DSIE

Doctoral Symposium in Informatics Engineering

12

www.fe.up.pt / dsie12

Editors Eugénio Oliveira | Gabriel David | A. Augusto Sousa

PROCEEDINGS OF THE 7TH DOCTORAL SYMPOSIUM IN INFORMATICS ENGINEERING

DEI Departamento de Engenharia Informática

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

U.PORTO

universidade de aveiro

Universidade do Minho

DEI - FEUP - PORTO - PORTUGAL 26 | 27 January 2012

Copyright

1st Edition 2012
ISBN: 978-972-752-141-8
Editors: Eugénio Oliveira, Gabriel David and António Augusto Sousa
Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, 4200-465 Porto

DSIE'12 Secretariat

Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, s/n
4200-465 Porto, Portugal
Telephone: +351 22 508 21 34
Fax: +351 22 508 14 43

E-mail: dsie12@fe.up.pt
website: http://www.fe.up.pt/dsie12

Preface

2012 Doctoral Symposium in Informatics Engineering – DSIE'12 is an event representing the 7th edition of a scientific meeting usually organized by PhD students of the FEUP Doctoral Program in Informatics Engineering (ProDEI). However, the current edition is co-organized with PhD students from MAPi Informatics Doctoral Program. This fact brought up expectations on both quantity and quality increase of the submitted papers.

This kind of meetings have been held since the school year 2005/06 and the main goal has always been to provide a forum for discussion on, and demonstration of, the practical application of a variety of scientific research issues, particularly in the context of information technology, computer science and computer engineering. DSIE symposium comes out as a natural conclusion of mandatory ProDEI and, this year, MAPi courses called, respectively, "Methodologies for Scientific Research" (MSR) and "Seminar", leading to a formal evaluation of the students learned competencies.

The aim of those specific courses (MSR and Seminar) is to teach students the processes, methodologies and best practices related to scientific research, particularly in the mentioned areas, as well as to improve their capability to produce adequate scientific texts. With a mixed format based on multidisciplinary seminars and tutorials, the course culminates with the realization of DSIE meeting, seen as a kind of laboratory test for the concepts learned by students. In the scope of DSIE, students are expected to play various roles, such as paper's authors, both scientific and organization committee members as well as reviewers, duly accompanied by more senior lectures and professors.

DSIE is then seen as a "leitmotif" for the students to write scientific correct and adequate papers following the methods and good practices currently associated to outstanding research activities in the area. Even though some of the papers are still at an embryonic stage and lack some maturity and even though some just report a state of the art, we already can find some interesting research work or interesting perspectives about future work. At this time, it was not essential, nor even possible, for most of the students in their PhD first year, to produce strong and deep research results. However, we hope that the basic requirements for presenting an acceptable scientific paper have been fulfilled.

DSIE'12 Proceedings include 35 articles accepted in the previously defined context. They cover a large spectrum of topics in computer science and engineering areas and can be grouped according to eight main clusters. These clusters include some different, although related, topics since, as it was expected, the paper themes are of a large diversity. These clusters are named as follows: Data Mining (7 papers), HM Interaction (5 papers), Image Processing and Computer Graphics(5 papers), Information Retrieval (4 papers), Data Management (3 papers), Distributed Computing and Protocols (3 papers), Mobile and Ubiquitous Computing (3 papers) and other isolated topics (4 papers). The complete DSIE'12 meeting encompasses a two days program that includes also two invited talks by outstanding researchers in Formal Methods for Programming and artificial life, autonomous robots and self-organization in multi-agent systems.

Professors responsible for both ProDEI and MAPi programs current edition are proud to participate in DSIE'12 meeting and would like to acknowledge all the students who have been deeply involved in the success of this event that, we hope, will contribute for a better understanding of the themes that have been addressed during the above referred courses, the best scientific research methods and the good practices for writing scientific papers and conveying novel ideas.

January 2012

Eugénio Oliveira and António Augusto Sousa (ProDEI)
Gabriel David (MAPi)

Conference Committees

Steering Committee

Eugénio Oliveira, PhD, FEUP
António Augusto Sousa, PhD, FEUP

Organizing Committee Co-Chairs

Nuno Cardoso, M.Sc.
Tiago Boldt Sousa, M.Sc.

Organizing Committee

Abbas Abdolmaleki, M.Sc.
António Ferreira, M.Sc.
Inês Coimbra Morgado, M.Sc.
João Vinagre, M.Sc.
Mário Cordeiro, M.Sc.
Pedro Ferreira, M.Sc.
Waldir Pimenta, M.Sc.
André Riboira, M.Sc.
Hugo Barbosa, M.Sc.
João Carneiro, M.Sc.
José Serra, M.Sc.
Nuno Silva, M.Sc.
Tiago Marques, M.Sc.

Scientific Committee Co-Chairs

José Luís Pereira, M.Sc.
Pedro Nogueira, M.Sc.

Scientific Committee

Abbas Abdolmaleki, M.Sc.
André Riboira, M.Sc.
António Ferreira, M.Sc.
Diogo Pratas, M.Sc.
Hadi Fanaee, M.Sc.
Helder Fontes, M.Sc.
Inês Coimbra Morgado, M.Sc.
Nuno Cardoso, M.Sc.
João Vinagre, M.Sc.
José Serra, M.Sc.
Luís Bastião, M.Sc.
Mário Antunes, M.Sc.
Mushtaq Raza, M.Sc.
Pedro Ferreira, M.Sc.
Pedro Silva, M.Sc.
Ricardo Gonçalves, M.Sc.
Sameh Eisa, M.Sc.
Tiago Boldt Sousa, M.Sc.
Waldir Pimenta, M.Sc.
Alina Trifan, M.Sc.
Anisa Allahdadi, M.Sc.
Cláudio Sá, M.Sc.
Eduardo Brito, M.Sc.
Hamid Arabnejad, M.Sc.
Hugo Barbosa, M.Sc.
João Carneiro, M.Sc.
João Rocha da Silva, M.Sc.
José Luís Pereira, M.Sc.
Klaus Schaefers, M.Sc.
Madiha Sher, M.Sc.
Mário Cordeiro, M.Sc.
Nuno Silva, M.Sc.
Pedro Nogueira, M.Sc.
Pedro M. Teixeira, M.Sc.
Ricardo Nobre, M.Sc.
Sara Fernandes, M.Sc.
Tiago Marques, M.Sc.

Affective/Ubiquitous Computing
Ricardo Morla, PhD, FEUP

Human-Computer Interaction
António Coelho, PhD, FEUP
José Creissac Campos, PhD, UMinho
Teresa Galvão, PhD, FEUP
Miguel Coimbra, PhD, FCUP
Rui Rodrigues, PhD, FEUP
Verónica Orvalho, PhD, FCUP

Information Retrieval
Alípio Mário Guedes Jorge, PhD, FCUP
Orlando Belo, PhD, UMinho
João José Ferreira, PhD, FEUP

Information Systems
Falcão e Cunha, PhD, FEUP

Intelligent Systems
Henrique Lopes Cardoso, PhD, FEUP
Michel Ferreira, PhD, FEUP
Paulo Novais, PhD, UMinho
Luis Paulo Reis, PhD, FEUP
Nuno Lau, PhD, UAveiro
Rosaldo Rossetti, PhD, FEUP

Multimedia & Image Processing
António Neves, PhD, UAveiro
Jaime Cardoso, PhD, FEUP
Luis Paulo Santos, PhD, UMinho
António Ramires Fernandes, PhD, UMinho
Jorge Barbosa, PhD, FEUP

Software Engineering
Ademar Aguiar, PhD, FEUP
Hugo Sereno Ferreira, PhD, FEUP
João Pascoal Faria, PhD, FEUP
Ana Cristina Paiva, PhD, FEUP
João Miguel Fernandes, PhD, UMinho
Manuel Alcino Cunha, PhD, UMinho

System Architectures
Pedro Brandão, PhD, FCUP
Rui Maranhão, PhD, FEUP

Sponsoring Institutions

DEI Departamento de Engenharia Informática

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

U.PORTO

universidade de aveiro

Universidade do Minho

Sponsors

Delta CAFÉS

efacec

glintt

inova inova+ | inovamais S.A.

labs.sapo/up

Vitalis ÁGUA MINERAL NATURAL

DISTINCTIVE PORT DALVA EST. 1862

Caixa Geral de Depositos

Millennium bcp

Table of Contents

Invited Speakers

Session I - Human Computer Interaction

Session II - Information Systems

Session III - System Architectures

Session IV - Information Retrieval

Session V - Intelligent Systems

Session VI - Multimedia & Image Processing

Session VII - Software Engineering

Session VIII - Affective Computing & Ubiquitous Computing

Talk: Taking a PhD in Informatics Engineering, a biased perspective

A PhD in Informatics Engineering (IE) has many aspects in common with any other area but also some specialities regarding the scientific area of IE. In this talk we will analyse both aspects from a personal perspective, built on personal experience and long time observation of different cases. The talk will approach the theme considering the student's side, the supervisor's side and the institutional and social environment. We will isolate a few essential aspects that a IE PhD student must take into account when pursuing his research work. With this we expect to challenge some made ideas and to foster discussion in the symposium.

About the speaker

Luis Correia (http://www.di.fc.ul.pt/˜lcorreia) was born in Lisboa, Portugal, in 1959. He obtained an Electrical Engineering degree (5 year degree) in 1982, from Instituto Superior Técnico of Universidade Técnica de Lisboa, and a PhD in Informatics (Behaviour based mobile robots) from Universidade Nova de Lisboa, in 1995. In 2009 he got habilitation from Universidade de Lisboa. From 1982 to 2003 he was practicing assistant, assistant and assistant professor at Universidade Nova de Lisboa. Since 2003 he is associate professor at Universidade de Lisboa. From 2004 on he has been leading LabMAg research centre. His research interests are in the area of artificial life, autonomous robots and self-organisation in multi-agent systems. He has participated in several research projects, four of them funded by the EU, with a leading or co-leading role of the national team in two of them. He has supervised 5PhD and 17 MSc thesis and wrote around 100 scientific articles, with revision, for journals and conferences. He is member of the editorial review board of the International Journal of Natural Computing Research (IJNCR) and of the advisory board of EPIA and Editorial Polimetrica. Regularly participates in program committees of conferences in the area of artificial life and evolutionary computation.

Talk: Towards a Linear Algebra of Programming

Besides the development of metrics for software quality, there is a trend towards quantitative methods in the software sciences which includes probabilistic programming and probabilistic reasoning. Fault tolerance and fault injection can also be addressed quantitatively. In this talk we address the foundations of probabilistic functional programming. Probabilistic functions are half way between relations and functions: they express the propensity, or likelihood of ambiguous outputs. A basis for a Linear Algebra of Programming (LAoP) enabling probabilistic reasoning is put forward as extension to the standard algebra of programming (AoP) based on relations. We show that, if one restricts to discrete probability spaces, categories of matrices provide adequate support for the extension, while preserving the reasoning style typical of standard program calculation.

About the speaker

José Nuno Oliveira (http://www.di.uminho.pt/~jno) is Associate Professor with Habilitation in Computer Science at the U. Minho. He graduated from the U.Porto in Electrical Engineering and received his MSc and PhD in Computer Science from the U.Manchester (UK), where he became interested in formal methods. He is a member of the scientific group IFIP WG2.1 - Algorithmic Languages and Calculi and of the Formal Methods Europe (FME) association, where he convenes a subgroup on education. Since his PhD work on data-flow program transformation, he has lectured and published on calculational techniques in software design for nearly 30 years. He has served the programme committee of many conferences in the series of FME symposia, MPC, TFM, SEFM, SBMF, ICTAC, ESOP, etc. He was chair of MPC'00, FME'01 and TFM'09.

Session I

Human Computer Interaction

Chairman: Eduardo Brito

Length of Text Line on the Bases of Eye Blink to Reduce Maximum Focus Losses
Mushtaq Raza

Workload and Situational Awareness management in UAV teams through interface modelling
António Sérgio Ferreira

Length of Text Line on the Bases of Eye Blink to Reduce Maximum Focus Losses

Mushtaq Raza
MAP-i Doctoral Student, University of Porto, Porto, Portugal
uomian49@yahoo.com

Abstract. The usage of web shows the importance of usability and readability of the web applications. Web application fails to encounter the userȏs requirements in effective manner, because the designers are unaware of some of the important factors affecting readability, reading from the screen. Although focus losses at two positions, when eye blink, middle of text line and when text line ends, specifying suitable length of text line on the basis of Eye Blink will reduce the focus losses. The study has evaluated participants assuming three typographical variables that are font style, font color and font size with white background for improving the overall readability from the screen. The study finally suggested the Length of Text Line in three units of different standards combinations. Presenting text materials in the suggested line length will deduce the problems of causing reading, converting maximum focus losses to minimum ones.

Keywords. Readability, Line length, understandability, appearance, Eye Blink.

1 Introduction

In present era of information technology, plethora of information are available, in this fast and busy life people want and wish to get the desired information as easy as possible. The usage of computer and internet increases day by day, today online reading from computer systems become more common. According to Google data centre June 2006, about 900 million computer users exist in the world [29]. According to Internet World States, internet users in the world distributions in the world 2011 are 2,095,006,005, which are approximately 2100 million or 2 billions [30]. It means a large population use internet to get the desired information easily, cheaply, and accurately. Though information readily available, and Internet is the source which provides information with comfort and almost with no cost, but these content is not up to the mark. As huge part of information is consists of textual data, it is difficult and tedious to spend a lot of time in front of a computer because user facing a lot of problems during reading from computer screen, known as readability problems. During reading from the computer screen, usability of the web contents must be consider with significance and should not be ignored. Typography plays a major role in the web contents usability, many studies exist regarding typography, standards has been deduce after different studies for soft copy and hard copy. Web usability is an approach to make web sites easy to use for an end-user, non specialized web user, without any specialized training, and without having any pre knowledge. Keep it in mind that huge amount of information are access through World Wide Web and Web browsers. There is an urgent need to increase our knowledge about the usability or readability factors influence reading from computer screen. Few problems create inefficiency and keep users away from online reading for long time. One of the main problem that act as a huddle in reading from computer screen is Eye Strain, focus loss during reading, brain fog, and headache. These entire problem are interrelated, the focus is to reduce the focus loss during reading and hence eye strain become less or to get large time to get tired. During reading from the screen users become tired after some time and get bored. It may involve many reasons but this study and the previous study specifies suitable Length of Text Line with special relationship to Eye Blink on the basis of font style, font size, font color and background color. The study reduces the problem caused by reading and increase readability that will ultimately lead to efficiency and comfort. The structure of the paper is as follow. First section describes the focus area and importance. Second section describes how to deal with the problem and approach that reduce or solve the problem. Third section introduces related work. Next section includes methods and procedures, analysis, facts and Figure about experiment, lastly conclude the whole work.

2 Problem

Online reading or Reading from computer screen creates a lot of problems which lead to inefficiency and degradation. Reading from the computer screen from nearer distance requires more focus and attention that led the reader or user to rapid fatigue and concentration loss position. During reading from the screen it is generally observed that the user feel problem related to eye e.g. Eyestrain, which causes them less productive work by preventing them from online reading. Eyestrain does not permanently damage eyes or vision loss. However, it can be very uncomfortable and lead to productivity loss, in the form of getting tired quickly and preventing user not to use computer for long time. The part of eye that take part in reading is called macula, which is 10% of the retina center and responsible for reading [25]. The Problems during reading from the computer screen can include: a feeling of being unable to focus on the screen that is focus losses again and again, confusion, attention deficit, brain fog, irritability, headache, eye Strain or neck pain, dizziness or faintness, queasiness or vomiting, and an uncomfortable feeling down through the chest. [1]. People or computer user most of the time avoid online reading because it is time consuming as compared to the reading from the paper. Reading from the screen is 25% slower than reading from the paper, because they loss their focus from the position they reading [17]. During reading from computer screen human losses concentration or accommodation at two positions, first when the line (text line) breaks and second when eye blink when reading or line still continue. In reading eyes naturally blink in order to relax muscles from contractions form which loss the focus or concentration from the position from where he/she is reading. Starting reading again, the muscles goes to contraction form to get focus. In same way when the line breaks, sweep from the end of the line to the beginning of the next line loss focus and the eye muscles goes from contraction to rest and then from rest to concentration. If this process happens as many times the eye become tired and will cause problems. This process involve two types of eye muscles internal muscles that are sphincter of the pupil, ciliary body and eye lens, external muscles such as medial rectus, superior rectus, superior oblique, inferior rectus, inferior oblique and lateral rectus.

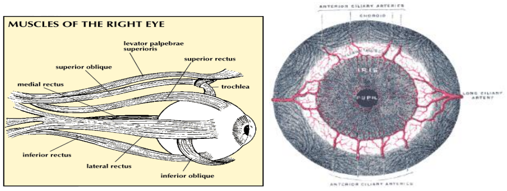


Fig. 1. Iris sphincter muscle

The mentioned problems can be overcome to some extent by decreasing the process of focus loss and focus gain which will protect eye from rapid fatigue. Deducing or adjusting the length of text line to an adequate size will convert two focus losses to one, it means that when human eye blink the line should be end which will lead to convert maximum focus losses to minimum focus losses that will protect eye strain and indirectly all other relevant problems. The efficiency and readability will increase. The most important factors that affect length of text line are Font Size, Font Style, and Spacing between lines, Text color and Background Color.

3 Related work

Related experiments are summarizing in the following section. Line Length: (WEBER, 1881) he suggested that an ideal line length was 4 inches equal to 100 millimeters. He stated further that the maximum never should exceed 6 inches or 150 mm. The same year Javel (1881) reported that line lengths should be no longer than 3.6 inches or 90 mm [7]. (COHN, 1883) suggested that 3.6 inches or 90 mm was the best length of text line, and that 4 inches or 102 mm was the longest tolerable line length for printed media [3]. (TINKER & PATERSON, 1929) found that line lengths between 3 inches and 3.5 inches or from 75 to 90 mm yielded the fastest reading performance Using 10-point black type on white paper. These recommendations were for Printed materials [4]. (SPENCER, 1968) proposed that the characters should not be exceeded from 70 characters per line, this recommendations also for printed materials or for hard copy [5]. (JACKSON & MCCLELLAND, 1979) analyze and measure the reading rate for effective reading performance on the bases of users or readers comprehension. These authors multiplied reading rate by the comprehension score to produce their index. However, they do acknowledge that multiplication may not be the optimal formula, as sacrificing comprehension for very fast reading may exaggerate reading ability [6]. (DUCHNICKY & KOLERS, 1983) found an affect reading rate for reading from screen and suggested appropriate Line length for text line, and specified that line length should be about 75 characters per line. The study has been done on the base of studying different display technologies [8]. (MASSON, 1985) The effect of line length on reading rate may be dependent upon the overall reading speed, means how fast a reader can read, as speeding up reading may result in different patterns of eye movements. Masson has reviewed research on the characteristics of naturally fast readers and found that ñsuper readers" make fewer fixations during reading from the printed materials and also from the computer screen [9]. (RAYNER & POLLATSEK, 1989) They deduce that Tinker's work identified an optimal line length of 52 characters per line. The trade-off between two opposing factors that are: If line lengths are too long, the return sweeps to the beginning of the next line are difficult. If the lines are too short, readers cannot make use of much information in each fixation [10]. (DYSON & KIPPING, 1998) Kipping et al suggest in their study that 4-inch line length of text line produced the slowest reading rate and the 7.3 inch line length produced the fastest. These experiments are done with using 12-point type font size [11]. (YOUNGMAN & SCHARFF 1999) according to Youngman and Scharff 8 inch line length of text line elicited the fastest speed as compare to the line of 4 inches or 6 inches text line lengths. They used 12-point type for font size [12]. (DYSON & HASELGROVE, 2001) A medium line length of 55 characters per line appears to support effective reading at normal and fast speeds. The investigation was made on three effecting factors that are; comprehension, reading rate and scrolling patterns [13]. (BERNARD, FERNANDEZ & HULL, 2002) had participants read 12 point prose text with line lengths of 9.6 inches or 245 mm, 5.7 inches or 145 mm and 3.3 inches or 85 mm. Their adult subjects preferred the two shorter line lengths [14]. (BOB BAILEY, 2002) Users tend to read faster if the line lengths are longer up to 10 inches. If the line lengths are too short, means around 2.5 inches or less then it may impede rapid reading. Users prefer lines that are moderately long in length that is 4 to 5 inches [15]. (DAWN SHAIKH, 2005) reported a study that examined the longest line length of 95 characters per line or 10 inches resulted in the fastest reading speed [16]. (MUZAMMIL & MUSHTAQ, 2010) suggested different text line lengths for three variables on the bases of eye blink during reading from computer screen. The study concluded if the text for reading is provided in the given length it can reduce eye strain and enhance reading capabilities from the computer screen. The experimental setup was explained i.e. equipment, participants, material for experiment, the collection of data procedures, and then explain three main major factors thatôs must be consider during reading. Previous study presents eight different combinations from the combination of three given variables for experiment, i.e. Font color, style and size. Understandability, appearance, and line length on the bases of eye blink were three major factors that were considered. According to Muzammil et al combination ñVerdana, 10, Blackò and ñVerdana, 10, Greenò are best, when consider clarity and appearance [24]. Text Layout: (J. NIELSEN, 2000) Use colors with high contrast between the text and the background. Optimal legibility requires black text on white background and so-called positive text. White text on a black background and called it as negative text. Negative text is considered to be equally good, but has no proof [17]. RICHARD and HANNA provide systematic approach to prove Nielsen statement [18]. (BRADLEY WILSON) Times are the most popular body font that is 29%, then Palatino 13%, Garamond 8% and the rest fonts 50%. The font sizes that are normally used are, Average 9.99, Median 10, Largest 12 and Smallest 8.5 [19]. City of Seattle Web Presentation and Accessibility Standards Version 2.5 are; the minimum font size for basic page body text will be or appear equal to Verdana 10 points. All page body text will be black. All page body text will be presented in Verdana font [20]. (MUSHTAQ RAZA, 2007) Preliminary mean analysis of the fonts revealed that font type Time New Roman is the most readable, conveyed personality Business like and first preference of the participants as compared to Courier New, Arial and Bradley [21]. Eye Blink or Blink rate: The analysis of the eye movement in reading is blinks, when readers close their eyes. Blinking rate increases with increasing reading time, resulting in high data losses, especially for older adults or reading impaired subjects [22]. Spontaneous blink rate was significantly larger in women than in men (19 vs. 11 blinks per minute); older women blinked more frequently than younger women. Eyelid displacement was greater in young than in older subjects [23]. When eyes are focused on an object for an extended period of time, such as when reading, blinking rate decreases up to about 3-4 times per minute [28].

