviour of this method.

For our implementation, the samples were generated with Monte Carlo simulation for 1, 2, 3, 4 and dimensions for the five distributions in two different formats. One is used for training the FGNB method and the other for testing the method. Their settings obey the following rules:

a) Random training sample: used for the training of the method, this sample has 40000 observations for all the 4 classes.

b) Random test sample: after the training, the method used this to the assessment. This sample was composed by 30000 observations for each class, totaling 120000 observations.

The assessment method FGNB was implemented to all the variety of dimensions and the respective classification matrices can be stored in order to assess the accuracy of this methodology in practical applications. In particular, we had used FGNB as a kernel of an online assessment method of virtual reality simulators for training [12][13].

### A. SIMULATION

To use the method, random samples were generated for the 5 statistical distributions. The samples were generated in *software R* [20] using the following parameters for each distribution:

*1) Negative Binomial:* For the Binomial distribution denoted by $X \sim BN(p,k)$, 2 parameters are necessary for samples generation. The parameters used were:

TABLE I. PARAMETERS OF THE NEGATIVE BINOMIAL DISTRIBUTION.

| NEGATIVE BINOMIAL<br>$X \sim BN(p,k)$ | CLASS 1 | CLASS 2 | CLASS 3 | CLASS 4 |
|---|---|---|---|---|
| DIMENSION 1 | (0.4,10) | (0.4,30) | (0.2,30) | (0.4,130) |
| DIMENSION 2 | (0.6,10) | (0.3,30) | (0.4,20) | (0.5,140) |
| DIMENSION 3 | (0.4,30) | (0.4,10) | (0.4,130) | (0.3,47) |
| DIMENSION 4 | (0.3,10) | (0.4,80) | (70,0.5) | (0.4,130) |

*2) Logistic:* Samples were generated from the logistics distribution, using the following parameters:

TABLE II. PARAMETERS OF THE LOGISTIC DISTRIBUTION.

| LOGISTIC<br>$X \sim L(\mu,\sigma)$ | CLASS 1 | CLASS 2 | CLASS 3 | CLASS 4 |
|---|---|---|---|---|
| DIMENSION 1 | (0,2) | (20,2.5) | (43,2) | (60,3) |
| DIMENSION 2 | (13,3) | (60,4) | (35,2) | (90,4) |
| DIMENSION 3 | (20,3) | (40,2) | (108,4) | (72,4) |
| DIMENSION 4 | (79,5) | (6,3) | (110,2) | (40,4) |

*3) Log-Normal:* For the generation of Log-Normal distribution samples, the following parameters were used:

TABLE III. PARAMETERS OF THE LOG-NORMAL DISTRIBUTION.

| LOG-NORMAL<br>$(\mu,\sigma)$ | CLASS 1 | CLASS 2 | CLASS 3 | CLASS 4 |
|---|---|---|---|---|
| DIMENSION 1 | (2,0.2) | (3,0.3) | (3.8,0.3) | (4.5,0.2) |
| DIMENSION 2 | (2.8,0.2) | (2,0.3) | (4.7,0.2) | (3.7,0.3) |
| DIMENSION 3 | (2,0.3) | (2.65,0.2) | (3.7,0.3) | (4.5,0.2) |
| DIMENSION 4 | (4.2,0.2) | (3.5,0.1) | (2,0.4) | (3,0.3) |

*4) Gamma:* The gamma distribution has the following notation $X \sim Gamma(shape, scale)$ and the parameters used for generation of the samples were:

TABLE IV. PARAMETERS OF THE GAMMA DISTRIBUTION.

| GAMMA<br>$(shape, scale)$ | CLASS 1 | CLASS 2 | CLASS 3 | CLASS 4 |
|---|---|---|---|---|
| DIMENSION 1 | (20,0.25) | (40,0.25) | (60,0.25) | (90,0.25) |
| DIMENSION 2 | (12,1.0) | (32,1.0) | (65,1.0) | (110,1.0) |
| DIMENSION 3 | (50,0.33) | (80,0.33) | (120,0.33) | (170,0.33) |
| DIMENSION 4 | (80,0.17) | (130,0.17) | (190,0.17) | (250,0.17) |

*5) Weibull:* The Weibull distribution has two parameters called shaped parameter and scale parameter. The parameters used to generate the samples were:

TABLE V. PARAMETERS OF THE WEIBULL DISTRIBUTION.