4 Methods and Procedures

4.1 Considered Variable

The study involves; Typographical Variables that is Fonts size 10 points (web standard), 12 points (print media standard), and Font face Verdana (web standard), Time New Roman (print media standard) and Color Combination (Foreground and Background) that is Black/White & Green/White for female users between 20 to 40 years old.

4.2 Equipment for Experiment

The experiment has done on laptop, i.e. the text for the test was provided and the required data was collected on a system having specifications: Intel(R) Core(TM) 2 Duo CPU T5750 @ 2.00GHz, 1995 MHz Laptop of model Dell Studio 1535.

4.3 Participants

For experiment and collection of data Thirty Two (32) female volunteer were considered. They were aged from 21 to 36, that is 18 female participants are age ranged between 21 and 25, 9 participants are age between 26 and 30, and 5 female volunteered participants are age ranged between 31 and 36. Overall average age is 25. The participants were having correct vision, and there is no disability. There are 16 or 50% participants, graduate students, 7 or 22% were postgraduate students, 6 or 19% professionals, and 2 or 9% were chosen randomly.

4.4 Experimental Setup

The data is collected in the month of August. The temperature were around 30 degree centigrade (30^{o}), weather were dry in those days i.e. no rain from few days. All the data are taken in under normal room light and the users are sat 24 to 28 inches away from the screen, means that follow the safety precautions that should be following during using computer. The text provide for experiment in ñInternet Explorerò web browser.

4.5 Experimental Material

A web page is designed with 8 different paragraphs with different typographical properties. The design page is presented to the users or participants. Each paragraph is the combination of different parameters, means that each paragraph has one parameter different than all other paragraph. Each graph is around 5 to 6 lines long and the length of each line is around 10.5 inches per line. This textual material is provided in Internet Explorer Web browser. The considered variables in the experiment are the two different standards, i.e. Print media standards and Web standards. The composition of each paragraph is given in the following passage. Short hands: Style, Size, Color. Paragraph 1: Time New Roman, 10, Black. Paragraph 2: Time New Roman, 12, Black. Paragraph 3: Verdana, 10, Black. Paragraph 4: Verdana, 12, Black. Paragraph 5: Time New Roman, 10, Green. Paragraph 6: Time New Roman, 12, Green. Paragraph 7: Verdana, 10, Green. Paragraph 8: Verdana, 12, Green

4.6 Data collection Procedure

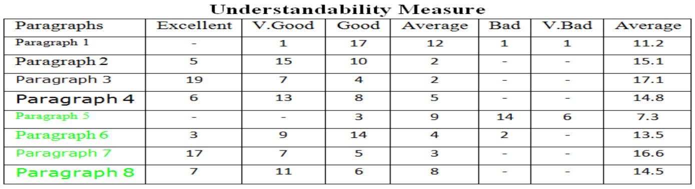
The data has been collected in the following manner. Step 1: The user is positioned and instructs how to read the passages in order to collect correct data. Step 2: The video has been taken from the user or reader but they are unaware from it because of; to get natural blink during reading. Step 3: After the video been taken, the user is than briefly explained with the process, means the purpose of research and why the passages is read by them in a said constraint. Step 4: A questioner has been conduct in order to collect information about the different combinations. **Questioner:** A questioner contains of 16 questions, 8 about ñto find out the degree of understandability or clarity of words in the paragraph based on fontsôstyle and sizeò. And 8 about ñto find out the degree of pleasant appearance or good-looks of the paragraph based on fontsô style and sizeò. Step 5: A thorough discussion has been made on each Question, and the conclusion was deduce. Step 6: Each video analyzed carefully in order to find out the suitable length of Text Line on the bases of Eye blink for each combination, through which the maximum focus losses is replaced by fewer one. Step 7: The values are put in EXCEL tool and the Result are deduced in the form of Charts and Tables.

5 Results Analysis and Discussions

5.1 Clarity and Understandability

The user is asked to rate the paragraph, on the basis of ñthe degree of understandability/clarity of words in the paragraph based on fontsô style and sizeò. Average perception about each paragraph: Paragraph 1 is too small, looking to much condense, and difficult to read. Paragraph 2 is a bit easier to read than the first combination but the words are looking attached to each other. The words are looking shaky, that is unstable. Paragraph 3 is Clearer, Understandable and very much easier to read as compare to the previous ones. Paragraph 4 is also clear and easy to read but it is observe that the user lost correct line when the finish one and starting new line it may be because the lines are looking very close. During discussion it is observed that the readers get bored during reading such large font size textual materials. Paragraph 5 is too much difficult to read because there is not high contrast between background and Text color, the words are too small and looking very condensed plus the green color font is looking very sharp with white background. Paragraph 6 is difficult to read because small, low contrast and looking unstable. Paragraph 7 is ok and is clear than the paragraph 5 and 6, observing that the readers have positive approach towards this combination. Paragraph 8 is ok but looking a bit larger and same problem as in paragraph 4.

Table 1.



Understandability Measure

Paragraphs	Excellent	V.Good	Good	Average	Bad	V.Bad	Average
Paragraph 1	-	1	17	12	1	1	11.2
Paragraph 2	5	15	10	2	-	-	15.1
Paragraph 3	19	7	4	2	-	-	17.1
Paragraph 4	6	13	8	5	-	-	14.8
Paragraph 5	-	-	3	9	14	6	7.3
Paragraph 6	3	9	14	4	2	-	13.5
Paragraph 7	17	7	5	3	-	-	16.6
Paragraph 8	7	11	6	8	-	-	14.5

When the Questioner is evaluated then the userôs data is converted into simple form as shown in table 1 and 2. For the conversion of qualitative measures into quantitative measure, Categories are assigned to some numeric values, that as Excellent = 0.6, V-Good = 0.5 é V-Bad = 0.1 and then each numeric value is multiplied by the number of users chose the respective option. That is; 0.6 * X_1 + 0.5 * X_2 + 0.4 * X_3 + 0.3 * X_4 + 0.2 * X_5 + 0.1 * X_6 = Average. Where X_x is the number of user select the respective category/choice. Automatically Graph has been generated. As shown in the Figure. 2 for table 1 and Figure 3 for table 2

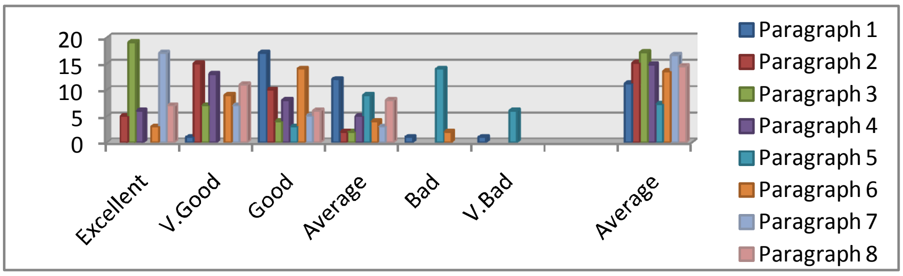


Fig. 2. Understandability and clarity for each combination

The Table and Chart shows the understandability and clarity for each combination, specified in each paragraph. The average in the table shows the clarity of each combination and in chart the BAR height shows the clarity of each paragraph. Each combination can be ranked as; Paragraph 3: Verdana, 10, Black. Paragraph 7: Verdana, 10, Green. Paragraph 2: Time New Roman, 12, Black. Paragraph 4: Verdana, 12, Black. Paragraph 8: Verdana, 12, Green. Paragraph 6: Time New Roman, 12, Green. Paragraph 1: Time New Roman, 10, Black. Paragraph 5: Time New Roman, 10, Green

In simple words, combination in paragraph 3 should be use regarding understandability and clarity which is also the result of previous study.

5.2 Pleasant Appearance

The questioner contains fifty percent questions regarding appearance of each paragraph. The users were asked to rate the different combination on the basis of ñthe degree of pleasant appearance or good-looks of the paragraph based on fontsô style and sizeò. **Average perception of users about each paragraph:** Paragraph 1 is looking very nice but not comfortable; the text is not easy to read. Paragraph 2 is looking very much nice and nice to read. Observed that the user like the chemistry of the font. Most of the users like the font style. Paragraph 3 is very much simple and very stable and much comfortable, enjoying reading. Paragraph 4 is looking boring, very simple and extra large. Paragraph 5 is appearing nice but too much uncomfortable, but as it is looking shaky, therefore reading this combination is more irritating. Paragraph 6 is appearing to be very fine than the paragraphs 5, 7 and 8. Paragraph 7 is looking fine and comfortable than others. Paragraph 8 is ok but looking a bit larger and boring.

Table 2.

Appearance Measures

Paragraphs	Excellent	V.Good	Good	Average	Bad	V.Bad	Average
Paragraph 1	-	8	13	11	1	-	12.7
Paragraph 2	10	13	6	3	-	-	15.8
Paragraph 3	8	14	5	5	-	-	15.3
Paragraph 4	3	12	10	7	-	-	13.9
Paragraph 5	-	9	14	6	3	-	12.5
Paragraph 6	9	11	9	3	-	-	15.4
Paragraph 7	12	8	10	2	-	-	15.8
Paragraph 8	4	9	14	5	-	-	14

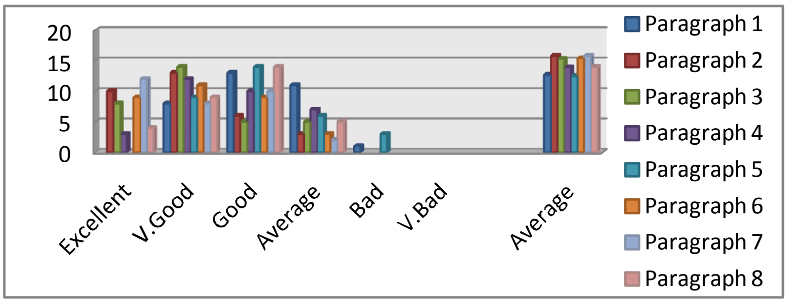


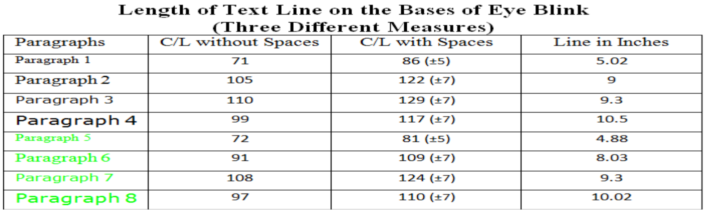
Fig. 3. Appearance and look for each combination

The Table and Chart shows the appearance and look for each combination, specify in each paragraph. The strength of appearance for each combination is shown in the form BAR height. Each combination can be ranked as; Paragraph 7: Verdana, 10, Green And Paragraph 2: Time New Roman, 12, Black. Paragraph 6: Time New Roman, 12, Green. Paragraph 3: Verdana, 10, Black. Paragraph 8: Verdana, 12, Green
Paragraph 4: Verdana, 12, Black. Paragraph 1: Time New Roman, 10, Black
Paragraph 5: Time New Roman, 10, Green. In simple words, combination in paragraph 7, paragraph 2 and paragraph 6 is best typographic combination regarding appearance and looks.

5.3 Length of Text line on the Basis of Eye Blink

Data Analysis Procedure: During reading video has been taken from each user without informing them (after taking video the whole process and research purpose is explained to the user), in order to observe eye blink naturally when the users reading the textual materials. Calculate the length of Text line for each Blink for every paragraph and find out the Average of each user. Similarly the process of calculating length of text line is repeated for each user, and then finds out the grand average for all the users. The length has been specified. The length is measured in three different units that are Characters per Line without spaces, Characters per line with spaces and length in Inches per line.

Table 3.



Length of Text Line on the Bases of Eye Blink
(Three Different Measures)

Paragraphs	C/L without Spaces	C/L with Spaces	Line in Inches
Paragraph 1	71	86 (±5)	5.02
Paragraph 2	105	122 (±7)	9
Paragraph 3	110	129 (±7)	9.3
Paragraph 4	99	117 (±7)	10.5
Paragraph 5	72	81 (±5)	4.88
Paragraph 6	91	109 (±7)	8.03
Paragraph 7	108	124 (±7)	9.3
Paragraph 8	97	110 (±7)	10.02

The results has been obtained in three different measurable units that is Characters per Line without space count, Characters per Line with space count where (±5 or ±7) means the number of spaces vary in range of 10 spaces per line that is 93 (±5) means 88 to 98 and 133 (±7) means 126 to 140. The length per line in C/L without spaces and Length per line in inches give us exact measure than the C/L with spaces. Figure 4 shows length of line as Character per Line without spaces (spaces are not counted) for each combination of parameters. The paragraph 3 (Verdana, 10, Black) accommodate 110 characters per blink which is greater than all other combinations and shows to be the most efficient typographical combination. The combination in paragraph 7 and paragraph 2 accommodate 108 and 105 characters per blink respectively and so on.

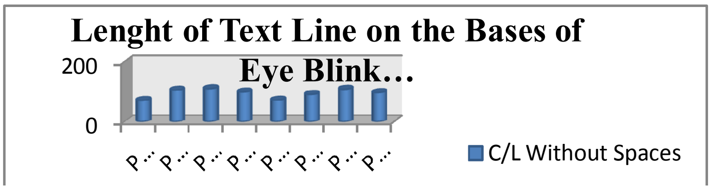


Fig. 4. Charter per line without spaces

Figure.5 shows the number of characters per line with spaces for one natural eye blink. The length is same as in above Figure.5 but not precise and accurate like other units because the number of spaces varies up to ±5 or ±7 spaces per line and depends upon the words used

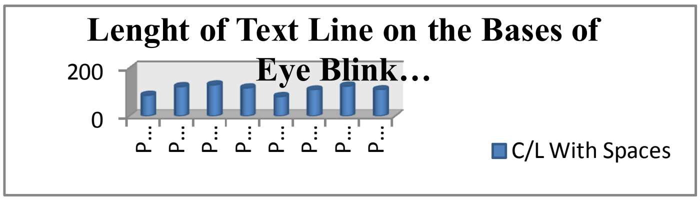


Fig. 5. Charter per line with spaces

Figure 6 shows length of text line of each combination in inches. As the length of paragraph 4 is longest one but it does not means that it is more efficient than other paragraphs because the number of words are less than from paragraph 3, 7 & 2 because the size is bigger which increase the length of text line but accommodate less words.

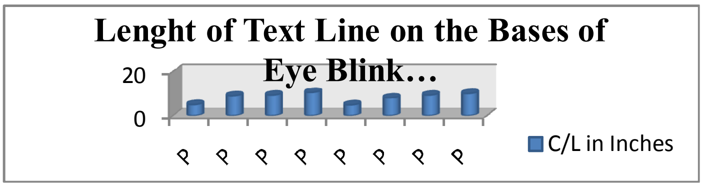


Fig. 6. Charter per line in inches

6 Conclusion

The main objective of this study is to suggest suitable length of text line for web page or online reading, while keeping Eye forefront. On the bases of eye blink suggest the suitable length of text line considering different typographical variables. This scrutinize illustrate three important things about various combination of standards. One; the length of text line on the bases of eye natural blink second; understandability or clarity, third; the good attractive appearances of the combinations. The study suggested the Length of Text Line in three units of different standards combinations. Presenting these text materials for reading through internet or generally from computer screen in the suggested line length will deduce the problems of causing reading, by converting maximum focus losses to minimum ones. This study proves that ñparagraph 3ò; the combination ñVerdana, 10, Blackò shows overall promising results for efficient line length, understandings, and for attractiveness. The combinations in paragraph 2 and paragraph 7 are the competitive combinations showing good results in all three aspects. These recommendations serve materials that study online or from computer screen, for example Thesis, e books, research papers etc. in simple words, during designing materials for reading from computer screen, the recommendations should be considered for different combinations for maximum efficiency. Future work is to compare the previous study [24] to this study in the same direction to achieve more results.

Reference

1. RICHARD CONRAD, Ph.D, SUBLIMINAL FLICKER Part I: Computer screens, TV's and Flicker Sensitivity.
2. ANDREW DILLON; HUSAT Research institute UK, Reading from the paper versus screen. Originally approved 2003 by the Web Governance Board and the Business Management Council.
3. COHN, H. (1883), *Die Hygiene des Auges in den Schulen,* Leipzig. [See Tinker and Paterson, 1929]
4. TINKER AND PATERSON, D.G. (1929), Studies of typographical factors influencing speed of reading: Length of line, The Journal of Applied Psychology.
5. SPENCER, H. (1968). The visible word. London: Royal College of Art. & TINKER, M. A. (1963). Legibility of Print. Ames: Iowa State University Press.
6. JACKSON, M. D. & MCCLELLAND, J. L. (1979). Processing determinants of reading speed. Journal of Experimental Psychology.
7. WEBER, A. (1881), Ueber die Augenuntersuchungen in den hoheren schulen zu Darmstadt, Abtheilung fur Gesunheitspflege, Marz.
8. DUCHNICKY, R. L & KOLERS, P. A. (1983). Readability of text scrolled on visual display terminals as a function of window size.
9. MASSON, M. E. J. (1985). Rapid reading processes and skills. In G. E. MACKINNON & T. G. WALLER, Eds. Advances in theory and Practice, Vol. 4.
10. RAYNER, K. & POLLATSEK, A. (1989). The Psychology of Reading. Hillsdale, NJ: Lawrence Erlbaum.
11. DYSON, M. C. & KIPPING, G. J. (1998). The effects of line length and method of movement on patterns of reading from screen, Visible Languages 32(2)
12. YOUNGMAN, M. and SCHARFF, L. (1998), Text width and margin width influences on readability of GUIs.
13. MARY C. DYSON AND MARK HASELGROVE (2001), Department of Typography & Graphic Communication, The University of Reading, 2 Earley Gate, Whiteknights, Reading RG6 6AU, UK, The influence of reading speed and line length on the effectiveness of reading from screen.
14. BERNARD, M., FERNANDEZ, M. and HULL, S. (2002). The effects of line length on children and adults' online reading performance, Usability News.

15. BOB BAILEY (2002), Optimal Line Length: Research Supporting How Line Length Affects Usability.
16. DAWN SHAIKH, A.D. (2005), The effects of line length on reading online news, UsabilityNewshttp://psychology.wichita.edu/surl/usabilitynews/72/LineLength.htm
17. J. NIELSEN (2000), Designing web usability: The practice of simplicity.
18. HALL, R. and HANNA, P. (2004), The Impact of Web Page Text-Background Color Combinations on Readability, Retention, Aesthetics, and Behavioral Intention, Behavior & Information Technology, forthcoming.
19. BRADLEY WILSON, Readability First at wilsonbrad@aol.com
20. City of Seattle Web Presentation and Accessibility Standards Version 2.5: Revised by Internet Board, approved by the Technology Board, January 2009 originally approved 2003 by the Web Governance Board and the Business Management Coun.
21. MUSHTAQ RAZA, SZABIST Islamabad Pakistan, Perception of Participants about fontôs readability, style, youthfulness and fun, business likeness and general preference.
22. Reconstruction of eye movements during blinks: Max-Planck-Institut für Physik Komplexer Systeme, Nöthnitzerstr, Germany.
23. Spontaneous blinking in healthy persons: an optoelectronic study of eyelid motion. Functional Anatomy Research Center (FARC), Università degli Studi, Milano.
24. MUZAMMIL KHAN, MUSHTAQ RAZA, and NASIR RASHID (2010). Appropriate Length of Text Line with Special Relationship to Eye Blink to Reduce Maximum Focus Loss. International Conference on Internet Computing 2010: 97-102
25. Pasadena Eye Associates, Frequently Asked Questions : http://www.pasadenaeye.com/faqs.html
26. Taylor, JR; Elsworth, JD; Lawrence, MS; Sladek Jr, JR; Roth, RH; Redmond Jr, DE (1999). "Spontaneous blink rates correlate with dopamine levels in the caudate nucleus of MPTP-treated monkeys". *Experimental neurology* **158** (1): 214ï 20
27. L. E. Ebite, T. C. Ozoko & A. O. Eweka : Rate Of Blinking Among Medical Students In Delta State Nigeria: Is The Eyelid A Polygraph? . *The Internet Journal of Ophthalmology and Visual Science.* 2009 Volume 6 Number 2
28. Figure 1 & 2, http://media-2.web.britannica.com/eb-media/47/63347-004 610F94B5.gif & http://upload.wikimedia.org/ wikipdia/commons/2/2f/Gray878.png
29. Google Data Centers vs. Microsoft Infrastructure: http://dondodge.typepad.com/the_next_big_thing/2006/06/google_data_cen.html
30. Internet World States, World Internet Users and Population Statistics: http://www.internetworldstats.com/stats.htm

Workload and Situational Awareness management in UAV teams through interface modelling

António Sérgio Ferreira

Faculdade de Engenharia da Universidade do Porto, Underwater Systems and Technology Laboratory (USTL),
Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal
{asbf}@fe.up.pt

Abstract. The practical use of Unnamed Aerial Vehicles (UAV) to extract the human element from dangerous situations is becoming a more feasible scenario with every new iteration of technological advancement. Cost and implementation time reductions have led to the emergence of various *off-the-shelf* solutions which allow a wide array of usage of these systems. However, on the most part, the development emphasis of these systems does not reside on usability quality or human factor effects. This hampers the viability of theses systems in scenarios were human operators must be focused on other duties other then vehicle control and supervision, therefore a priority must be given to the control interface. Through the implementation of a Real-Time Strategy (RTS) game interface paradigm, in an already existing command and control framework, it was possible to develop a control and supervisory interface console which achieved a decrease of workload in preliminary testing.

Keywords: HCI, HRT, UAV, Workload, Situational Awareness, Real-time strategy games

1 Introduction

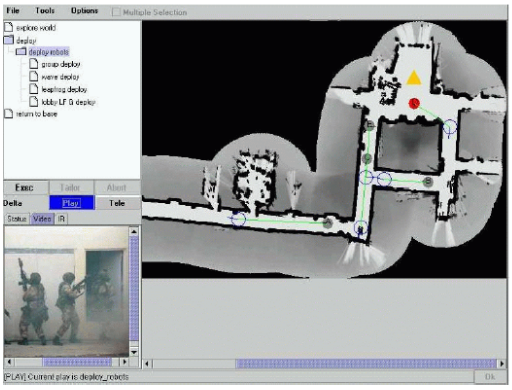
With every new chapter of technological advancement, either in the hardware and software sectors, it becomes easier or more cost-effective to develop and apply Human-Robot Teams (HRT) in operation scenarios of dangerous nature. A wide array of scenarios, either simple Search and Rescue missions [1] or more complex military missions [2], are steadily becoming routine theatres of operations for autonomous vehicles. Among the various types of automated vehicles UAVs, through their ability to quickly cover large tracts of ground from an aerial perspective, have demonstrated be a valuable asset when aiding the activities of human teams in a number of complex and demanding situations [3].

In order to do so these vehicles rely on complex control systems which allow the human team member to take advantage of their capabilities, however the rising complexity of said systems has led to an increasing need for more human operator per UAV platform. Nevertheless steps are being taken to allow future unmanned vehicles systems to invert the operator-to-vehicle ratio so that one operator can control multiple vehicles connected through a decentralized network [4]. However this decentralization puts an increasingly high strain on the human operator's workload [5] reducing the team's overall performance [6] if steps aren't taken to deal with these variables.

There are various ways to cope with the workload reduction challenge. It can be achieved by an emphasis on further developing and extending the control schema by creating frameworks which allow an adaptive level of automation [7] [8] and dealing with the inner-workings of control algorithms or by addressing interface usability issues, prioritizing the human user experience.

In order to achieve a low-entropy interface with the human operator, in a stress filled situation, a great amount of attention must be given to the saturation point of information the operator can handle at one given moment [9][10]. Conceiving the interface using a multi-modal approach is advantageous since it's one way of avoiding the natural bottleneck that arises from using only one communication channel between the UAV and the operator [11].

Nevertheless, even with these approaches to reduce the operator workload there is another factor which must be taken into account when dealing with Human-Computer Interaction (HCI). Every interface has a learning curve which the user must overcome in order to fully take advantage of its capabilities, therefore it is safe to assume that the more familiar the interaction schema the lower the learning curve will be. It can than be assumed that video games could prove a valuable asset when looking for familiar interface layouts and techniques. On this premise there have been various successful applications of game-based interfaces in HRT control [12][13]. For this paper the RTS game genera will be used as the basis for interface analysis. This type of game paradigm has already shown promise in other research [14] surrounding autonomous vehicle control.



[PLAY] Current play is deploy_robots

Fig. 1. Interface example of the Macbeth command and control system[21].

To deal with the emergent need to simplify the method of interaction with more complex and capable UAVs the Neptus framework, developed by the Underwater Systems and Technology Laboratory at FEUP, has been extended from it's original role as a Command, Control, Communications and Intelligence (C4I) [19] for autonomous underwater vehicles (AUV) to also incorporate UAV control and supervision. It is upon this already established framework that the validity of an RTS paradigm will be tested. All development efforts were made so to minimize the use of additional peripherals than the standard single monitor setup to on the one hand, reduce logistical system costs and on the other hand, increase the number of operational scenarios where the console might be of viable use.

In order to evaluate both workload and situational awareness metrics specific tests will be employed. For workload evaluation the method used was the NASA Task Load Index (NASA-TLX) questionnaire, already widely used in military and commercial aviation pilot workload tests [15][16]. For situational awareness the method used was the Situation Awareness Global Assessment Technique (SAGAT) also widely used, not only in the aviation field but also HRT testing[17] [18]. Although time constraints and availability of certified humam operators limited the testing phase to only one account, the test proved promising results and will be augmented in future developments.

In Section 2 a quick overview of the framework used as basis for this paper is given followed by a conceptual description of the developed prototype in Section 3. All the tests performed on the prototype are detailed in Section 4 and the subsequent conclusions are presented in Section 5. In Section 6 an analysis of the envisioned future work is made.

2 Related Work

There already exist command and control frameworks which demonstrate an emphasis on the quality of the human interface, as seen in Fig. 1, with the intent of reducing workload in situations where one sole human operator must command various autonomous vehicles. However the recurring standard interface paradigm is one of high complexity and high logistical and maintenance cost, even for the control of single autonomous platform, as demonstrated in Fig. 2.

One which attempts to reverse the aforementioned standard is the Neptus C4I framework which is a distributed framework for operations with networked vehicles, systems, and human operators. Neptus supports all the phases of a mission life cycle: planning, simulation, execution, and post-mission analysis. Moreover, it allows operators to plan and supervise missions concurrently [19].

The Neptus framework communicates with all autonomous vehicles thought the IMC [20] communication protocol. This protocol, based on XML, augments the frameworks compatibility with various kinds of autonomous vehicles by allowing a level of abstraction which promotes decoupled software development.

Furthermore, in order to minimize undesirable interdependencies throughout the framework's architecture, the Neptus framework consists of a series of independent plugins surrounding a core section of functionalities.

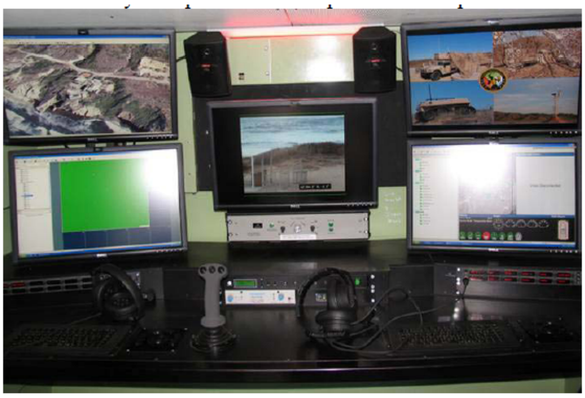
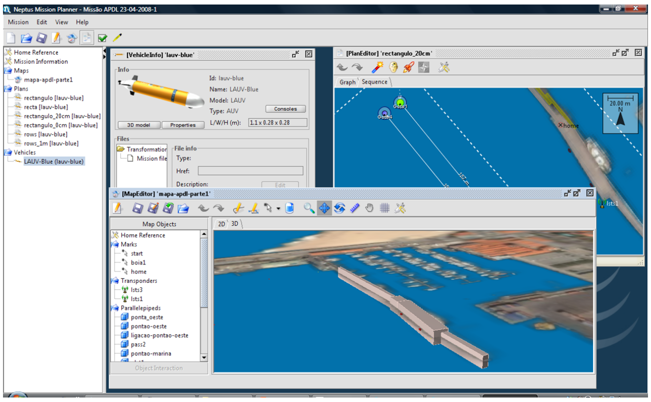


Fig. 2. Interface example of the MOCU military system[2].

Nevertheless, Neptus encompasses a console building application which facilitates the rapid creation of new operation consoles for new vehicles with new sensor suites, as well as the remodelling of old consoles for current vehicles (Fig. 3). It is one such console that ultimately will be conceived as a result of the work developed.



Neptus Mission Planner - Missão APDL 23-04-2008-1

[VehicleInfo] 'lauv-blue'

Id: lauv-blue
Name: LAUV-Blue
Model: LAUV
Type: AUV
L/W/H (m): 1.1 x 0.28 x 0.28

[PlanEditor] 'rectangulo_20cm'

[MapEditor] 'mapa-apdl-parte1'

Fig. 3. Example of a normal operational console provided by the Neptus C4I framework.

3 Prototype Development

Using the Neptus framework, development began towards the creation of a console that would incorporate conceptual ideas implemented in modern RTS interfaces in order to further expand its usability by the human operator.

3.1 Conceptual Design Details

One of the major difficulties in this kind of endeavour is the simplification of the large amount of data deemed necessary for the safe operation and supervision of UAV systems. The usually overwhelming quantities of numeric and textual data tend to lead to a multi-monitor solution which quickly raises the complexity of the interaction experience.In order to avoid this necessity a lot of information fusion is needed.

The standard RTS game relies heavily on representative icons to convey information to the player. Distinct icons can, on their own, quickly transmit large amounts of information, that would otherwise require a large amount of textual representation, just by small changes in size, orientation and draw rate. With simple icon modifications one can summarize a vast amount of the UAV's relevant data. Coupled with this the RTS genera also employs color as a way to clearly indicate unit status and different levels of urgency. Both these elements combined on their own can account for a great amount of information simplification. However there are other aspects that can be extracted from the RTS example which can prove invaluable to the overall console's efficiency.

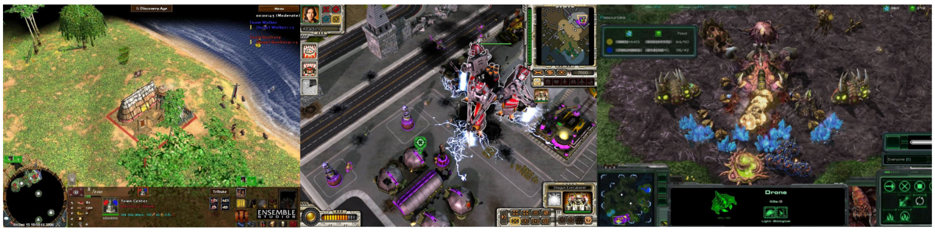


Fig. 4. Example of similarities in layouts between different RTS games interfaces.

Modern RTS games, although having different themes at their core, follow a seemingly standardized formula when laying out their different interface components (Fig. 4). One can reason that this behaviour derived from the genera's vast years of experiments with different approaches which slowly converged into a model that is now amply used by the majority of developers. Moreover, even the various interface components converged to form a pre-set of core interface elements that are re-hashed from game to game (mini-map, unit status, etc...).

These unofficial standards, combined with the ample size of the game market, led to an unintentional spread of these tendencies to a wide audience thus making it a useful basis of development when aiming at a reduced interface learning curve.

3.2 Architectural Details

Following the conceptual design, a layout for the console's components was idealized. As shown in Fig. 5 the focus is to emulate the default RTS layout by having a central large map region (section 1), a mini-map to aid the human operator to not lose context of other UAVs operating outside his view scope (section 2), a series of smaller additional panels which will house simplified numerical data (sections 3 through 5) and two final panels which will be used to house contextual information about other autonomous vehicles operating in the area.

The central concept is to concentrate all the information with the highest priority in the main panel (section 1) through use of icon disposition and color, relegating textual information to one of the support panels.

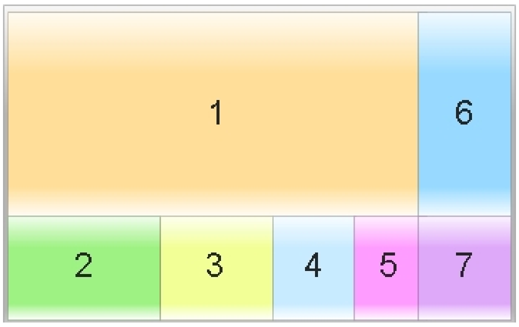


Fig. 5. Representation of the projected console layout.

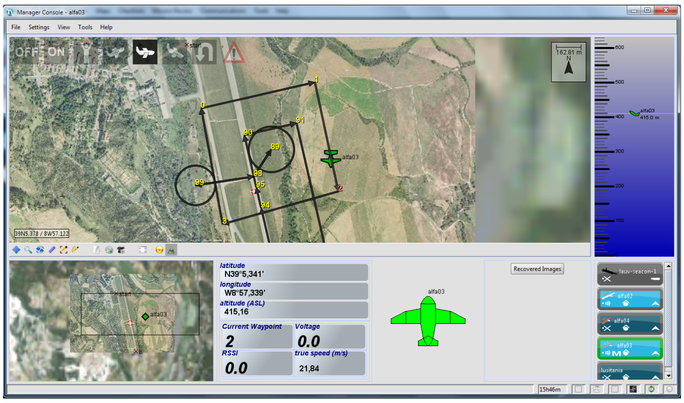
Before the development of this new operation console started there was already a pre-existent standard command console for UAV operations which served as a benchmark in the later testing stage, in order to determine the effectiveness of the new design implementation.

3.3 Conceived Prototype

In Fig. 6 we can see the finalized operational version of the developed console. As expected, from the RTS paradigm influence, we have a large main map, aided by a smaller mini-map, which provides a top-down view of the operation area. However the typical isometric point-of-view was not adopted due to rendering limitations when attempting to model 3D environments. On a normal operation scenario it is complicated to acquire an up-to-date and accurate 3D model of the area so it was deemed that, as this stag,e a simple map-based representation would be preferable.

Hovering over the map are icon indicators of the various stages of the UAV's flight plan, from the pre-launch stage to the final runway taxi quickly giving the operator a grasp of the mission's current stage.

The mini-maps viewport feature allows the operator to quickly know what part of the operation theatre is being shown on the main panel, as well as what are the current visual limits.



latitude N39°5,341'
longitude W8°57,339'
altitude (ASL) 415,16
Current Waypoint 2
Voltage 0.0
RSSI 0.0
true speed (m/s) 21,84

Fig. 6. Example of the prototype console developed.

Aiding the main viewing panel are a simple set of numerical data regarding velocity and GPS indications, however the UAV's actuator status is complied and provided by the miniature segmented UAV model. Each section informs the human operator of its status by alternating color, providing the operator with a faster means of understanding the UAV's overall condition.

At the far right of the console are two auxiliary panels one of which allows a quick visual reference of the all active autonomous vehicles' altitudes and another which lists all detected autonomous vehicles in the operation area. Through this list the operator can also switch between operating vehicles.

4 Tests and Results

In order to test the efficiency of the new proposed interface alterations, on reducing the operator's perceived workload and augmenting his situational awareness, a simulation was held at the USTL with one of its senior UAV operator.

The simulation asked for a team of 3 UAVs to accomplish a simple set of surveillance tasks over a pre-designated and already known area. One of the operators was target of the evaluation while the other two were only considered participants. Firstly the operator used the already existing Neptus UAV interface to execute his mission, afterwards the new developed interface was used.

4.1 Workload Analysis

At the end of the simulation the operator under evaluation was given a standard NASA-TLX questionnaire to determine his perceived workload. The results obtained are shown in Fig. 7.

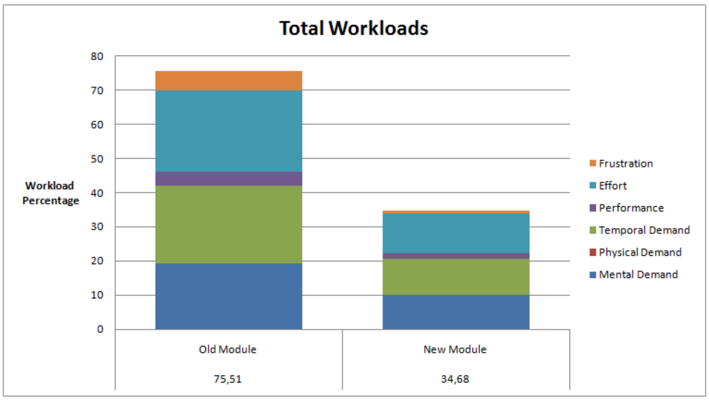


Fig. 7. Overall workload percentage comparison, segmented by NASA-TLX components.

4.2 Situational Awareness Analysis

During the simulation the operator would, from time to time, quickly be negated access to the operation interface in order to answer a set of pre-determined queries with the intent of ascertaining how aware of his surrounding environment he was. The queries are as follows:

1. Point-out the position of each UAV currently active;
2. Single-out the currently selected UAV;
3. Determine the altitude of all active UAVs;
4. Determine the altitude of the seleted UAV;
5. Determine the speed of the seleted UAV;
6. Determine the current UAVs waypoint;
7. Determine the current UAVs ID.

A total of ten interruptions were made and the results obtained are shown in Fig. 8.

These preliminary results show promise as it seems accurate to extrapolate that the modifications made on the command interface, using the RTS paradigm, have lowered the operators workload and, at the same time, risen his situational awareness during the mission's execution.

5 Conclusions

Throughout this paper references were made to the growing importance of UAV systems, paying special attention to their valuable application in real world scenarios. It was then presented the concepts behind a possible solution to be implemented into a pre-existent framework with the ultimate goal of managing UAV workload a mission scenario and the details around this solution were presented and discussed.

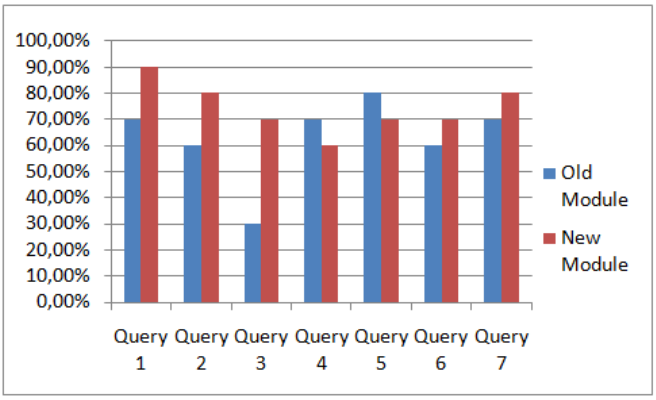


Fig. 8. Overall situational awareness percentage comparison, throughout a series of pre-determined queries.

The C4I operation console, ultimately created based on the presented solution, enabled the reduction of the workload felt by the operator while at the same time increasing his situational awareness.