| WEIBULL<br>$(shape, scale)$ | CLASS 1 | CLASS 2 | CLASS 3 | CLASS 4 |
|---|---|---|---|---|
| DIMENSION 1 | (50,5) | (100,10) | (150,20) | (200,20) |
| DIMENSION 2 | (50,5) | (100,10) | (150,20) | (200,20) |
| DIMENSION 3 | (50,5) | (15,20) | (100,10) | (200,20) |
| DIMENSION 4 | (200,20) | (50,5) | (150,20) | (100,10) |

### B. KAPPA COEFFICIENT

The Kappa Coefficient K proposed by Cohen [21] is a robust pondered measure which takes into account agreements and disagreements between two sources of information from a classification matrix [22]:

$$K = \frac{(P_0 - P_c)}{(1 - P_c)} \tag{6}$$

where: $P_0 = \frac{\sum_{i=1}^{M} n_{ii}}{N}$ and $P_c = \frac{\sum_{i=1}^{M} n_{i+}n_{+i}}{N^2}$, where $n_{ii}$ is the total of main diagonal in the classification matrix; $n_{i+}$ is the total of line i in this matrix, $n_{+i}$ is the total of colum in the same matrix, M is the total number of classes and N is the total number of possible decisions presented in the matrix.

The variance of Kappa Coefficient K, denoted by $\sigma_K^2$ , is done by:

$$\sigma_K^2 = \theta_1 + \theta_2 + \theta_3. \tag{7}$$

where $\theta_1$ , $\theta_2$ , and $\theta_3$ are given by:

$$\theta_1 = \frac{P_0(1 - P_0)}{N(1 - P_c)^2} \tag{8}$$

$$\theta_2 = \frac{2(1 - P_0) + 2P_0P_c - \theta_4}{N(1 - P_c)^3} \tag{9}$$

$$\theta_3 = \frac{(1 - P_0)^2\theta_5 - 4P_c^2}{N(1 - P_c)^4} \tag{10}$$

and the parameters $\theta_4$ and $\theta_5$ are done by:

$$\theta_4 = \frac{\sum_{i=1}^{M} n_{ii}(n_{i+} + n_{+i})}{N^2} \tag{11}$$

$$\theta_5 = \frac{\sum_{i=1}^{M} n_{ii}(n_{i+} + n_{+i})^2}{N^3} \tag{12}$$

An approximation to the first component of 7 can be used for calculations. However, in this paper, we used the complete formula for variance and the Kappa Coefficient was computed for the best results. This coefficient is widely used in the literature of pattern classification [2].

According to Landis and Koch, the Kappa coefficient can be interpreted as presented in Table VI [23]. By these interpretation it is possible to distinguish where lies the classification data.

TABLE VI. CLASSIFICATION OF KAPPA COEFFICIENT.

| Kappa Statistic | Strength of Agreement |
|---|---|
| <0.00 | Poor |
| 0.00-0.20 | Slight |
| 0.21-0.40 | Fair |
| 0.42-0.60 | Moderate |
| 0.61-0.80 | Substantial |
| 0.81-1.00 | Almost Perfect |

## IV. RESULTS

### A. Negative Binomial Distribution

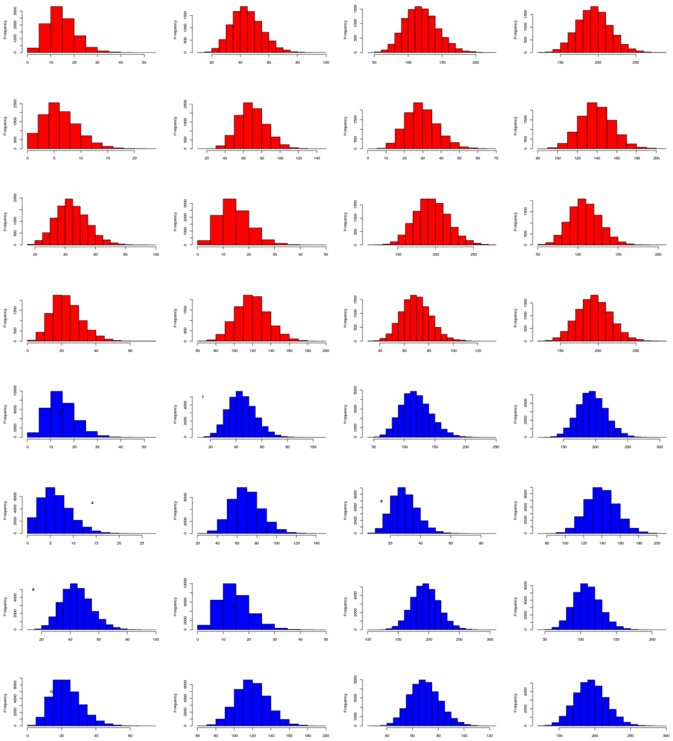


Figure 1. Random numbers generated for Negative Binomial distribution four sets of training and test samples (lines) for four classes (columns).