6 Future Work

Although promising these type of results are, by their nature, subjective to the human operator's sense of workload and own experience of operation procedure. This means that further testing must be performed in order to ascertain if the overall improvements detected prevail in a wide variety of human operators. Furthermore, a closer look into the level of situational awareness gained must be taken by providing more challenging and complex scenarios to work with, while expanding the set of tasks to be evaluated.

References

1. Kadous, M.W., Sheh, R.K.-M., Sammut, C.: Controlling Heterogeneous Semi-autonomous Rescue Robot Teams. 2006 IEEE International Conference on Systems, Man and Cybernetics. pp. 3204-3209. IEEE (2006).
2. Powell, D.: Multi-robot operator control unit. Proceedings of SPIE. p. 62301N-62301N-8. SPIE (2006).
3. Jones, G., Berthouze, N., Bielski, R., Julier, S.: Towards a situated, multimodal interface for multiple UAV control. 2010 IEEE International Conference on Robotics and Automation. pp. 1739-1744. IEEE (2010).
4. Cummings, M.L., Clare, A., Hart, C.: The Role of Human-Automation Consensus in Multiple Unmanned Vehicle Scheduling. Human Factors: The Journal of the Human Factors and Ergonomics Society. 52, 17-27 (2010).
5. Prewett, M.S., Johnson, R.C., Saboe, K.N., Elliott, L.R., Coovert, M.D.: Managing workload in humanrobot interaction: A review of empirical studies. Computers in Human Behavior. 26, 840-856 (2010).

6. Crandall, J.W., Cummings, M.L.: Developing performance metrics for the supervisory control of multiple robots. Proceeding of the ACMIEEE international conference on Humanrobot interaction HRI 07. 33 (2007).
7. Bocaniala, C.D., Sastry, V.V.S.S.: On enhanced situational awareness models for Unmanned Aerial Systems. 2010 IEEE Aerospace Conference. pp. 1-14. IEEE (2010).
8. de Brun, M.L., Moffitt, V.Z., Franke, J.L., Yiantsios, D., Housten, T., Hughes, A., Fouse, S., Housten, D.: Mixed-initiative adjustable autonomy for human/unmanned system teaming. AUVSI Unmanned Systems North America Conference (2008).
9. KABER, D., WRIGHT, M., SHEIKNAINAR, M.: Investigation of multi-modal interface features for adaptive automation of a humanrobot system. International Journal of Human-Computer Studies. 64, 527-540 (2006).
10. Yanco, H.A., Drury, J.L., Scholtz, J.: Beyond Usability Evaluation: Analysis of Human-Robot Interaction at a Major Robotics Competition. Human-Computer Interaction. 19, 117-149 (2004).
11. Maza, I., Caballero, F., Molina, R., Pea, N., Ollero, A.: Multimodal Interface Technologies for UAV Ground Control Stations. Journal of intelligent and robotic systems. 57, 371391 (2010).
12. Hassell, A.J., Smith, P., Stratton, D.: An evaluation framework for videogame based tasking of remote vehicles. Proceedings of the 4th Australasian conference on Interactive entertainment. p. 10. RMIT University (2007).
13. Aubert, T., Corjon, J., Gautreault, F., Laurent, M.: Improving situation awareness of a single human operator interacting with multiple unmanned vehicles: first results. 26-27 (2010).
14. Jones, H., Snyder, M.: Supervisory control of multiple robots based on a real-time strategy game interaction paradigm. 2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236). pp. 383-388. IEEE (2001).
15. Whetten, J.M., Goodrich, M.A.: Beyond robot fan-out: Towards multi-operator supervisory control. 2010 IEEE International Conference on Systems, Man and Cybernetics. pp. 2008-2015. IEEE (2010).
16. Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., Goodrich, M.: Common metrics for human-robot interaction. Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction. pp. 3340. ACM, New York, New York, USA (2006).
17. Adams, M.J., Tenney, Y.J., Pew, R.W.: Situation awareness and the cognitive management of complex systems. Human Factors The Journal of the Human Factors and Ergonomics Society. 37, 85-104 (1995).
18. Stanton, N.: Handbook of Human Factors and Ergonomics Methods. CRC Press (2004).
19. Dias, P., Pinto, J., Gonalves, R., Gonalves, G., Sousa, J.: Neptus, command and control infrastructure for heterogeneous teams of autonomous vehicles. ICRA - IEEE International Conference on Robotics and Automation. 2768-2769 (2007).
20. Martins, R., Dias, P.S., Marques, E.R.B., Pinto, J., Sousa, J.B., Pereira, F.L.: IMC: A communication protocol for networked vehicles and sensors. OCEANS 2009-EUROPE. pp. 1-6. IEEE (2009).
21. Simmons, R., Apfelbaum, D., Fox, D., Goldman, R.P., Haigh, K.Z., Musliner, D.J., Pelican, M., Thrun, S.: Coordinated deployment of multiple, heterogeneous robots. Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113). pp. 2254-2260. IEEE (2000).

Session II

Information Systems

Chairman: Klaus Schaefers

SmartAd: Advertisement System for Next Generation Networks
Mário Antunes

Time-aware collaborative fitering: a review
João Vinagre

SmartAd: Advertisement System for Next Generation Networks

Mário Antunes

Instituto de Telecomunicações Aveiro

Abstract. Next generation networks and 3GPP/IMS framework opens a new series of business opportunities. Telecommunication operators have a new business model that allows the inclusion of services developed by 3^{rd} party developers. Through the composition of services it is possible to build innovative services that fully integrates with the network of operators. This paper presents a novel service designated SmartAd. The proposed service advertises publicity through a new channel, next generation networks. It takes advantage of the fixed-mobile convergence and is built using SOA methodology on top of 3GPP/IMS framework. A prototype was developed in order to demonstrate the technological viability of the project. Also a economic study as performed in order to demonstrate the economical viability of the project.

Keywords: Fixed-mobile convergence, NGN, 3GPP/IMS, SIP, SOA

1 Introduction

Telecommunications operators always had interest in providing services beyond telephony. This interest was the origin of convergence of services, namely audio, video and data, commonly known as Triple Play. Currently with New Generation Networks (NGN) [9] there is a new set of opportunities to enchant the operators network with 3^{rd} party applications or services. One of the biggest advantages of NGN is the reduction of the maintenance cost of the network. The concept of a NGN consists basically on transporting all types of data in a packet switching network that uses the IP protocol.

The generality of current services of a telecommunications operator on the Walled Garden model built on top of the OSA/Parlay [14] architecture. The Walled Garden model is a closed model where the telecommunications operators possess total control under all the provided services. The services developed under this architecture are monolithic and vertical, which means services are highly dependent on the network, as show in Figure 1, where each technology posses different components to the same functionalities. Still the OSA/Parlay architecture introduced the concept Service Delivery Platform (SDP). It is a uniform and consistent interface to access the resources that are available in the Walled Garden of the telecommunication operator, for instance, message system, telephony system, accounting system, among others.

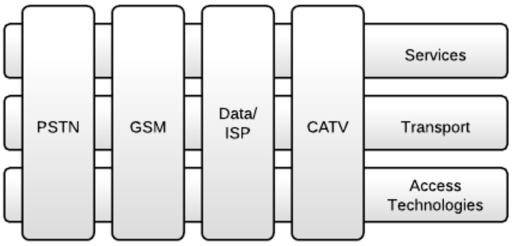


Fig. 1. Legacy telecommunication model

Nowadays the telecommunications operators possesses a close business model, with little appeal to 3^{rd} party developers. Because of this the telecommunications operators are becoming simple bit pipes [5]. With the current trend the only source profit of a telecommunications operator is the transport of data between the network. All profit related to value-added services are underestimated.

Currently the business paradigm of the telecommunications operators is evolving towards an open model, similar to the Internet. Internet provides several free services that can be acceded from practical every network available, this is possible because the services are not bound to a specific operator.

The evolution in the business model of the telecommunication operators is driven by the transition between OSA/Parlay architecture to 3GPP/IMS [1,12] architecture. 3GPP/IMS architecture was proposed by 3GPPP (3rd Generation Partnership Project) for the third generation of mobile telephony. It follows the principles of IETF (Internet Engineering Task Force) in order to break the barrier between conventional telephony and Internet, one example is the adoption of SIP [8] as the signalization system instead of the traditional H323 protocol [10]. The H323 protocol provides audiovisual communication between any packet switching network, it is recommended by ITU (International Telecommunication Union). This protocol is rather complex because it requires physical gateway to communicate with common telephony network (PSTN). In the other hand the SIP protocol, developed by IETF, allows to establish interactive sessions between users. The main advantage of SIP is simplicity, it is a pure software solution that only communicates through IP protocol.

An new approach is needed in order to open the business model of the telecommunication operators. One possibility is based on Service-Oriented Architecture (SOA) [4], a common architecture in the area of Information Systems. Each resource, designated as service, is independent from others which allows it to be used and reused for several applications and functions. This methodology allows the build horizontal services, each service is built though several layers as show in Figure 2. An application can be deployed in several types of network and the only different if the layer that access the network. This origins uncoupled services and allows the composition and orchestration of new services, based on existent ones.

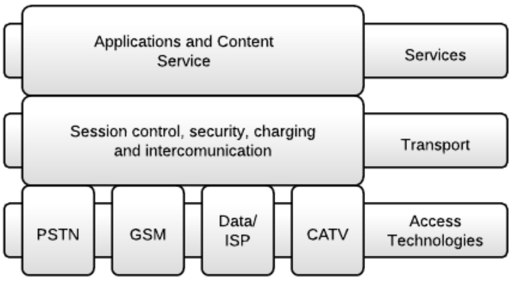

Applications and Content Service
Services
Session control, security, charging and intercomunication
Transport
PSTN
GSM
Data/ ISP
CATV
Access Technologies

Fig. 2. Next generation telecommunication model.

By opening the business model, the telecommunications operators obtains a new source of income, as they provide a series of services that can be combined or orchestrated into new services developed by 3^{rd} party developers. In other words the Walled Garden was opened and a new series of well defined and independents services is provided to 3^{rd} party developers. This allows 3^{rd} party developers and telecommunications operators coexist and make profit together, without increasing the risk to the telecommunication operator.

In this paper it is presented a novel 3^{rd} party service built through composition of services on top of 3GPP/IMS platform. The service is designated by SmartAd, an advertisement distribution services. SmartAd advertises publicity in a new channel through next generation network. The paper is divided as follows, Section 2 presents the state of the art of NGN. Section 3 presents the novel service propose in this paper. Section 4 describes the details of the implementation of the service. Section 5 presents the technological and economic viability of SmartAd. Section 6 presents the conclusions and future work.

2 Related work

The migration of telephonic services to the Internet is a reality, in this the process the telecommunication operators become only bit pipes. Unable to combat the actual tendency the telecommunication operators had to adapt, opening the Walled Garden and adopting a open business model similar to the Internet. This migration is motivated by two principal reasons. The fixed-mobile convergence, unification of fixed communication with mobile communication, this allows the reducing the maintenance cost of the network since it is only need to maintenance one physical network instead of two. The adoption of horizontal services allows wider profit margins, since it is a open model any developer can create and deploy new services on top of the existing ones. Other advantage is that a service developed to a NGN can be accessed though a conventional computer, a mobile phone of any telecommunications operator, tablet or any device that was Internet connection. This means that this services can reach a higher number of clients.

There are several efforts in the creation of a uniform platform to develop, deploy and distribute services into a NGN. As previously referred one of the most relevant and successful attempts to creates an uniform platform was the 3GPP/IMS platform [1]. The SPICE platform [13] is a extension of the 3GPP/IMS platform that hides the complexity associated to network convergence and allows to efficiently distribute services composed from the basic provided services.

There are already several services deployed into NGN, one interesting example is IPTV [3, 7] given origin to services commonly known as Triple Play. In this services there only one network is used to send and receive Internet data, telephony and television signal.

Amjad Akkawi et.al. [2] developed a gaming platform based on 3GPP/IMS. This platform allows the games to flexibly adapts to the network conditions and reserve resources, in order to improve user experience. The authors proposes a platform with several basic services, such as chat rooms, scoreboards and other elements present in several multi-player games. Developers can then develop games, that in this platform are perceived as services, this allows the game to be a composition of the basic services. The platform is responsible for the communication between the players and the ensures quality of service to them, without any intervention from the game developers.

A. De Rogatis et.al. [6] proposes a service that allows a client to automatically pay a vehicle parking with a mobile terminal. The service monitors the client's terminal as they enter the parking lot and connects to the service, accounts the duration of the parking based on the time that the terminal is reachable through the network installed in the parking lot. In the presented prototype the authors assume that the terminal is left inside the vehicle. When the vehicle leaves the parking lot, the service can not reach the mobile terminal and discount the parking value from the credit of the client.

Other important technologies, that have being used to achieve innovative business model are SMS, almost every person knows how to send and receive them. This makes SMS one of the most simplest interfaces possible between persons and services. The usage of SMS as a mean of communication to a business is well known [11, 16]. Are used to alert the client about determinate event, for example sports result, birthdays and meetings. Other usage of SMS are queering services for diverse types of information, a well known example is Paginas Amarelas, from 2003 offers a the possibility for search based on SMS exchange.

Nowadays there are several services that allows the user to customize the the waiting tone of their mobile terminals, examples of these services are TMN Waiting Ring [1] and Vodafone Welcome Tone [2]. Those services allows the receiver to personalize the waiting tone that the caller will ear.

[1] http://www.tmn.pt/portal/site/tmn

[2] http://www.vodafone.com/content/index.html

3 SmartAd

SmartAd is a new multimedia service build using SOA methodology and that takes advantage of the capacities of the NGN, in particular pure software solution, using voice over IP (VoIP) and geographic location of the caller. This novel service explores a new opportunity for business, devalued in telecommunication business. Each time that a client makes a call, he listen the monotonous awaiting tone, or in some recent cases some type of pre-selected audio, selected by the recipient. Examples of this type of service is TMN Waiting Ring and Vodafone Welcome Tone. Based on this situation a new business idea was born, occupy this time slot with some sort of lucrative audio.

The main idea behind the proposed service is each time a client makes a call, the call is deviated and instead of the waiting tone it will be reproduced audio advertisements that has especially selected to the caller. After the advertisement the call is redirected to the original recipient as shows in Figure 3. As an example lets assume the user Alice pretends to call the user Bob:

1. Call is established.
2. The call is deviated by the service, that selects one audio advertisement based in the context of user Alice.
3. User Alice listens the advertisement instead of the normal waiting tone.
4. At the end of the advertisement the call is redirected to the user Bob, becoming a conventional call.

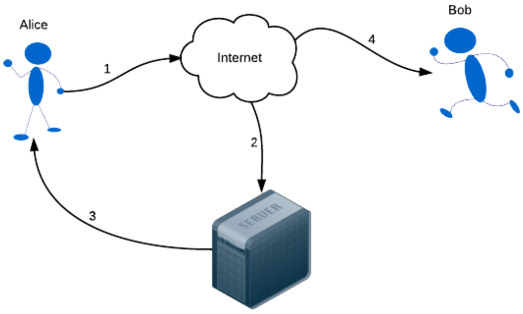


Fig. 3. Example of the proposed service.

All this process is seamless to the users, the only difference between this service and a normal call is the existence of audio publicity. This is a very important aspect since the users are reluctant to change common habits. If the service introduced a significant change in the process of making a call only a small part of the users would subscribe the service. One important note, emergency calls (for example, calls to police, fireman, emergency number) have a private share of the bandwidth. So the proposed service will not interfere with emergency calls.

Other functionality of the service is request information, enterprise's geographic position, by SMS. An client sends one SMS with an keyword of his interest and the service searches its internal repository of enterprises and returns a list of enterprises that match the keyword sorted by the geographic distance between the client and the enterprises. Through this functionality it is possible to infer some preference of the client.

One advantage of the service is the use of context information in the process of selection of audio advertisements. In this service, context information means geographic position of the client and his preferences collected through the SMS information service. This concept could be extended with other sources of information, for example several social networks, such as Facebook [3] or MySpace [4].

The service offers opportunities to the telecommunications operators, as previously stated, but also advantages to clients. Since the telecommunication operators have a new source of profit from the publicity agencies, they can reduce the price of the calls. It is appealing to the advertisement agencies, since this service offers a wider target audience than conventional advertisement channels and allows to pinpoint the correct advertisement to a specific client, instead of the common mass attack. The details about the duration of the advertisement or the distribution of the service among the clients is discussed in Section 5

4 Implementation

To demonstrate the technological viability of the proposed service a prototype was developed. The service can be roughly divisible into four different modules and a main application that composes the modules into a service. Each single module provides a different functionality to the service, while the main application combines the functionalities of each module and delivers a coherent service. The main applications holds the business logic.

The service was developed following the SOA methodology, which means each module is independent, consistent and can be re-used as many times as needed. Figure 4 shows the modular structure of the proposed service.

The geographic location module is responsible for register and update the geographic position of the clients. This is a service that could be provided by the telecommunication operator. But a service from a operator is costly and for the purpose of prototyping this simple module is enough.

The SMS center module is responsible for the reception and sending of SMS's. It allows the clients to communicate with the service to search enterprises.

The session manager module is responsible for deviate and redirect the calls. It deviates a call to the media gateway that reproduces the selected advertisement to that client, after the call is re-routed to the original recipient.

[3] https://www.facebook.com/

[4] http://www.myspace.com/

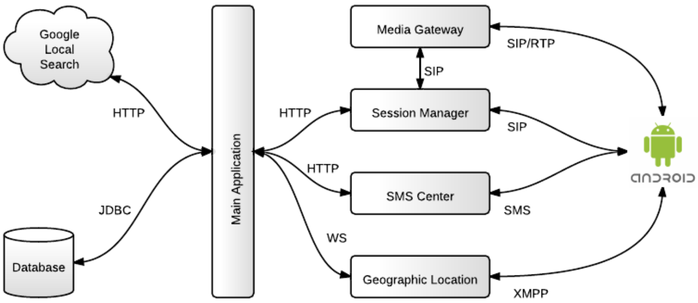


Fig. 4. Modular structure of the proposed service. Each module represents a independent service.

As the name suggests the media gateway module is responsible to reproduce audio publicity in the mobile terminal of the client.

SmartAd is deployed onto 3GPP/IMS platform, one abstraction of the network that provides functionalities to distribute multimedia content. For the purpose of the prototype the implementation selected was OpenIMS [5] since it is a open source implementation that can be deployed in only one desktop computer. The signalling protocol that was used is SIP, since it is the protocol used by 3GPP/IMS.

In the client side the application was deployed into a Android system for two reasons. First it provides GPS that facilitates the development of the geographic location module. Second there are several applications in android market to establish SIP calls.

4.1 Main application

The main application composes the SmartAd service by combining all the previously referred models. It possesses the business logic and all the information needed in order to select for each client the best advertisement available. A website was also developed for the purpose of add new enterprises and theirs advertisements into the service.

The selected database to store companies was Apache derby[6], but other relational database could be used. Each entry in the database has the name of the company, a list of keywords related to the company and the geography position of the company. Each company have a list of associated audio advertisements, and each of the audio advertisement can have a time period of reproduction specified by the company. This time period allows the company to decided in which part of the day they want the advertisement to be reproduce, for example advertisement for restaurant have bigger impact at lunch and dinner time.

[5] http://www.openimscore.org/

[6] http://db.apache.org/derby/

As previously stated the main application is responsible for the selection of the correct advertisement based in the context of the client. In SmartAd service, context is defined by three different types of information:

1. Geographic position of the caller
2. Time of the call
3. Preferences of the caller

Based on the previously mention information the advertisement for one caller is selected as follows. The list is filtered based on the time period of the advertisement, only remain advertisements that are valid to the time of the call. The list of advertisement is ordered by geographic distance based on the caller and the enterprise. Finally the list is again filtered based on the preferences of the caller, for this prototype a very simple algorithm was devise. Each caller registered in the system has a weigh for each type of enterprise available, and based on the queries of the user into the service the weights are updated. For example a caller that usually query the system for sport shops, stadiums and other sport related companies, the weight related with sport will increase. For the filtered list of advertisements one is randomly selected with a condition probability based on the weights.

4.2 Geographic location module

As previously stated telecommunication operators do not provide the geographic location of the caller for free. For this reason it is necessary a service that allows to register and query the geographic location of the callers. There are two different possibilities to achieve this goal. First option, the mobile terminal periodically sends to the service it geographic location. This approach has a major drawback that is selecting an period of update, a large period means less messages to the system but bigger errors in the position of the mobile terminal. Small period means a more accurate positioning of the mobile terminal but largely increases the number of messages send to the system. Second option is adopting the publish/subscriber method. The mobile terminal has a daemon that sends its geographic location every time it changes. The biggest advantage of this method is it energetic efficiency, that is especially important with battery powered devices.

XMPP [15] was the protocol selected to transmit the position of the mobile terminal to the system. This protocol supports the publish/subscriber method and is based on XML messages which makes a very light and extensible protocol.

The architecture of the module is showed in Figure 5. Each time the GPS of the mobile terminal, generates a change position event the developed daemon communicates the new position to the system. The geographic position server has two principal functions, listens to the updates from the mobile terminals and provides a web service that allows to query the geographic position of a terminal. This functionally is used by the main application of SmartAd in the process of selection the advertisement to a caller.

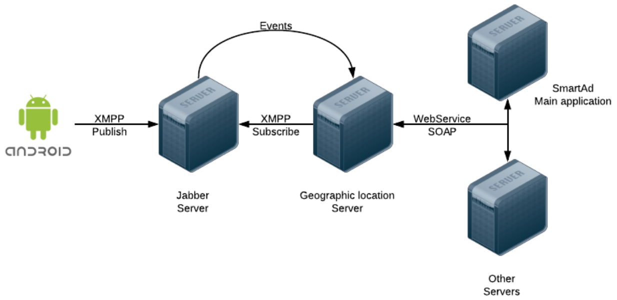


Fig. 5. Architecture of the geographic position module.

4.3 SMS center

To build a SMS center it is necessary a SMS gateway, for this prototype it was selected Kannel [7]. It is open-source and allows to receive and send SMS's through a 3G board connected to a desktop computer.

Each time the SMS gateway receives a message it queries the main application of SmartAd for the keyword present in the message. At the same time the preferences of that client are updated. The main application searches in its internal repository of enterprises and creates a list of enterprises that match the keyword send by the client. If there isn't enough enterprises in the internal database, the main application searches the keyword in google. After extracting the list of enterprises, it is sorted by geographic distance to the client and sent back to the SMS gateway. The SMS gateway then sends the a SMS wit the reply to the client.

4.4 Session manager

One of the most important components of the service is the session manager module. The 3GPP/IMS platform uses SIP for signalling and session management, the session manager module has to have the ability to manage a SIP session in order to deviate SIP calls.

Each time the application server receives a SIP call, deviates it to the session manager. The session manager query the main application of the SmartAd service in order to select the most appropriate advertisement for the caller. The session manager communicates to the media gateway the advertisement that should be reproduced in that call. Deviates the call to the media gateway that behaves as a normal recipient and begins transmit the advertisement. At the end of the advertisement the session manager module redirects the call to the original recipient and on this point beyond it becomes a conventional SIP call.

[7] http://www.kannel.org/

4.5 Media gateway

The media gateway needs the capacity to convert audio files into flows of RTP (Real-time Transport Protocol) packets, in order to transport multimedia content through IP network.

The communication between the media gateway and the session manager is made through SIP messages, there is no explicit communication between the media gateway and the main application.

The media getaway awaits for a SIP Message indicating what advertisement should be reproduced in the next SIP call. When it receives a SIP message it loads the selected advertisement to the media pipeline and waits a SIP Invite that marks the beginning of the SIP call. When receives a SIP Invite, processes the Session Description Protocol (SDP) that describes the multimedia session and starts transmitting the advertisement converted to the negotiated codec. At the end of advertisement it send a SIP Bye to the session manager, to mark the end of the advertisement.

5 Results

The described prototype was developed and tested. There were used five computers, connected through a wireless network. Each desktop computer runs one of the modules, and one of them also runs OpenIMS. The client was simulated with a Htc Magic android mobile.

Each module was tested individually with positive results. Finally all modules were combine in one service. Five dummy enterprises were added to the service and several queries were effectuated to the service in order to develop a profile to the client. After that several calls were made in order to test the advertisement reproduction. The service behave correctly.

Its important to refer that in this paper a new service is proposed and prototype was build just to prove the concept. Only a qualitative analysis was performed to demonstrate the technological viability.

5.1 Economical viability

The proposed service is a novel advertisement distribution service. The service provide several advantages over the conventional advertisement distribution channels (television, radio and Internet). Allows to pinpoint advertisement to clients, instead of the conventional mass advertisement distribution. Provides extensive and accurate statistics of advertisement reach. This solution do not requires a new infrastructure to distribute advertisement, it uses mobile terminals and theirs networks.

Some disadvantages were also identified. Since is a completely new approach is difficult to predict the reaction of the clients. It uses the structure of telecommunications operator, so it depends the success of the operator.

Several positions for the service were considered, service provider, advertisement enterprise and mobile virtual network operator (MVNO). But the most economical favourable position was service provider associated to a marketing department of a telecommunication operator with a revenue share model, as show in Figure 6. The marketing department of the telecommunication operator manages the advertisers. Using the described service the advertisements are distributed to the clients registered in the operator. One important advantage of this position is that all information of the clients is protected inside the telecommunication operator. With this position all the external relations are manage by the operator.

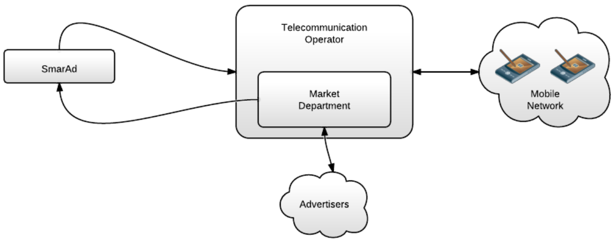


Fig. 6. Position of the service into a telecommunication operator.

Several enterprises were contacted to answer to a questionnaire in order to understand what is the value of the service. From the questionnaires 31% of the enterprises is willing to paid 10 cents for 30 seconds of advertisement. All the other companies respond that 30 seconds of advertisement worth more than 10 cents. Based on these values a economical study was performed. For each contact the advertisers pay 10 cents to the telecommunication operator, 95% of the profit of the service is for the telecommunication operator. Was considered Optimus, the smallest telecommunications operator in Portugal[8], and only a small part of the operator's clients adheres to the service. Even with this conditions at the third year the investment is paid and the service produces profit to the developers, while providing a desirable income to the telecommunication operator.

6 Conclusion

It was presented a novel service, designated by SmartAd, that advertises enterprises through a new channel of communication. The service allows to pinpoint advertisements to each caller while tries to minimize the intrusion level.

A prototype was developed in order to verify the technological viability of the service, as previously described the prototype worked correctly. A economical studied was also performed in order to verify the economical viability of the project, as previously described it appears to be a profitable service.

[8] based on financial report 2010 of ANACOM(http://www.anacom.pt)

The next step should be improving the prototype, in order to be used on a server with a real workload from callers. While the prototype demonstrates that technologically it is possible to develop the proposed service, the dimension of clients introduces new requirements to the system. It is also necessary to contact a telecommunications operator in order to subscribe the necessary services to compose the proposed system.

References

1. 3GPP: Ip-multimedia subsystem (January 2012), http://www.3gpp.org/article/ims
2. Akkawi, A., Schaller, S., Wellnitz, O., Wolf, L.: A mobile gaming platform for the ims. In: Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games. pp. 77 –84. NetGames '04, ACM (2004)
3. Al-Hezmi, A., Carvalho de Gouveia, F., Sher, M., Friedrich, O., Magedanz, T.: Provisioning ims-based seamless triple play services over different access networks. Network Operations and Management Symposium, 2008. NOMS 2008. IEEE pp. 927 –930 (April 2008)
4. Bell, M.: Service-Oriented Modeling: Service Analysis, Design, and Architecture. Wiley Publishing (2008)
5. Cuevas, A., Moreno, J., Vidales, P., Einsiedler, H.: The ims service platform: a solution for next-generation network operators to be more than bit pipes. Communications Magazine, IEEE 44(8), 75 –81 (August 2006)
6. De Rogatis, A., Garufi, R., Robustelli, A., Addesso, P., Longo, M.: An automatic-payment parking service integrated within the 3g-ims architecture. Networking and Services, 2009. ICNS '09. Fifth International Conference on pp. 1 –6 (April 2009)
7. Friedrich, O., Al-Hezmi, A., Arbanowski, S., Magedanz, T.: Next generation iptv services for an extended ims architecture. Autonomous Decentralized Systems, 2007. ISADS '07. Eighth International Symposium on pp. 429 –436 (March 2007)
8. IETF: Session initiation protocol (sip) (January 2012), http://datatracker.ietf.org/wg/sip/charter/
9. ITU: Definition of next generation network (January 2012), http://www.itu.int/ITU-T/studygroups/com13/ngn2004/working_definition.html
10. ITU: H.323 forum (January 2012), http://www.h323forum.org/
11. Krishnamurthy, N.: Using sms to deliver location-based services. Personal Wireless Communications, 2002 IEEE International Conference on pp. 177 – 181 (December 2002)
12. Poikselka, M., Niemi, A., Khartabil, H., Mayer, G.: The IMS: IP Multimedia Concepts and Services. John Wiley & Sons (2006)
13. Tarkoma, S., Bhushan, B., Kovacs, E., van Kranenburg, H., Postmann, E., Seidl, R., Zhdanova, A.V.: Spice: A service platform for future mobile ims services. World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a pp. 1 –8 (June 2007)
14. Unmehopa, M., Vemuri, K., Bennett, A.: Parlay / OSA: From Standards to Reality. John Wiley & Sons (2006)
15. XMPP: Xmpp standards foundation. http://xmpp.org/ (January 2012)
16. Xu, H., Teo, H.H., Wang, H.: Foundations of sms commerce success: lessons from sms messaging and co-opetition. System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on p. 10 pp. (January 2003)

Time-aware collaborative fitering: a review

João Vinagre

Faculdade de Ciências, Universidade do Porto, Portugal
joaoms@joaoms.com

Abstract. Collaborative Filtering (CF) has been a popular research topic in the field of recommender systems. Traditional CF algorithms rely on the assumption that data is stationary and disregards the dynamic nature of usage data. However, user preferences typically change over time, and traditional static CF algorithms do not have the ability to reflect these changes. In the past few years a new generation of CF algorithms has emerged that considers temporal effects. This paper provides a short survey on these time-aware CF algorithms and challenges for the near future.

Keywords: Recommender systems, Collaborative filtering, Data Streams

1 Introduction

In the past few years, an increasing number of researchers have focused on Recommender Systems. Collaborative Filtering (CF) is one of the most popular recommendation methods and has been successfully used in a large number of applications, such as e-commerce websites [24] and on-line communities in a series of domains [30,15,37]. The Netflix competition [3] has motivated many notorious contributions from the scientific community.

Although great effort has been put in the development of new algorithms and the improvement of CF methods, many challenges are still present, mainly related with the accuracy and scalability of CF recommenders. These challenges are mainly caused by the huge amount of sparse data needed to feed CF algorithms. As these algorithms become more scalable and accurate, new issues are now being approached. This paper focuses on one of these issues that is the dynamic nature of usage data. It is reasonable to assume that user preferences change over time, causing temporal effects that have not yet been considered until very recently. We review the literature on state-of-the-art algorithms that try to reflect the temporal dynamics in CF recommenders.

The remainder of this paper is structured as follows. In the next section we provide an introduction to traditional CF methods. Section 3 introduces the key concepts in the field of data stream mining and their relation to CF. In section 4 we review the literature in time-aware CF. A brief comparative study is presented in section 5. In section 6 we discuss possible trends and future challenges in the field. Finally, we conclude in section 7.

2 Classic collaborative filtering methods

In virtual communities such as music or movie websites there is usually a large number of users that browse through items in the system. Items can be movies, cd's, books, restaurants or any other kind of product of interest. In many of such systems, users are allowed to give their personal opinion about items, either explicitly – e.g. using a "like" button or a 5 star rating scale – or implicitly – e.g. number of times a user listens to a music track, or whether a user has bought some item or not. Suppose a system has n users and m items. By collecting ratings from users in a system, it is possible to build a ratings matrix $R_{n \times m}$ containing all ratings given by users to items. This matrix is the starting point for CF. Typically R is a sparse matrix, meaning that there is a very high proportion of missing values – users usually rate only a small part of the items in the system. The core task of CF algorithms is to predict those missing values. Recommendations are usually returned as a ranked list of the items with the highest predicted ratings.

2.1 Neighbourhood methods

Neighbourhood-based CF algorithms take rows or columns of R and compute correlations between them to obtain a similarity value. If the rows of R are the users and the columns correspond to items, similarity between two users u and v is obtained computing the correlation between the rows corresponding to those users, $R(u)$ and $R(v)$. Similarity between two items i and j can be obtained computing the correlation between the columns corresponding to those items $R^T(i)$ and $R^T(j)$. In the first case we are using a user-based algorithm and in the latter we are using an item-based algorithm.

User-based CF Given two users u and v, the similarity between them is given by some metric. The most commonly used similarity metrics are the Pearson Correlation and the Cosine measure.

Cosine measure For two users u and v, the cosine measure considers two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ in a space with dimension equal to the number of items rated by u and v:

$$sim(u,v) = \cos(\boldsymbol{u},\boldsymbol{v}) = \frac{\boldsymbol{u} \cdot \boldsymbol{v}}{||\boldsymbol{u}|| \times ||\boldsymbol{v}||} = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2} \sqrt{\sum_{i \in I_v} r_{vi}^2}} \tag{1}$$

where $\boldsymbol{u} \cdot \boldsymbol{v}$ represents the dot product between $\boldsymbol{u}$ and $\boldsymbol{v}$, I_u and I_v are the sets of items rated by u and v respectively and $I_{uv} = I_u \cap I_v$ is the set of items rated by both users u and v.

A common problem with the cosine is that different users may use the rating scale differently. For example, in a system with a rating scale of integers from 1 to 5, one user can interpret the value 3 as a positive rating, while another user can see it as negative rating. This means that different preference levels can be expressed using the same value. Conversely, equal preference levels may result in different ratings.

Pearson Correlation For users u and v, the Pearson Correlation is given by:

$$sim(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}} \tag{2}$$

where $\bar{r}_u$ and $\bar{r}_v$ are the average ratings given by users u and v, respectively. These averages have a normalizing effect on ratings given by different users, minimizing the scale interpretation problem.

Rating prediction To compute a rating prediction $\hat{r}_{ui}$ given by the user u to item i, an aggregating function is used that combines the ratings given to i by the subset $U' \subseteq U$ of the k users most similar to u. Two examples of this function are [1,4]:

$$\hat{r}_{ui} = n \sum_{u' \in U'} sim(u,u') r_{u'i} \tag{3}$$

$$\hat{r}_{ui} = \bar{r}_u + n \sum_{u' \in U'} sim(u,u')(r_{u'i} - \bar{r}_{u'}) \tag{4}$$

Equation (3) performs a weighted average, in which weights are given by the similarities between u and u'. Equation (4) incorporates the average ratings given by u and u' to i in order to minimize differences in how users use the rating scale. In both equations (3) and (4) n is used as a normalizing factor.

Item-based CF Similarity between items can also be explored to provide recommendations [35,24]. Although item-based CF differs very little from user-based CF, there is an important difference: item-based CF retrieves recommendations directly from the similarity matrix and does not require that the matrix R is kept in memory.

The main motivation for the use of item-based algorithms is that most systems have a larger number of users than items. In such systems the dimension of the similarity matrix is significantly reduced using item-based CF [35,24]. Cosine and Pearson Correlation can be used as item-based similarity measures:

Cosine measure Let U_i and U_j be the set of users that rated items i and j respectively, and $U_{ij} = U_i \cap U_j$ the set of users that co-rated *both* items i and j. The similarity between i and j is given by the cosine of the angle formed by vectors $\boldsymbol{i}$ and $\boldsymbol{j}$, whose coordinates are the ratings given by all users to each of the items:

$$sim(i,j) = \cos(\boldsymbol{i},\boldsymbol{j}) = \frac{\boldsymbol{i} \cdot \boldsymbol{j}}{||\boldsymbol{i}|| \times ||\boldsymbol{j}||} = \frac{\sum_{u \in U_{ij}} r_{ui} r_{uj}}{\sqrt{\sum_{u \in U_i} r_{ui}^2} \sqrt{\sum_{u \in U_j} r_{uj}^2}} \tag{5}$$

Pearson Correlation The Pearson Correlation, as a measure of similarity between i and j is given by:

$$sim(i,j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_i)^2} \sqrt{\sum_{u \in U_{ij}} (r_{uj} - \bar{r}_j)^2}} \tag{6}$$

where $\bar{r}_i$ and $\bar{r}_j$ are the average ratings of i and j, respectively.

Rating prediction According to [35], the prediction of the rating given u to item i is obtained like this: let J be the set of items j closest to i. Then,

$$\hat{r}_{ui} = \frac{\sum_{j \in J} sim(i,j) r_{uj}}{\sum_{j \in J} sim(i,j)} \tag{7}$$

Equation (7) is the weighted average of the ratings given by u_a to the items similar to i. The similarity values $sim(i,j)$ are the weighting factors.

2.2 Matrix factorization methods

Two properties of the ratings matrix R motivate the use of matrix factorization methods. These are its high dimensionality and its sparseness. Factorization algorithms provide a way to obtain lower rank and denser matrices than the original matrix R, with minimal loss of information. The most popular matrix factorization method used in CF is Singular Value Decomposition (SVD).

Singular Value Decomposition Singular Value Decomposition (SVD) is a matrix factorization method that can be used to produce low rank approximations to a matrix, in a similar fashion to Latent Semantic Indexing (LSI) [8]. Given a $m \times n$ matrix R with rank s, its SVD is given by:

$$R = A \times \Sigma \times B^T \tag{8}$$

where A and B are orthogonal matrices with dimension $m \times s$ and $n \times s$ respectively (B^T is $s \times n$), and Σ is a $s \times s$ diagonal matrix containing only s non-zero values $(\sigma_1, \sigma_2, \ldots, \sigma_s)$ in its diagonal, called the *singular values.* These singular values are positive and in non-increasing order, such that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_s > 0$.

If we take the $s' < s$ highest singular values in Σ we obtain a new matrix $\Sigma_{s'}$ with rank s'. Reducing matrices A and B accordingly, we obtain a new pseudo-ranking matrix $R_{s'}$, also with rank s':

$$R_{s'} = A_{s'} \times \Sigma_{s'} \times B_{s'}^T \tag{9}$$

The dimension of $R_{s'}$ is controlled by the parameter s' that acts as a trade-off between reduction amount and information loss. Low values of s' produce low dimension matrices, but with high information loss. It is known, however, that $R_{s'}$ is the best approximation to R with rank s'. The optimal s' can be obtained by cross-validation or using optimization.

Rating prediction In a CF algorithm, if R is the original matrix, A spans the user space and B spans the item space. We can then predict the unknown rating of user u to item i as:

$$\hat{r}_{ui} = \bar{r}_u + A_{s'}\sqrt{\Sigma_{s'}}^T(u).\sqrt{\Sigma_{s'}}B_{s'}(i) \tag{10}$$

where $\bar{r}_u$ is the mean rating given by u.

The complexity of SVD resides on the algorithm to perform the actual decomposition, since rating predictions are made in constant time $O(1)$.

Other methods closely related to SVD are probabilistic Latent Semantic Analysis (pLSA), used in [16] and [39], and CF via Principal Component Analysis (PCA) [13].

Extensions to SVD In [33], an incremental SVD algorithm is proposed and evaluated. Regularized SVD with biases is presented in [29] and [20]. A time-aware SVD model with biases is also presented by Koren in [21]. This method is further explained in subsection 4.2.

2.3 Other methods

Other alternatives have been proposed to solve CF problems. In [4] two probabilistic strategies are presented, one is *clustering* and the other is based on Bayesian networks.

In more recent work [12,38,5], *bi-clustering* – or *co-clustering* – simultaneously clusters users and items, with gains in computational performance and without significant accuracy loss. In fact, it is frequent that users in a cluster only appreciate a specific subset of items.

Other probabilistic approaches are based on Graph Theory [2]. For systems with binary ratings, Association Rules [34] and Markov Chains [36] are also used.

3 Data streams

A data stream is a continuous and potentially unbounded flow of data. Algorithms that process data streams are constantly being fed with new data at some fixed or variable rate, posing new challenges in storage, computation and data analysis [11]. Such streams of information do not have persistent relations, as new data keeps adding up.

When processing data streams, some key differences from static datasets need to be taken in consideration. Data elements arrive on-line at unpredictable rates and order. Also, it should be assumed that data streams do not fit in the working memory. Data elements must be discarded or archived at some point.

CF algorithms take usage data to compute models and make recommendations. This usage data shares all the above characteristics and thus can be looked at as a data stream.

P. Domingos and G. Hulten [11] identify the following desirable properties for an algorithm that learns from data streams:

- It must require small time for each data element;
- Main memory requirements must be bounded and independent of the number of data elements;
- Only a single pass over the data should be necessary to build the model;
- The model must be available anytime, not only when it finishes processing the data, since data processing may never cease;
- The model should be able to adapt to drifts in the underlying concept (concept drift).