The negative binomial distribution is a discrete distribution, in which are considered some conditions: the experiment consists on an undetermined amount of repeated attempts, the probability of success is the same in each trial and the trials are independent. Using the method FGNB on Negative Binomial distribution with one dimension, was obtained a percentage accuracy of 69.91% and Kappa coefficient of 59.88% with a variance of $9.29 \times 10^{-6}$ . To dimension of size 2, the percentage of correct answers was 93.94%, the Kappa cefficient 91.92% and variance $2.53 \times 10^{-6}$. With the dimension equal to 3, the percentage accuracy and Kappa coefficient were greater than

99% and the variance obtained was $2.62 \times 10^{-7}$. The best results were obtained with the data dimension equal to 4: the percentage of right classifications was 99.9%, the Kappa 99.8% and the coefficient variance $6.10 \times 10^{-8}$. The histograms of the samples for this distribution are available in Figure 1, where the red and blue ones represent the training sample and the test samples, respectively. In each histogram, we represent the dimensions on the lines and the classes on the columns.

### B. Logistic Distribution

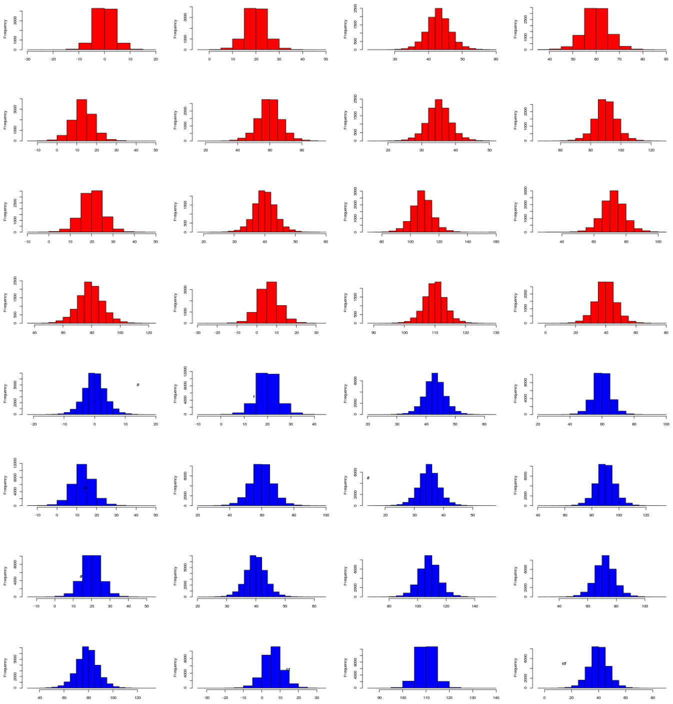


Figure 2. Random numbers generated for Logistic distribution four sets of training and test samples (lines) for four classes (columns).

The logistics distribution is a continuous distribution used in studies of population growth and agricultural production. It is also used in replacement of normal distribution due to the shape similarity of them in some specific studies. Figure 2 shows samples from logistics distribution used in this study. For a dimension equal to 1, the method proved to be more efficient for the logistic distribution than to the negative binomial distribution. The percentage accuracy was 84.41% and the Kappa coefficient was 79.22% with a variance of $1.94 \times 10^{-6}$. With dimension 2, the percentage of right classifications was 98.49%, with Kappa coefficient 97.99% and variance $2.19 \times 10^{-7}$.

The percentage of right classifications reached 99.9% when the point size was 3 under these conditions the Kappa coefficient was 99.5% and the variance $1.64 \times 10^{-8}$. For a dimension equal to 4, the method for distribution logistics FGNB achieved 99.9% of right classifications with a Kappa coefficient of 99.7% and variance $3.45 \times 10^{-9}$.

### C. Log-Normal Distribution

The Log-Normal distribution is continuous and can be used to feature the lifetime of products and materials (semiconductors, diodes and electrical insulation, among others). The histograms of random numbers generated for simulations using Log-Normal distribution are presented in Figure 3. In a simulation with only one dimensional data, the Kappa coefficient was 67.69% with variance $2.71 \times 10^{-}6$ and 29074 misclassifications. With two dimensions was obtained Kappa equal to 83.35% with $1.61 \times 10^{-}6$ of variance. When using 3 dimensions, the results were 84.62% and $1.50 \times 10^{-}6$ for Kappa and its variation. And with 4 dimensions, 13513 misclassifications occurred and the kappa was 84.98% with $1.47 \times 10^{-}6$.