From a data stream mining perspective, time-aware CF focuses on the ability to deal with concept drift, in the sense that user preferences are assumed to drift over time and new users and items keep entering in the system.

4 Time-aware collaborative filtering

In recent years, new approaches have emerged that take into account the time dimension in ratings data. The main motivation for these new approaches is that user preferences naturally evolve over time. Moreover, new users and items keep entering the system, changing the underlying relationships. Traditional CF algorithms do not take this into account and therefore need to be frequently trained to keep up with the evolving data. Both neighbourhood-based and SVD-based models have been recently adapted to take temporal effects into consideration.

4.1 Time-aware neighbourhood models

The most natural way to adapt neighbourhood-based algorithms to temporal effects is to somehow give more relevance to recent observations, and less to past observations. This can be achieved using a series of techniques, most of which are based on either discrete time windows [27,22] or continuous decay functions [9,10,25].

Decay function algorithms Early work on time-aware neighbourhood-based CF is presented in [9], in an item-based algorithm. The authors use a monotonic decreasing time function $f(t) = e^{-\alpha t}$ in the recommendation step, giving more weight to recently rated items by a user. In function f, t is the time at which a rating was given and α is a parameter that controls the decay rate relative to time. In later work, the same authors propose a similar method in [10] using recency-based weighting. The rating prediction for an item is weighted by the most recent ratings given to similar items.

In more recent literature [25], Liu et al. introduce decaying time functions in both the similarity computation and the rating prediction steps of the item-based algorithm. For the similarity computation step, the decreasing exponential function of time $f(t) = e^{-\alpha t}$ is used with a parameter α that controls the decay rate. This function provides weights for the similarity computation, based on the cosine measure. In practice, this causes pairs of items to be less and less similar as their ratings are given farther apart in time. At the rating prediction step a similar decay function $g(t) = e^{-\beta t}$ is used to make rating predictions, using the same method as in [9]. The only difference between f and g are the decay parameters α and β – which can actually be the same. The authors argue that allowing different decay factors for similarity computation and rating prediction provides finer grain control over the algorithm. One notable feature of this algorithm is that the similarity computation is incremental and of low complexity, allowing the algorithm to perform on-line updates as ratings are continuously given by users.

Time-window algorithms Time-window algorithms work by considering only data in a window (sliding window) that contains either the latest N instances – for example, the latest 1000 ratings – or the instances contained in the latest time interval – for example, all ratings given in the last 24 hours.

A user-based neighbourhood algorithm is proposed in [27] that uses a sliding window containing a fixed number of instances. The algorithm computes similarities between the latest user sessions. Each user session consists of a number of ratings given by a user in a short period of time – for instance, during 1 hour.

A different approach using time intervals is used in [22]. The authors use a set of item-based algorithms differing only in the number of nearest neighbours considered to predict ratings. Algorithms are retrained at fixed time intervals – 7 days, with variable number of instances – with data in that same interval. Error is continuously monitored for all algorithms, and the algorithm with the lowest error so far is selected to provide recommendations.

4.2 Time-aware factorization models

Factorization algorithms have recently been adapted to cope with temporal effects. In his notable work [21], Koren extends his optimized SVD algorithm [20] to tackle temporal effects. The algorithm in [20] combines neighbourhood and factorization in a unified model. One interesting aspect is the use of user and item biases. These biases are used in the rating prediction step to reflect user and item deviations from global averages. To tackle temporal effects, these biases are computed as the time functions for user bias and for item bias. For item biases, time is split into discrete 10 week intervals (bins), and the biases are computed according to the time window in which they fall. User biases are computed using a decay function. One other decay function is used to weight past ratings at the prediction computation. Parameters for the algorithm are obtained using optimization techniques.

In [40] a factorization model is used on a three-dimensional tensor obtained by adding time factors to the ratings matrix. By splitting time in w equal length intervals and introducing the time dimension in the $m \times n$ ratings matrix R, we obtain a three-dimensional tensor T with dimension $m \times n \times w$. Each cell in T contains a rating given by a user within one time interval. The authors use Probabilistic Matrix Factorization algorithms studied in [31] and [32] and introduce a new similar algorithm with a simple constraint in the time dimension to achieve acceptable scalability with large datasets.

Another interesting work was carried out by Das et al. [6]. The proposed algorithm combines a neighbourhood model with pLSA and MinHash clustering [18] using the MapReduce [7] programming framework in a news recommender.

4.3 Data stream algorithms

In [23], Li et al. propose an approach to drifting preferences of individual users using the CVFDT algorithm [17]. This is a popular classification algorithm for high speed data streams that automatically adapts to concept drifts. The CVFDT algorithm is used to build a decision tree for each item in the dataset, given the ratings of other highly correlated items. The ratings given by users to these correlated items are used to predict the ratings for the target item. The algorithm can be extended to use item hierarchies – if they exist – with considerable improvements, and the mechanics of CVFDT provides automatic adjustment to drifts in user interests, avoiding accuracy degradation.

In the previously mentioned work [27] a second algorithm uses the TECHNO-STREAMS stream clustering algorithm [28], using a sliding window through user sessions.

5 Comparative study

For a better understanding of benefits and shortcomings of time-aware CF algorithms, we provide a comparative study, based on the characteristics of the main algorithms and results obtained in the reviewed literature. Table 5 summarizes this study.

5.1 Discussion

In terms predictive ability, all time-aware CF algorithms studied in this paper present improvements when compared to their time-unaware counterparts. However, a thorough comparison between all algorithms is not possible by reviewing the literature. This is mainly due to considerable differences in methodologies, datasets and metrics used by the authors to evaluate their algorithms.

One striking aspect of most of the aforementioned work is that run time performance and scalability are somewhat overlooked, with the notable exception of [25]. While the accuracy of CF algorithms are undoubtedly important, scalability and run time complexity performance are also major issues in this field of research, and can be decisive factors in the choice of a recommender system.

6 Future challenges

Because time-aware CF is still a very recent topic, most real-world systems still rely on static CF. This means that models have to be frequently retrained which

Approach	Algorithm	Key strategies
Neighbourhood	Ding et al. 2005 [9]	Decay function in rating prediction.
	Ding et al. 2006 [10]	Recency-based weighting in rating prediction.
	Liu et al. 2010 [25]	Decay function in both similarity computation and rating prediction.
	Nasraoui et al. 2007 [27]	Data stream approach with sliding window.
	Lathia et al. 2009 [22]	Adaptive neighbourhood size controlled by error rate.
Factorization	Koren 2010 [21]	Unified neighbourhood and SVD model with time variant biases.
	Xiong et al. 2010 [40]	3D ratings tensor with time dimension. Probabilistic tensor factorization.
	Das et al. 2007 [6]	pLSA with MinHash clustering in a MapReduce software framework.
Data streams	Li et al. 2007 [23]	CVFDT stream mining algorithm for drifting user preferences.
	Nasraoui et al. 2007 [27]	TECHNO-STREAMS stream clustering algorithm.

Table 1. Approaches to time-aware CF algorithms and main strategies.

given the large amounts of data can be a complex task. Temporal CF has the potential to improve accuracy by better reflecting the current reality. However, many open issues persist. In real-world systems, usage data keeps adding up as new users and items enter the system and new ratings are provided. It is a fundamental requirement that CF algorithms are scalable enough to cope with these increasing amounts of data. In terms of scalability, the research in recommender systems seems to be pointing at the ability of the algorithms to update their models on-line. This means that the ideal CF algorithm should be able to update the recommendation model at a faster rate than new data arrives. Thus, a data stream approach to recommender systems seems a viable way to pursue future work in the field of recommender systems.

One important aspect of the research in recommender systems is evaluation. The majority of the literature focuses on off-line accuracy and scalability evaluation using well studied evaluation protocols and metrics. However, real-world systems are usually sensitive to a larger number of environmental variables that cannot be reproduced in the laboratory. Users of on-line systems are humans with naturally biased perspectives on the quality and the utility of a recommender [26]. Moreover, the main task of a recommender system may vary considerable with the application [14]. For instance, users might be willing to sacrifice accuracy to obtain serendipitous, less obvious recommendations. In other applications, such as news recommendation, the recency of recommended items is a key factor and may be preferred to high accuracy. On the other hand, scalability issues typically have considerable investment implications. This type of factors cause algorithms with good off-line performance to not translate directly into good on-line performance. For this reason, on-line evaluation and user feedback may be determinant to the choice of algorithms and their parameters. One practical way of evaluating on-line recommenders is by conducting controlled experiments [19].

7 Conclusions

We have reviewed the relevant literature on time-aware CF. Although this is yet a recent topic, time-aware CF has shown to be competitive, and multifaceted in the sense that diverse techniques have been adopted. Temporal CF has the potential to become a standard starting point for researchers in the field of recommender systems. Challenges like the on-line evaluation and on-line performance of CF algorithms, as well as scalability improvements are the main issues to be faced in the near future.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. 17(6), 734–749 (2005)
2. Aggarwal, C.C., Wolf, J.L., Wu, K.L., Yu, P.S.: Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), August 27-31, 1998, New York City, New York, USA. pp. 201–212. AAAI Press, New York, NY, USA (1999), chairman-Fayyad, Usama and Chairman-Chaudhuri, Surajit and Chairman-Madigan, David
3. Bennett, J., Lanning, S., Netflix, N.: The netflix prize. In: In KDD Cup and Workshop in conjunction with KDD (2007)
4. Breese, J.S., Heckerman, D., Kadie, C.M.: Empirical analysis of predictive algorithms for collaborative filtering. In: Cooper, G.F., Moral, S. (eds.) UAI '98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, July 24-26, 1998, University of Wisconsin Business School, Madison, Wisconsin, USA. pp. 43–52. Morgan Kaufmann (1998)
5. de Castro, P., de Franca, F., Ferreira, H., Von Zuben, F.: Applying biclustering to perform collaborative filtering. In: Intelligent Systems Design and Applications, 2007. ISDA 2007. Seventh International Conference on. pp. 421–426 (Oct 2007)
6. Das, A., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: Williamson, C.L., Zurko, M.E., Patel-Schneider, P.F., Shenoy, P.J. (eds.) WWW. pp. 271–280. ACM (2007)
7. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. In: OSDI. pp. 137–150 (2004)
8. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. JASIS 41(6), 391–407 (1990)
9. Ding, Y., Li, X.: Time weight collaborative filtering. In: Herzog, O., Schek, H.J., Fuhr, N., Chowdhury, A., Teiken, W. (eds.) CIKM. pp. 485–492. ACM (2005)

10. Ding, Y., Li, X., Orlowska, M.E.: Recency-based collaborative filtering. In: Dobbie, G., Bailey, J. (eds.) ADC. CRPIT, vol. 49, pp. 99–107. Australian Computer Society (2006)
11. Domingos, P., Hulten, G.: Catching up with the data: Research issues in mining data streams. In: DMKD '01: Workshop on Research Issues in Data Mining and Knowledge Discovery (2001)
12. George, T., Merugu, S.: A scalable collaborative filtering framework based on co-clustering. In: ICDM 2005: Proceedings of the 5th IEEE International Conference on Data Mining, 27-30 November 2005, Houston, Texas, USA. pp. 625–628. IEEE Computer Society (2005)
13. Goldberg, K.Y., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A constant time collaborative filtering algorithm. Inf. Retr. 4(2), 133–151 (2001)
14. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22(1), 5–53 (2004)
15. Hill, W.C., Stead, L., Rosenstein, M., Furnas, G.W.: Recommending and evaluating choices in a virtual community of use. In: CHI 95 Conference Proceedings, Denver, Colorado, May 7-11, 1995. pp. 194–201 (1995)
16. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst. 22(1), 89–115 (2004)
17. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: KDD. pp. 97–106 (2001)
18. Indyk, P.: A small approximately min-wise independent family of hash functions. In: Tarjan, R.E., Warnow, T. (eds.) SODA. pp. 454–456. ACM/SIAM (1999)
19. Kohavi, R., Longbotham, R., Sommerfield, D., Henne, R.M.: Controlled experiments on the web: survey and practical guide. Data Min. Knowl. Discov. 18(1), 140–181 (2009)
20. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Li, Y., Liu, B., Sarawagi, S. (eds.) KDD. pp. 426–434. ACM (2008)
21. Koren, Y.: Collaborative filtering with temporal dynamics. In: IV, J.F.E., Fogelman-Soulié, F., Flach, P.A., Zaki, M.J. (eds.) KDD. pp. 447–456. ACM (2009)
22. Lathia, N., Hailes, S., Capra, L.: Temporal collaborative filtering with adaptive neighbourhoods. In: Allan, J., Aslam, J.A., Sanderson, M., Zhai, C., Zobel, J. (eds.) SIGIR. pp. 796–797. ACM (2009)
23. Li, X., Barajas, J.M., Ding, Y.: Collaborative filtering on streaming data with interest-drifting. Intell. Data Anal. 11(1), 75–87 (2007)
24. Linden, G., Smith, B., York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing 7(1), 76–80 (2003)
25. Liu, N.N., Zhao, M., Xiang, E.W., Yang, Q.: Online evolutionary collaborative filtering. In: Amatriain, X., Torrens, M., Resnick, P., Zanker, M. (eds.) RecSys. pp. 95–102. ACM (2010)
26. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: Olson, G.M., Jeffries, R. (eds.) Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006. pp. 1097–1101. ACM (2006)
27. Nasraoui, O., Cerwinske, J., Rojas, C., González, F.A.: Performance of recommendation systems in dynamic streaming environments. In: Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA. SIAM (2007)

28. Nasraoui, O., Uribe, C.C., Coronel, C.R., González, F.A.: Tecno-streams: Tracking evolving clusters in noisy data streams with a scalable immune system learning model. In: Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, Florida, USA. pp. 235–242. IEEE Computer Society (2003)
29. Paterek, A.: Improving regularized singular value decomposition for collaborative filtering. In: Proceedings of KDD Cup and Workshop. vol. 2007, pp. 5–8 (2007)
30. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: An open architecture for collaborative filtering of netnews. In: CSCW '94, Proceedings of the Conference on Computer Supported Cooperative Work, October 22-26, 1994, Chapel Hill, NC, USA. pp. 175–186 (1994)
31. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Platt, J.C., Koller, D., Singer, Y., Roweis, S.T. (eds.) NIPS. MIT Press (2007)
32. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using markov chain monte carlo. In: Cohen, W.W., McCallum, A., Roweis, S.T. (eds.) ICML. ACM International Conference Proceeding Series, vol. 307, pp. 880–887. ACM (2008)
33. Sarwar, B.M., Karypis, G., Konstan, J., Riedl, J.: Incremental SVD-based algorithms for highly scalable recommender systems. In: Fifth International Conference on Computer and Information Technology. pp. 27–28 (2002)
34. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: ACM Conference on Electronic Commerce. pp. 158–167 (2000)
35. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001. pp. 285–295. ACM (2001)
36. Shani, G., Heckerman, D., Brafman, R.I.: An mdp-based recommender system. Journal of Machine Learning Research 6, 1265–1295 (2005)
37. Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating "word of mouth". In: CHI 95 Conference Proceedings, Denver, Colorado, May 7-11, 1995. pp. 210–217. ACM/Addison-Wesley (1995)
38. Symeonidis, P., Nanopoulos, A., Papadopoulos, A., Manolopoulos, Y.: Nearest-biclusters collaborative filtering with constant values. In: Nasraoui, O., Spiliopoulou, M., Srivastava, J., Mobasher, B., Masand, B.M. (eds.) Advances in Web Mining and Web Usage Analysis, 8th International Workshop on Knowledge Discovery on the Web, WebKDD 2006, Philadelphia, PA, USA, August 20, 2006, Revised Papers. Lecture Notes in Computer Science, vol. 4811, pp. 36–55. Springer (2006)
39. Takacs, G., Pilaszy, I., Nemeth, B., Tikk, D.: On the gravity recommendation system. In: Proceedings of KDD Cup and Workshop. vol. 2007 (2007)
40. Xiong, L., Chen, X., Huang, T.K., Schneider, J.G., Carbonell, J.G.: Temporal collaborative filtering with bayesian probabilistic tensor factorization. In: SDM. pp. 211–222. SIAM (2010)

Session III

System Architectures

Chairman: Sara Fernandes, Anisa Allahdadi

Sharing medical imaging over the cloud services
Luis Bastião

Developing a Program Logic for the SPARK Programming Language
Eduardo Brito

Improving Logical Clocks in Riak with Dotted Version Vectors: A Case Study
Ricardo Gonçalves

List Based Task Scheduling Algorithms on Heterogeneous Systems - An overview
Hamid Arabnejad

Examining Checkpoint and Storage Schemes for Fault Tolerance in Computing Clusters
Tiago Santos

NS-3 EmuNetDevice evolution to support binding to PPP interfaces
Helder Fontes, Gustavo Carneiro and Manuel Ricardo

Sharing medical imaging over the cloud services

Luís A. Bastião Silva

University of Aveiro/IEETA
bastiao@ua.pt

Abstract. Over the past decades, healthcare institutions adopted Picture Archive and Communication Systems in their workflows. The exchange and interaction between different equipment is performed with Digital Imaging Communication in Medicine (DICOM), which is a very extensive protocol covering many areas of the imaging laboratories. However, the communication of a wide domain composed by several medical institutions is not well supported.
In the last few years, Cloud computing has been used to allow communication anytime and anywhere. This new paradigm creates new opportunities to share information that can be always available. The proposed implementation is supported on the public cloud resources that are available on the Internet, creating the opportunity to exchange information between the medical devices inside the institutions with another devices located in another institution.
Despite of the advantages of the cloud computing, it also brings new challenges regarding the data privacy when the medical data are transmitted over different domains. This paper presents a solution to share DICOM services across healthcare institutions without breaking DICOM based on cloud. A solution to tackle these issues was proposed, creating a ciphered channel between the entities that are sharing DICOM services. In this paper we explored the application of cloud computing to share medical imaging data across different institutions providing privacy and confidentiality to the involved entities.

Keywords. PACS, DICOM, Medical Imaging, Cloud computing

1 Introduction

The use of collaborative work environments has greatly increased in healthcare in the past over decade. This trend had changed in the healthcare delivery systems and the exchange of medical data across institutions is quite common in several modalities [1]. Their importance was increased due to cost-saving for the medical institutions and it can be used in different areas, such as, expertise consultation, cooperative work and sharing of images between multiple image centres.

Nowadays, PACS (Picture Archiving and Communication System) is one of most valuable tools supporting medical decision and treatment procedures. A PACS is a key point to store, retrieve and distribute medical images in the various steps of the clinical practices. Digital Imaging Communication in Medicine (DICOM) supports the distribution of the medical imaging, although this standard is oriented to a single institution. The communication of a wide domain composed by several medical institutions is still a challenge. Commonly, the image repositories or PACS archive is not shared between medical centres due to technical challenges and security concerns.

Although the DICOM standards support SSL/TLS layers, there are many medical devices that do not support these features. This creates restrictions for the users located outside the institution from accessing the PACS archive in a secure way. The medical institutions often use VPN (Virtual Private Networks) to share medical resources. However, this solution requires point-to-point configurations, which is not scalable. Other possibilities to exchange exams between medical institutions are processed through CD/DVD delivered for instance, by conventional mail or email.

Cloud computing is large used to share files over the Internet and allow users to communicate with each other using external infrastructures. This technology allows access to applications and data without any infrastructure inside the medical institutions [2]. However, there are also some important issues that must be considered during the implementation of a solution (infrastructure and/or application) in a public Cloud provider [3]. Namely, there are critical concern related with data security and privacy.

The main idea of this paper is to promote DICOM inter-institutional communications, allowing the establishment of shared workflow and exchange of documents across them. The proposed DICOM relay service aims to be a communication broker, allowing search, store and retrieve of medical images over a group of hospitals, in different sites. This solution allows, for instance, remote access to the institutional PACS archive. The communication between different islands is supported on the cloud services, but it keeps the interoperability with the devices adopted by the medical community. The proposed DICOM routing mechanism has a transparent application to end-user without any breaks with actual standards used by medical imaging devices and repositories. Finally, the architecture provides several security services associated to connections.

This paper is structured in 5 sections. The section 2 gives the background of the area and introduces concepts that are useful in the rest of the paper, section 3 presents the implemented architecture, section 4 presents the results and compares the solution with other solutions, and section 5 presents the final conclusion of the work.

2 Background

2.1 Medical workflow in collaborative scenarios

PACS presents significant advantages over traditional analogical systems based on film and also create an excellent opportunity for telemedicine, telework and collaborative work environments. Although medical digital imaging brought many benefits, it also presents new challenges for storing, indexing and sharing their data.

Currently, most devices in medical institutes follow the DICOM standard to communicate, store and visualization information. In theory, DICOM standard solved all issues regarding the communication between different collaborators, but it still has some gaps in real environments, mainly in inter-institutional cooperation, which have barriers in “many-to-many” collaboration. Teleradiology is one of important cooperative areas in medicine and it increased in the last two decades. The medical centres cannot afford specialists from all areas and it is very usual to outsource some services, including reporting of procedures produced inside institution. In same areas, there are hospitals or small centres that have technicians and acquisition devices to perform examination of specifics modalities, for instance, computer radiology (CR) or magnetic resonance (MR). However, they do not have enough radiologists, i.e. physician specialists, to report all these exams. In those cases, the remote reporting is a practice quite common. There are also others user cases, for instance, the tele-work scenarios, where healthcare professionals need to have remote access to medical repositories and information systems of their institutions.

As referred, the technologic challenges associated with the sharing of medical imaging between multi-institutions are still not solved. In the last decades there are several approaches to tackle this issue. The DICOM over email is an approach proposed in several papers [4] [5]. However, these solutions still has some associated latency due to the restrictions of email protocol. On the other hard, the Grid computing paradigm has been also explored to provide federated access to distributed image repositories [6-8]. Another approach [9] presents a similar scenario based on Cloud computing. In this case, the solution focuses on exchanging, storing and sharing medical images across the different hospital. The last two approaches are relevant, however, in both cases, a central repository is presented and there are several institutions or departments that do not intend to share or outsource the repository to outside of the institution.

Telematics platforms appear as fundamental tools to support identified services and processes. Moreover, those new technologies can be decisive in some scenarios, mainly in regions with difficult access or with few habitants.

2.2 DICOM protocol

DICOM protocol is quite important for interoperability between medical devices. The standard allow the interoperability with medical equipment of the different manufactures. This protocol works over TCP/IP granting a reliable connection between actors. Over the protocol layer, DICOM has services that follow client/server architecture. In the DICOM scenario there is Service Class Provider (SCP) and Service Client User (SCU). A DICOM equipment can have different roles, i.e. SCP or SCU, depending on the type of device. For instance a modality that produces images is responsible for storing the image in the PACS Archive. Thus, a modality is considered a SCU because it uses a service. On the other hand, PACS Archive is offering a service, and therefore it belongs to SCP. Each DICOM device has an Application Entity Title (AETitle) that identifies the DICOM device. DICOM AETitle works as an addressing mechanism, similarly to IP and port in the lower layer of the network stack.

To communicate with a DICOM device, the first step is to purpose an exchange of information, called DICOM association. In this procedure, devices negotiate several parameters for the association, such as, what kind of information will be transferred, how it is encoded and the duration of the association. After the negotiation, the service commands are executed between SCU and SCP to perform the service goal.

Storage is a service that allows the SCU to store images in a PACS Archive (C-STORE command). Basically, the modality or image generator (i.e. Storage SCU) sends the images to the PACS archive (i.e. Storage SCP) - **Fig 1**. For each image, a C-STORE Request is invoked. All the contents of the DICOM objects are inside the C-STORE request message. A C-STORE response is sent from the Storage SCP after the file be received.

Query/Retrieve is a service composed of two commands. Query allows the SCU (i.e. workstation) to search for a study or patient, using the C-FIND command (**Fig 2**). The workstation can search over the image archive using several fields like, for instance, patient name, study date and modality. **Fig 2** illustrates a query action, which looks for exams from today with names starting with A and in the response, two studies were retrieved (Antonio and Ana).

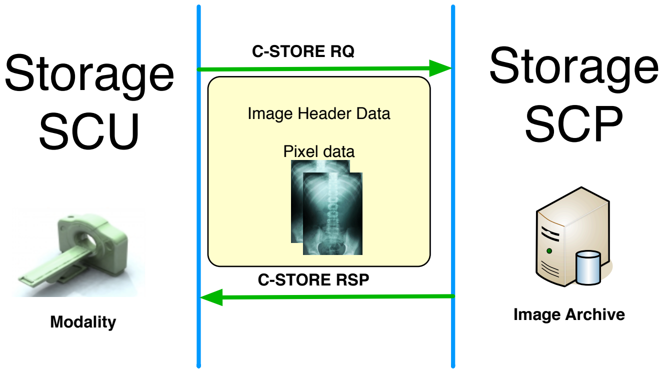


Fig 1: DICOM storage service. The Storage SCU is the client that are transferring a set of images to the PACS Archive (Storage SCP)

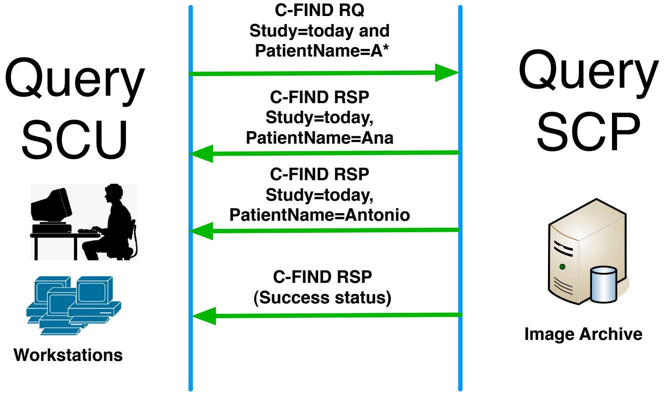


Fig 2: DICOM query (C-FIND). The Query SCU is the client that is searching the PACS repository (Query SCP).

Finally, retrieve method allows the SCU (i.e. workstation) to get/move image from the SCP (i.e. image archive) - **Fig 3**. The retrieve operation uses the C-MOVE or C-

GET command. The C-MOVE is a retrieve command that uses a C-STORE to transfer the images. The C-MOVE command does not download the images directly. Instead of transferring directly, it performs an action meaning the image archive sends the study to a specific location that typically is its own workstation.

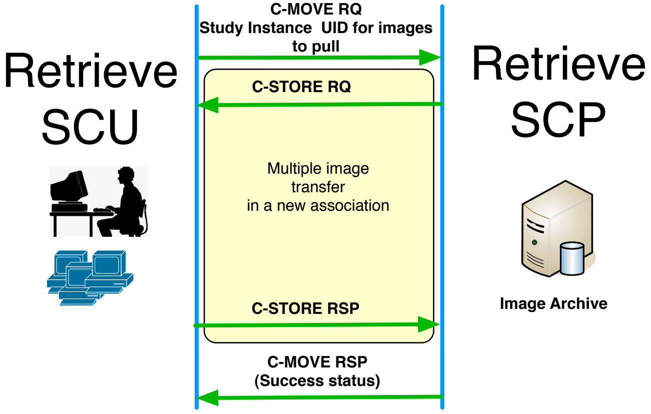

C-MOVE RQ
Study Instance UID for images to pull
Retrieve SCU
Retrieve SCP
C-STORE RQ
Multiple image transfer in a new association
C-STORE RSP
Image Archive
C-MOVE RSP (Success status)

Fig 3: DICOM retrieve image. Retrieve SCU is the client that is fetching a set of images from PACS archive (Retrieve SCP).

2.3 Cloud computing

Cloud computing is a rising technology that allows the enterprises to hold scalable resources without having any IT infrastructure. There are several cloud providers, such as, Amazon AWS, Google and Rackspace that embrace many areas, since storage, databases, and notification systems. These providers supply elastic computing power and unlimited storage [10] [11] to their customers.

There is a huge amount of interest in the IT industry to migrate services to Internet Cloud platforms [12]. In order to response to their request, many cloud companies have been created to meet their demands. There was a significant effort from Cloud providers to offer new features to clients and nowadays cloud computing is much more than a way to virtualize machines. It is an ecosystem with a range of complementary services to work well individually and together. For instance, Amazon Web Services has released many services to fulfil their customers' requirements: S3 [13], SQS, SimpleDB and many others. In turn, Google AppEngine [14], Windows Azure [15] and many others improved their solutions with new APIs to overcome the challenges of their targets.

It is evident that the computing-as-utility is a business model becoming prevalent in the electronic world and numerous institutions are adopting it. The emergence of Cloud computing providers creates a great opportunity to tackle the costs of purchase hardware and software.

The market is changing and there are new paradigms to deploy applications and to store information that are always available on the Internet. We believe that medical solutions will also adopt these new models to improve their business processes. Following the technological evolution, cloud computing has been adopted by several companies in the industry and in particular healthcare industry.

3 System architecture

In this section, an architecture to solve the problems with sharing medical images across institutions will be described. The component design and the workflows of the transfer and search over remote repositories will be presented.

3.1 Description

As explained earlier, DICOM standard is not very used to inter-institutional communication due to its limitation. Each hospital is an independent island, unable to communicate with other hospital infrastructures. The PACS integration across medical institution is an ad-hoc process, which has several barriers to deploy. Moreover, the telework can be difficult due to the restriction to access medical repositories outside of institution. In this paper we present a solution that creates an easy way to integrate different medical repositories of different institutions supported on cloud computing services.

Cloud computing is largely used to share files over the Internet. Cloud providers offer a high quality of service, mainly in the availability and scalability. Our solution takes advantage of the cloud computing services to exchange information between different locations. The communication between the components of the digital medical laboratories is mainly used through DICOM. This protocol runs over TCP/IP protocol, but contains its own addressing model through the AETitle that identifies the medical device [16]. Due to the network filters restrictions (i.e. firewall's), this communication does not perform well in WAN (Wide Area Network) scenarios. To extend the communication to different institutions, the proposed approach takes advantage of the DICOM addressing mechanism to route the information to the correct location (i.e. AETitle is the DICOM address mechanism).

The public cloud infrastructure is used as communication mechanism to support information forwarding among the involved entities through these routes. Furthermore, additional Cloud provider support is simplified due to a plugin-based system. To support abstraction with the cloud storage we developed a Cloud IO (Input/Output) stream mechanism. It allows writing in the cloud storage as a data stream. New cloud providers can easy supported, only need to implement the interfaces supplied by Cloud IO. Notification systems were used to perform communication between the several components of the architecture.

3.2 Components

The proposed DICOM relay service has two main goals: grants the secure and reliable connection between the players and create an easy solution to access the internal medical repositories anytime and anywhere. Our architecture (Fig 4) contains two software components: DICOM Bridge Router and DICOM Cloud Router, which we will explain with more details, in the sequel.

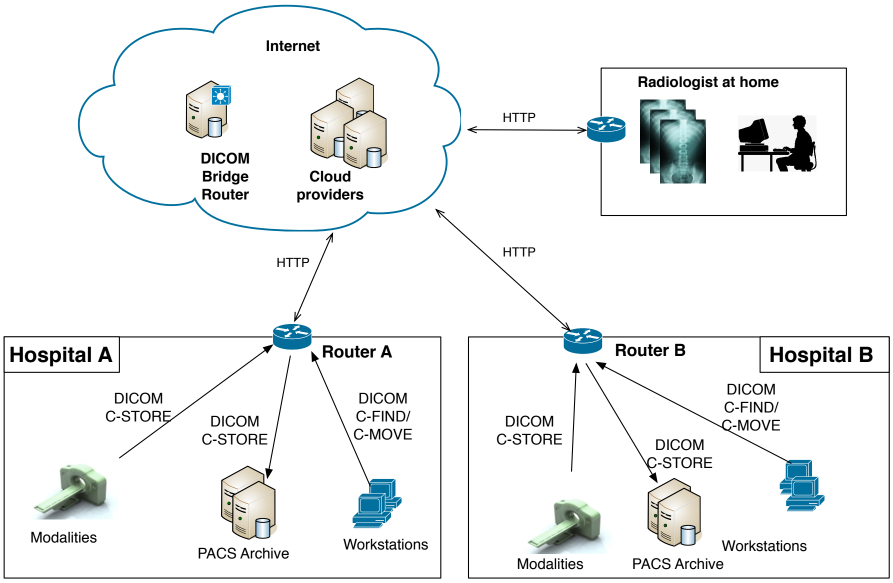


Fig 4: Architecture of the solution: Router in the boundary do the communication between the DICOM devices and the Cloud computing

DICOM Cloud Router.

The DICOM Cloud Router (Router) has the main responsibility of handling the DICOM services and forwarding messages to the correct place. To perform this process, it uses AETitle routing tables, i.e. for each AETitle belonging to the DICOM network domain, it contains associated the type of services that is providing and the username of the Router, which will allow to reach the correct router to forward the messages. In fact, manual management of those tables are actual practices because the DICOM standard does not provide a mechanism to auto-discovery of the DICOM nodes. Also, for security reasons, only allowed medical devices should be accessed from outside the medical institution and those tables also work as access control list. Thus, the Router has a graphical interface to setup the IP, port and the services available inside the medical institution.

Real world objects were mapped directly in the DICOM standard, for instance DICOM equipment is represented as a “Device” in the defined concepts of the standard. The Router supports multiple devices (i.e. as many as are online in the WAN DICOM network), each one with a different AETitle and transfer syntaxes (i.e. the data codification supported).

Finally, each medical institution or isolated DICOM network that wants to share services to the WAN DICOM network needs to run a Router inside the private network that will be saw as a standard DICOM node supporting several services (Fig 4).

DICOM Bridge Router.

The DICOM Bridge Router (Bridge), works as a relay mechanism between different DICOM Cloud Routers disperse over several locations. This component works in a partnership with the cloud providers. The huge amount of information that flows in WAN network needs to be uploaded/downloaded to the cloud providers. DICOM Bridge Router is an important part of the architecture because it stores information about all devices (i.e. AETitles) and corresponding services supported. Moreover, it has accounts from routers and a list of cloud providers that routers can use to store the temporary information. It needs to be always available over the Internet because routers need to write information in the Bridge to provide communications. It can be deployed in several places, for instance, in a private cloud detained by a medical institution or a public cloud provider. Due to privacy concerns, we strongly recommend deployment of this component in a trustable provider or in-house (i.e. medical institutions).

The network management is supported by a temporary information system and the Bridge is accessible through the web service mechanism (RESTful). It provides the credentials to validate authorized routers, AETitle of the DICOM networks and credentials to access to the cloud provider. Only validated users register on this entity can access to the DICOM WAN Network. Moreover, the Bridge is a very important component because it stores the session key used to cipher DICOM messages of an association. Thus, it should be located in a trustable location, to safeguard the proposed architecture.

The Bridge is considered the main component of the architecture because it performs the management of the relay service. It only contains a reduced amount of information, and during the dataflow it just store a minimum amount of data, i.e., the confidential shared key. The remaining information is transmitted through the cloud in a ciphered mode. It is used two different cloud services: blobstore and notification systems. The Cloud providers supply, on the one hand, temporary storage of blinded data (encrypted DICOM objects/commands) and, on the other hand, a notification service that allows us to establish communication between the routers when an event is triggered.

3.3 DICOM services workflow

The architecture was designed to support multi-centre shared repositories, for instance, a regional PACS or a network of imaging centres. In this paper we presented the implementation of two mainly used DICOM services: storage and query/retrieve. The integration of the proposed solution in the PACS workflow is effortless using the developed Router. The DICOM services allow interoperability between different manufactures, i.e. with existent devices in the institution.

The **Fig 5** presents the dataflow among devices in the medical institutions. In the Fig example, the Hospital A is accessing to the repository located in the Hospital B. The client workstation invokes the query to the Router inside their institution (i.e. Hospital A), with the AETitle of the PACS archive of Hospital B. The Router will forward the information using the cloud providers and the Bridge. The responses from PACS archive are forwarded via Router in Hospital B. The existence of a relay infrastructure is transparent to institutional DICOM devices because the Routers follow the DICOM standard rules.

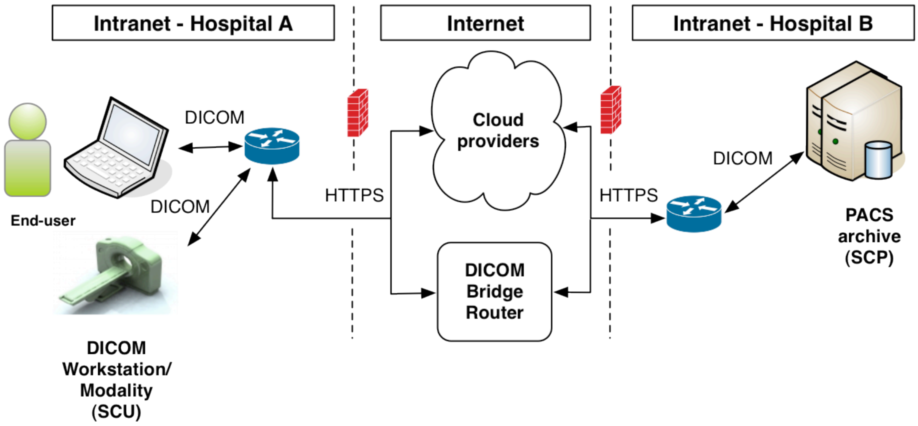


Fig 5: Basic dataflow of the architecture. Hospital A contains a workstation that accesses to the PACS archive of the Hospital B. The communication flows in the intranet to the Router, and the router forward the messages through the Cloud services and the Bridge.

The Router is able to forward the DICOM C-STORE, C-FIND and C-MOVE commands. The C-STORE is the most complex process, responsible to store images in a remote DICOM SCP, i.e. PACS repository. The medical studies in transit are stored in the cloud blobstore and they are uploaded in parallel method to improve the medical image transmission. The C-FIND service is simpler, and the DICOM messages are converted to XML and are relayed through the cloud blobstore and notification system. Finally, in C-MOVE service, the router notifies the other router, via cloud notification systems and the C-STORE command is used to process the transfer the images in inverse way, i.e. from SCP to SCU, as explained in the section 2.2.

4 Results and discussion

In order to assess the performance of DICOM relay service, a testbed were performed with two different networks in disperse locations We used several third party client workstations to test the cloud relay service, namely OsiriX [18], dcm4che2, dcmtk [19] and Conquest [20]. The prototype performs well and was able to transfer images and search over remote repositories.

The DICOM relay service has multiple benefits in the regional PACS according to their needs for teleradiology communications. As we already verified, to perform well it is necessary to have a good Internet connection in both sites. We considered that moving information is less expensive than moving the patient. The solution has the latency similar with the direct connections, and the final balance of implement a solution like that is strongly positive.

This solution has other benefits, due to easier application in a hospital and even in any computer that can work "anytime, anywhere". Moreover, actually several medical institutions still use conventional mail to transport CD/DVD, emails, etc. There are other solutions [23] based on email paradigm, but in same cases, the email is very restricted inside the medical institution. Another approach [24] was proposed, with a very well defined target: sharing neuroimaging, based on transfer techniques DICOM based. They provide a very good description of the architecture, but their target was to share other information of the workflow of neuroimaging. The proposed solution is very specific to the particular scenario and it cannot be generalized to all medical imaging modalities. Thus, our solution is generic enough to improve the workflow of this procedure and create a more easy way to share medical repositories across multiple institutions.

The proposed architecture to relay DICOM services has an unquestionable benefit, which is the interoperability between medical devices. Radiologists can work at home, in the same way that they do in the hospital, without changing their methods. They will be able to access PACS in the remote hospital without needing to waiting for the exams to arrive by email or other mechanisms.

Furthermore, for a regional PACS/Tele-imagiologic services, the proposed architecture can provide inter-institutional DICOM services and support distinct workflows:

Tele-image center: a shared repository between a group of hospitals and the PACS cloud archive meets their needs.

Remote Query/Retrieval: radiologists can perform query and retrieve to a remote PACS archive.

Auto forward auto (multiple repositories): an institution that has several distributed hospitals can have multiples repositories disperse the sites, shared with other hospitals. So all medical institutions of the same group have access to these repositories.

5 Conclusion

The presented solution allows DICOM standard communication between different medical devices located in distinct institutions. The proposed architecture allows the creation of a federated DICOM network located over distinct medical institutions, creating a unique view of all resources.

It is a fact that other solutions exist like, for instance, VPN and email. However, the former mechanism demand bureaucratic and time-consuming actions and, the latter, do not offer any privacy with regarding to the email provider.

DICOM relay service is a secure and easy to deploy in an institution and the end-user does. It does not need complex setups to start communicating with external repositories, allowing interoperability with any the DICOM standard device. Besides, the required infrastructure is not excessive because it supports its main resources on the Cloud.

6 Acknowledgements

I thank to Carlos Costa, José Luis Oliveira and bioinformatics group from Universidade de Aveiro for supervisor and infrastructure support.

7 References

1. Huang HK: PACS and imaging informatics: Basic Principles and Applications. 2004.

2. Rimal B, Choi E: A Conceptual Approach for Taxonomical Spectrum of Cloud Computing. In: *Ubiquitous Information Technologies & Applications, 2009 ICUT '09 - Proceedings of the 4th International Conference Fukuoka*: 1-6.