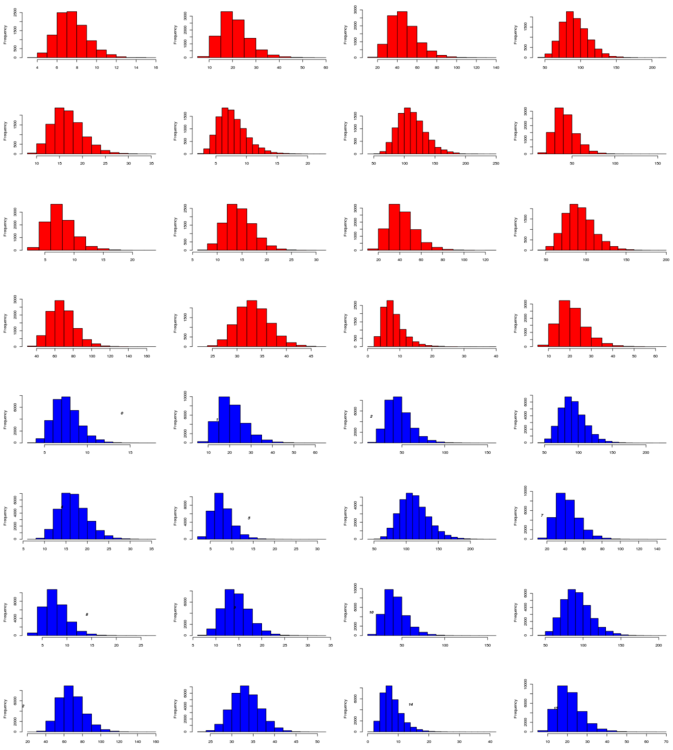


Figure 3. Random numbers generated for Log-Normal distribution four sets of training and test samples (lines) for four classes (columns).

### D. Gamma Distribution

The Gamma distribution is a continuous probability distribution, which has two parameters, the first one for shape and the second one for scale, and it requires that both parameters are greater than zero. The use of the FGNB method on the samples of the Gamma distribution that are present in Figure 4 produced the following results: 1 dimension, the percentage of right classifications was 72.33%, with a Kappa coefficient of 63.11% and variance of $2.92 \times 10^{-06}$. For dimension 2, the percentage of right classifications was 87.96%, with Kappa coefficient 83.96% and variance $1.57 \times 10^{-06}$. The percentage accuracy, for 3 dimensions, was greater than 97%, the Kappa coefficient 97.06% and the variance $3.199 \times 10^{-07}$. With dimension equal to 4, the kappa coefficient was 99.8% with a variance of $1.89 \times 10^{-08}$ and the percentage of right classifications 99.87% with 153 misclassifications.

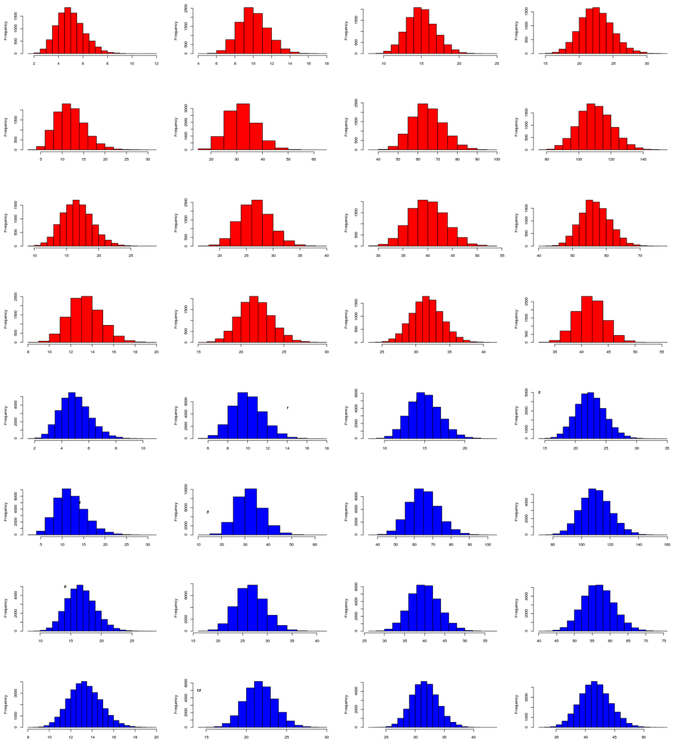


Figure 4. Random numbers generated for Gamma distribution four sets of training and test samples (lines) for four classes (columns).

### E. Weibull Distribution

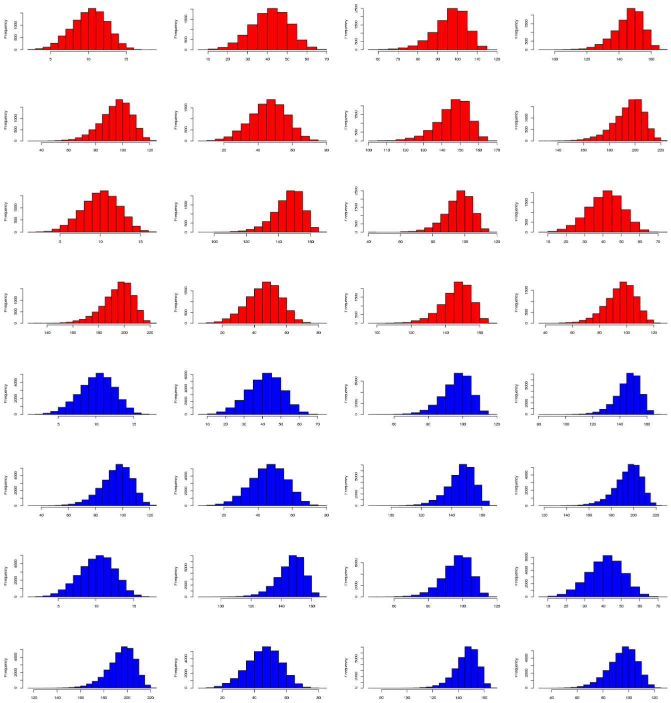


Figure 5. Random numbers generated for Weibull distribution four sets of training and test samples (lines) for four classes (columns).

The Weibull distribution is an important tool in the analysis of reliability and durability of equipment, such as resistance to fracture of the glass and flaws in semiconductors and capacitors. The histograms of random numbers generated for simulations using Continuous Uniform distribution are presented in Figure 5. In a simulation with one dimensional data and continuous uniform distribution, the best Kappa result was 66.72% with variance of $2.67 \times 10^{-6}$. The number of misclassifications was 29952. When dimension of data was increased for 2, the Kappa coefficient resulted in 99.93% with variance of $6.78 \times 10^{-9}$, and 55 misclassifications. For three dimensional data, the results were 99.48% and $5.74 \times 10^{-8}$ for Kappa and its variance, respectively. For last, the Kappa coefficiente pointed out 99.98% with variance of $1.48 \times 10^{-9}$ for 4 dimensional data.

TABLE VII. SUMMARY OF BEST RESULTS, BY STATISTICAL DISTRIBUTION, ACCORDING TO THE KAPPA COEFFICIENT

.

| Statistical Distribution | Number of Dimensions | Kappa Coefficient |
|---|---|---|
| Negative Binomial | 2 or more | $>$ 90.0 % |
| Logistic | 2 or more | $>$ 95.0 % |
| Log-Normal | 2 or more | $>$ 80.0 % |
| Gamma | 3 or more | $>$ 95.0 % |
| Weibull | 2 or more | $>$ 99.9 % |

Table VII presents a summary of the best results obtained by each statistical distribution in the simulations. For the Negative Binomial and Logistic distributions used in the FGNB classifier with three or more dimensions, was possible to achieve more than 99% of correct classification, according to the Kappa Coefficient. In a similar way, using the Gamma distribution with four dimensions, the FGNB classifier achieve more than 99% of accuracy. However, for Log-Normal distribution, the FGNB performance is reasonable, but its results are better in higher dimensions of data. The classification of the Weibull distribution presented excellent results. From the dimension two, the Kappa coefficient obtained was above 99.9%.

## V. CONCLUSION AND FUTURE WORK

In this paper, we presented an assessment of FGNB accuracy for classification tasks using data with different statistical distributions. We made simulations with five different statistical distributions: Negative Binomial, Logistic, Log-Normal, Weibull and Gamma. For each statistical distribution were analysed four different dimensions according to number of misclassifications, Kappa Coefficient and its variance. According to the results obtained, FGNB could be recommended to classify data from all distributions studied **in this paper**. For the distributions Negative Binomial, Logistic, Weibull and Gamma, the Kappa coefficient exceeded 90%. For the Log-Normal distribution, with maximum possible dimensions, the Kappa coefficient reached approximately 85%.

As future work, we would like to analyse the performance of this method with a database where each dimension can be given by a different statistical distribution. For instance, in a case of three dimensions, the first one would be a sample of Weibull distribution, the second one would be a sample of Gamma distribution and the last one would be a sample of Logistic distribution.

## VI. ACKNOWLEDGMENTS

This project is partially supported by grants 310561/2012-4 and 310470/2012-9 of the National Council for Scientific and Technological Development (CNPq) and is related to the National Institute of Science and Technology “Medicine Assisted by Scientific Computing”(181813/2010-6) also supported by CNPq.

## REFERENCES

[1] A. R. Webb and K. D. Copsey, Statistical Pattern Recognition., 3rd ed. Chichester: Wiley, 2011.

[2] R. O. Duda, P. E. Hart, and D. G.Stork, Pattern Classification, 2nd ed. Wiley Interscience, 2000.

[3] P. Domingos and M. Pazzani, “On the optimality of the simple bayesian classifier under zero-one loss.” Machine Learning, no. 29, 1997, pp. 103–130.

[4] G. John and P. Langley, “Estimating continuous distributionsin bayesian classifiers.” in 11th Conference on Uncertainy in Artificial Intelligence, Montreal, Canada, 1995, pp. 338–345.

[5] J. Hilden, “Statistical diagnosis based on conditional independence does not require it.” Computers in Biology and Medicine, 14, 429–435, 1984.

[6] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,” Machine Learning, no. 29, 1997, pp. 131–163.

[7] S. Monti and G. F. Cooper, “A bayesian network classifier that combines a finite mixture model and a naive bayes in model.” in 15th Conference on Uncertainy in Artificial Intelligence,, Stockholm, Sweden, 1999.

[8] C. M. Bishop, Pattern Recognition and Machine Learning., 1st ed. Singapure: Springer, 2007.

[9] R. M. Moraes and L. S. Machado, “Gaussian naive bayes for online training assessment in virtual reality-based simulators.” Mathware & Soft Computing, no. 16, 2009, pp. 123–132.

[10] L. A. Zadeh, “Fuzzy sets,” Information Control, no. 8, 1965, pp. 338–353.

[11] L. A. Zadeh., “Probability measures of fuzzy events,” J. Math. Anal. Applic., no. 10, 1968, pp. 421–427.

[12] R. M. Moraes and L. S. Machado, “Fuzzy gaussian naive bayes applied to online assessment in virtual reality simulators.” World Scientific, 2010, pp. 243–248.

[13] R. M. Moraes and L. S. Machado., “Online assessment in medical simulators based on virtual reality using fuzzy gaussian naive bayes,” Journal of Multiple-Valued Logic and Soft Computing, no. 18, 2012, pp. 479–492.

[14] R. M. Moraes, “Performance analysis of evolving fuzzy neural networks for pattern recognition,” Mathware & Soft Computing, no. 20, 2013, pp. 63–69.

[15] J. A. Ferreira, E. Soares, and R. M. Moraes, “Assessment of fuzzy gaussian naive bayes classifier using data with different statistical distributions,” in III Congresso Brasileiro de Sistemas Fuzzy (CBSF2014), Joao Pessoa, Brazil, August 2014.

[16] M. Ramonia and P. Sebastiani, “Robust bayes classifiers,” Artificial Intelligence, vol. 125, January 2001, pp. 209–226.

[17] R. A. Johnson and D. W. Wichern, Applied Multivariate Statistical Analysis, 4th ed. Prentice Hall, 1998.

[18] G. J. Klir and B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall, 1995.

[19] J. Gentle, Elements of Computational Statistics. Springer, 2005.

[20] R Development Core Team. R: A language and environment for statistical computing., R Foundation for Statistical Computing, 2009.

[21] J. Cohen, “A coefficient of agreement for nominal scales,” Educational and Psychological, no. 20, 1960, pp. 37–46.

[22] R. M. Moraes and L. S. Machado, “Psychomotor skills assessment in medical training based on virtual reality using a weighted possibilistic approach,” Knowledge-Based Systems, no. 70, 2014, pp. 97–102.

[23] J. R. Landis and G. G. Koch, “The measurement of observer agreement for categorical data.” Biometrics, vol. 33, no. 1, 1977, pp. 159–174.