3. Rosenthal A, Mork P, Li MH, Stanford J, Koester D, Reynolds P: Cloud computing: a new business paradigm for biomedical information sharing. *J Biomed Inform* 2010, 43(2):342-353.

4. Ribeiro LS, Bastião L, Costa C, Oliveira JL: EMAIL-P2P GATEWAY to Distributed Medical Imaging Repositories. In: *HealthInf 2010; Spain, Valencia*.

5. Weisser G, Engelmann U, Ruggiero S, Runa A, Schrˆter A, Baur S, Walz M: Teleradiology applications with DICOM-e-mail. *European radiology* 2007, 17(5):1331-1340.

6. Yang C, Chen C, Yang M: Implementation of a medical image file accessing system in co-allocation data grids. *Future Generation Computer Systems* 2010.

7. Sharma A, Pan T, Cambazoglu BB, Gurcan M, Kurc T, Saltz J: VirtualPACS--a federating gateway to access remote image data resources over the grid. *J Digit Imaging* 2009, 22(1):1-10.

8. Liu BJ, Zhou MZ, Documet J: Utilizing data grid architecture for the backup and recovery of clinical image data. *Comput Med Imaging Graph* 2005, 29(2-3):95-102.

9. Chen C, Wang W: Implementation of a Medical Image File Accessing System on Cloud Computing. In: *Computational Science and Engineering (CSE) - 2010; Hong Kong, China*.

10. Vaquero LM, Rodero-Merino L, Caceres J, Lindner M: A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review* 2008, 39(1):50-55.

11. Oliveira D, Bai„o FA, Mattoso M: Towards a Taxonomy for Cloud Computing from an e-Science Perspective. *Cloud Computing* 2010:47-62.

12. Hajjat M, Sun X, Sung Y-WE, Maltz D, Rao S, Sripanidkulchai K, Tawarmalani M: Cloudward bound: planning for beneficial migration of enterprise applications to the cloud. *SIGCOMM Comput Commun Rev* 2010, 40(4):243-254.

13. Amazon Simple Storage Service [https://s3.amazonaws.com/] Available in: June 2011

14. Google App Engine (GAE) [http://code.google.com/appengine/] Available in: June 2011

15. Windows Azure Platform [http://www.microsoft.com/windowsazure/] Available in: June 2011

16. DICOM-P7: Digital Imaging and Communications in Medicine (DICOM), Part 7: Message Exchange. In.: National Electrical Manufacturers Association; 2009.

17. Pianykh OS: Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide: Springer; 2008.

18. Osirix DICOM Viewer [http://www.osirix-viewer.com/] Available in: 2011

19. dcmtk [http://dicom.offis.de/] Available in: 2011

20. Conquest DICOM Software [http://ingenium.home.xs4all.nl/dicom.html] Available in: June 2011

21. PubNub [http://www.pubnub.com/] Available in: June 2011

22. Costa C, Ferreira C, Bastião L, Ribeiro L, Silva A, Oliveira J: Dicoogle - an Open Source Peer-to-Peer PACS. *Journal of Digital Imaging* 2010:1-9.

23. Ribeiro LS, Costa C, Oliveira JL: A proxy of DICOM services. In: 76280L.

24. Marcus DS, Olsen TR, Ramaratnam M, Buckner RL: The extensible neuroimaging archive toolkit. *Neuroinformatics* 2007, 5(1):11-33.

Developing a Program Logic for the SPARK Programming Language

Eduardo Brito

HASLab / INESC TEC
Universidade do Minho
Campus de Gualtar, 4710-057 Braga, Portugal
edbrito@di.uminho.pt

Abstract. Ada is one of the most used programming languages for the development of software in the critical systems arena. SPARK is a well known subset of Ada, with its own toolset for software verification, that is being increasingly adopted by the industry.
Although SPARK has been enjoying industrial (and academic) success, we have found some flaws in its (semi-)formal approach to verification. In this paper we identify some of these flaws and we provide an example of the formal semantics and program logic of SPARK.
The example is presented, along with the soundness proof, including the soundness theorem that was adapted to deal with the safety conditions inherent to SPARK.

Keywords: Ada, Formal Methods, Program Logics, Program Verification, Semantics of Programming Languages

1 Introduction

Ada is a highly respected programming language in the critical systems arena and one of the most popular programming languages used for the development of high integrity software (HIS)[2,14,3,15]. It is a full-fledged programming language with concurrency and real-time support built into the core language.

SPARK[4] is a (strict) subset of the Ada programming language with its own toolset for enforcing programming practices and providing program verification capabilities. SPARK achieves this using a semi-formal approach to program verification.

In this introduction we overview the programming languages and their aims and we focus on some of the deficiencies that SPARK has as a formal approach to verification.

The Ada programming language Ada has always been developed with the aim of providing a reliable, readable and safe programming language for use in critical systems, as it was initially commissioned by the DoD[1].

[1] Department of Defence of the United States of America.

Through the years, the programming language has been revised and expanded several times (Ada 83[11][2], 95[24,25], 2005[26], 2012[21,1]), bringing new features to the language while removing (some) ambiguities in the programming language and in its Reference Manual.

Although Ada has been evolving as a general purpose programming language, for the critical systems arena, especially for HIS, strict restrictions are placed upon the features of the programming languages that can be used. These restrictions are designed as ways to avoid common programming errors and to simplify the verification and validation process, which is of the utmost importance in these systems.

The SPARK Programming Language and Toolset SPARK[3] takes into account the usual restrictions that are placed upon HIS and implements tools to analyse and enforce (some of) these practices.

It is important to note that SPARK is a strict subset. This means that all (valid) code written in the SPARK programming language corresponds to, not only a valid Ada program, but to a Ada program with the exact same semantics (and independent of compiler implementations). SPARK also supports a subset of the Ravenscar profile for real-time concurrent aplications called RavenSPARK.

SPARK advocates the early adoption of the language and tools in the software development process. Late integration in the software development process leads to programmers trying to *"sparkify"* already existing code, which has been shown to be hard and costly[6].

SPARK promotes a Correctness by Construction approach[4] to software development. Correctness by Construction tries to guarantee that programs behave as specified by guaranteeing that each individual component (in the broad sense of the word; also applied to subprograms) is implemented as a refinement of an abstract specification. In SPARK this is implemented by using Programming by Contract in the packages' body and specification, generating Verification Conditions (VC) for refinement.

The SPARK toolset consists of various tools. In this paper we shall focus on the main tools of the toolset[5]: Examiner, Simplifier and POGS. The Proof Checker, Zombiescope and SPARKBridge will not be considered in this discussion. Although they are flagship tools they are outside the scope of this article. In Fig. 1 we present the usual verification cycle used with the SPARK tools.

The Examiner[22] is responsible for analysing the syntax and static semantics of SPARK source code. It is also the VC Generator (VCGen) of the toolset[23].

[2] Hoare criticised the Ada programming language in his Turing Award speech[9]. The version he addressed is what is known as "preliminary"/"pre"-Ada/Ada 80, the first draft of the language that was revised and extensively changed into the first official standard, Ada 83.

[3] SPADE (Southampton Program Analysis and Development Environment) Ada Ratiocinative Kernel.

[4] Correctness by Construction is also advocated by other formal methods such as the B method.

[5] The SPARK version that is considered in this article is SPARK 9 GPL.

The Examiner outputs the VCs into the FDL language which are then supplied to the Simplifer.

The Simplifier is an automated theorem prover tailored for SPARK. It attempts to discharge the VCs written in FDL using its own set of rules, that take into consideration the specific features of SPARK, while additional rules may be added by the user to aid in the proof process.

Problems related to the SPARK approach Although SPARK has shown itself able to certify critical software up to Common Criteria EAL-5[20] (and EAL-6 in some cases), because it is semi-formal, it can not reach the highest level of certification from Common Criteria, EAL-7[7]. The same can be said for the future DO-178C[16].

To be a fully formal approach, not only should the language be described in a mathematical formalism (both static and dynamic semantics) but also the tools that are used in the toolset should also provide correct and verifiable formal specifications. The implementation of the tools would then have to be proven correct regarding the specifications.

Although there is previous work on the formal semantics[19] (both static and dynamic) of SPARK, it is outdated and does not reflect the actual status of the language. Furthermore, this only addresses the language and not the program logic and VCGen, which are not formalized anywhere in the SPARK literature. Without the availability of such specification, results on the soundness and completeness of the logic can not be given (the same for the VCGen).

Development of a Program Logic for SPARK In this paper we present our work on the formal specification of a subset of the SPARK programming language, which we call mSPARK[6]. Although we have formalized mSPARK's semantics and program logic and proved the program logic's soundness (although not for the full mSPARK) using our own adaptation of the soundness theorem, which accounts for safety conditions on range/constraint types, we present here only a small but illustrative part of our work. For the detailed semantics, program logic and proof, please refer to the author's thesis[5].

[6] The m stands for mini and Minho.

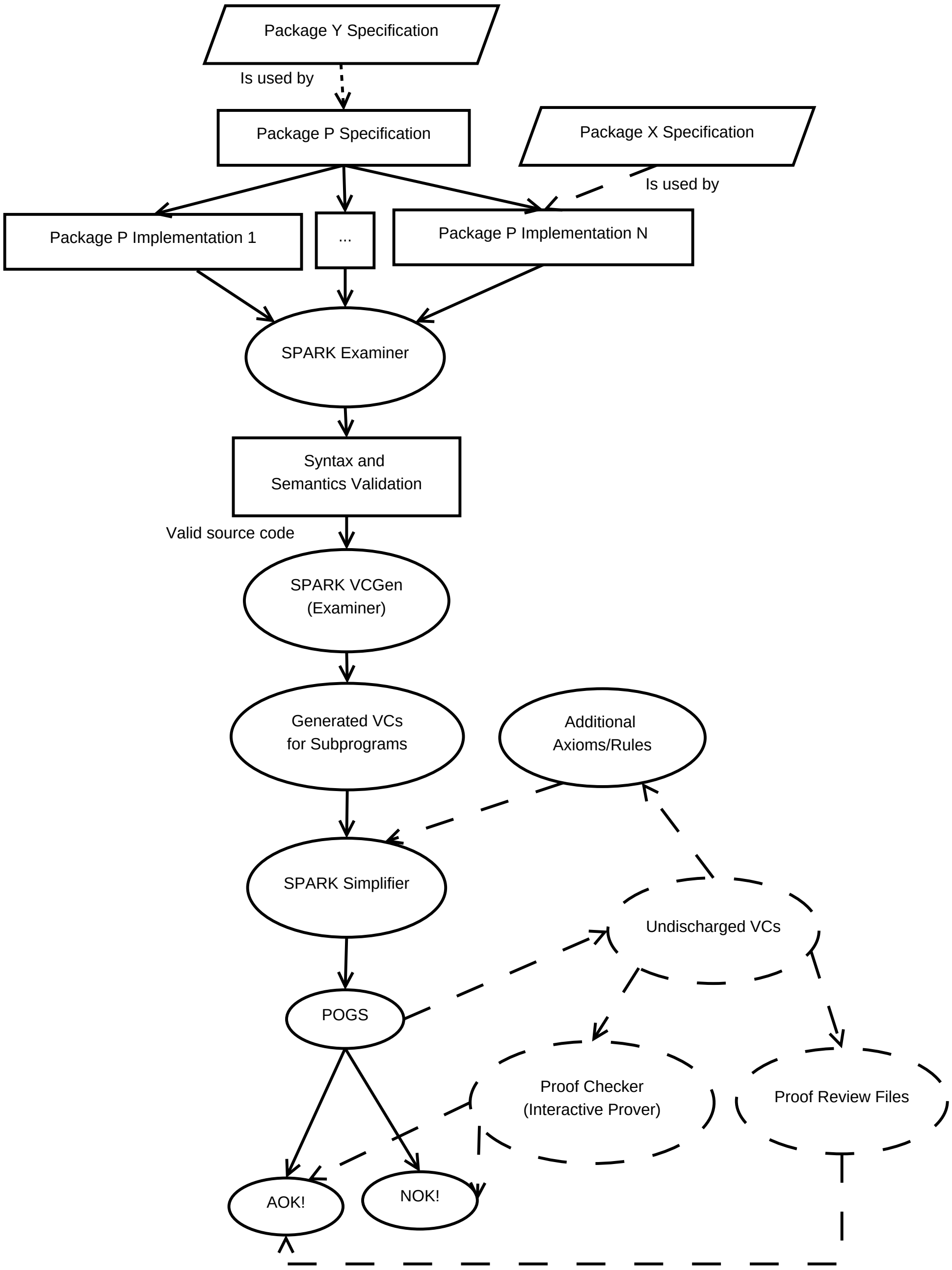


Fig. 1. Use of the SPARK toolset for program verification.

2 mSPARK - A Subset of SPARK

In this section we present our subset of the SPARK programming language, mSPARK. We start by presenting the features that were included in the subset, followed by a small discussion on why these features were chosen. We omit most of the operational semantics because of space constraints, choosing only a representative example. For further reference, this work has been previously presented in[5].

The mSPARK Programming Language As previously stated, mSPARK is a subset of SPARK. Given this, we present in Table 1 the list of features that can be found in the SPARK programming language and their status in our subset. Items marked with ✓are fully implemented, items marked with X are not implemented, and items with ± are partially implemented. We also added a column to the table, justifying some of our options.

Most features that are not implemented yet are, for our immediate purposes, not as important as the features we chose. The features that were chosen were based on our experience in developing programs in SPARK and other programming languages and also based on what we deemed to be the most important and interesting features for our first approach at SPARK formalization.

We also present a minimal example of a program written in mSPARK, which has a different syntax from SPARK (albeit very similar) to provide a small "flavour" of the programming language. This program finds the maximum of two numbers.

```
types

    type Short is range -65536 .. 65536;

variables

    x, y, res : Short;

subprograms

    function Max( a : Short; b : Short ) return Short is
        aux : Short;
    begin

        if a >= b then
            aux := a;
        else
            aux := b;
        end if;
        return aux;
    end Max;

execute

    x := 9001;
    y := 0;
    res := Max( x, y );

end;
```

Operational Semantics example The inference rule from Fig. 2 provides a representative example of the inference rules in our operational semantics of mSPARK, portraying the different safety problems that might occur during the evaluation of a command of mSPARK.

$$\frac{}{\{\phi\}\ a(e_1,\ldots,e_n) := e\ \{\psi\}} \qquad (\textsc{Assign Array})$$

$$\text{if} \models \phi \rightarrow (\mathrm{safe}(a(e_1,\ldots,e_n)) \wedge \mathrm{safe}(e) \wedge \mathrm{safe\_range}(a,e)) \text{ and } \models \phi \rightarrow \psi[a(e_1,\ldots,e_n \rhd e)/a]$$

$$\frac{[e_1,\ldots,e_n] \Downarrow_{\mathrm{Expr}(env)} [k_1,\ldots,k_n] \qquad [k_1,\ldots,k_n] \not{::} \mathit{type\_array\_ind}(env,a)}{(a(e_1,\ldots,e_n) := e,\ env) \rightsquigarrow \textbf{constraint\_error}} \qquad (\mathrm{AssignArr}_{index\_error})$$

$$\frac{\begin{array}{cc} [e_1,\ldots,e_n] \Downarrow_{\mathrm{Expr}(env)} [k_1,\ldots,k_n] & e \Downarrow_{\mathrm{Expr}(env)} k \\ [k_1,\ldots,k_n] :: \mathit{type\_array\_ind}(env,a) & k \not{::} \mathit{type\_array\_val}(env,a) \end{array}}{(a(e_1,\ldots,e_n) := e,\ env) \rightsquigarrow \textbf{constraint\_error}} \qquad (\mathrm{AssignArr}_{value\_error})$$

Fig. 2: Inference rules for array assignment with safety conditions.

The **Assign Array** inference rule is applied when all arguments of the array assignment are *safe*, including the indexing and range constraints of mSPARK types. The other two rules, **AssignArr$_{\textbf{index\_error}}$** and **AssignArr$_{\textbf{value\_error}}$**, deal with type errors on the ranges of the expressions being evaluated, regarding the declared type for indexes and arrays, respectively.

Table 1. Features of the mSPARK programming language.

Feature	mSPARK	Comments
Basic commands	✓	
Basic control structures	✓	
Extended control structures	✓	
Basic data types	±	Some types, such as fixed and floating-point and modular arithmetic are not included, for now, in the language.
Enumeration types	✓	
Range types/subtypes	✓	
Type casting	X	
Record types	X	
Subprograms	±	We allow parameterless procedures with frame conditions and pure functions (with and without parameters).
Named Parameters	X	
Assertion annotations	✓	
Data-flow annotations	±	We only use data-flow annotations for frame conditions.
Packages	±	We use a different syntax from SPARK. Also, we do not support some of SPARK's features.
Object Orientation	X	An important addition to future iterations of the language. SPARK has only partial support for it.
Refinement	X	One of the most important features we have on our "wishlist". The most important part will be to have the axiomatic semantics and VCGen for this feature.
RavenSPARK	X	Important feature for the future. We intend to have the operational and axiomatic semantics for this subset of concurrency.
Generics	X	SPARK's team has been struggling with this feature for some time. We would like to formally specify this in our language.

3 Developing a Program Logic for mSPARK

In this section we present an example of the program logic of mSPARK, the soundness theorem adapted for safety and we show how to prove the soundness of the program logic, by induction on the inference rules of the operational semantics. We start by specifying in Fig. 3 what is to safely evaluate an expression (for a restrict subset of expressions), presenting the definition of the safe predicate for any given environment (an environment contains the context information of the program such as variable declarations, types, subprograms and so on and so forth).

$$
\begin{aligned}
\llbracket \mathrm{safe}(k) \rrbracket(env) &= \top \\
\llbracket \mathrm{safe}(x) \rrbracket(env) &= \top \\
\llbracket \mathrm{safe}(-e) \rrbracket(env) &\iff \llbracket \mathrm{safe}(e) \rrbracket(env) = \top \\
\llbracket \mathrm{safe}(e_1 \;\Box\; e_2) \rrbracket(env) &\iff \llbracket \mathrm{safe}(e_1) \rrbracket(env) \wedge \llbracket \mathrm{safe}(e_2) \rrbracket(env) \wedge \\
&\qquad \mathrm{compatible\_types}(\llbracket e_1 \rrbracket(env), \llbracket e_2 \rrbracket(env)) \\
&\qquad \text{where } \Box \in \{+, -, *, =, <, <=, >, >=, /=\} \\
\llbracket \mathrm{safe}(e_1 \;/\; e_2) \rrbracket(env) &\iff \llbracket \mathrm{safe}(e_1) \rrbracket(env) \wedge \llbracket \mathrm{safe}(e_2) \rrbracket(env) \wedge \\
&\qquad \mathrm{compatible\_types}(\llbracket e_1 \rrbracket(env), \llbracket e_2 \rrbracket(env)) \\
&\qquad \wedge \llbracket e_2 \rrbracket(env) \neq 0, \text{ where } \llbracket e_2 \rrbracket(env) :: integer \\
\llbracket \mathrm{safe}(\mathbf{not}\;\; b) \rrbracket(env) &\iff \llbracket \mathrm{safe}(b) \rrbracket \\
\llbracket \mathrm{safe}(b_1 \;\mathbf{and}\; b_2) \rrbracket(env) &\iff \llbracket \mathrm{safe}(b_1) \rrbracket(env) \wedge \llbracket \mathrm{safe}(b_2) \rrbracket(env) \\
\llbracket \mathrm{safe}(b_1 \;\mathbf{or}\; b_2) \rrbracket(env) &\iff \llbracket \mathrm{safe}(b_1) \rrbracket(env) \wedge \llbracket \mathrm{safe}(b_2) \rrbracket(env) \\
\llbracket \mathrm{safe}(a(e_1, \ldots, e_n)) \rrbracket(env) &\iff \\
&\qquad \bigwedge_{i=1}^{n} (\llbracket \mathrm{safe}(e_i) \rrbracket(env) \wedge \llbracket \mathrm{safe}_{range\_ind}(a, i, e_i) \rrbracket(env)) \\
\llbracket \mathrm{safe\_range}(v, e) \rrbracket(env) &\iff \llbracket e \rrbracket(env) :: variable\_type(v, env)
\end{aligned}
$$

Fig. 3: Safe predicate interpretation.

Then, using the safe predicate definition, we introduce the program logic for array assignment in Fig. 4 and we define in Fig. 5 the meaning of an Hoare triple and its validity.

Soundness is the property that what can be stated in the program logic agrees with the operational semantics. This is defined in Theorem 1.

Theorem 1 (Soundness of mSPARK's program logic).

$$\vdash \{\phi\}\; S\; \{\psi\} \Rightarrow \models \{\phi\}\; S\; \{\psi\}$$

$$\frac{}{\{\phi\}\ a(e_1,\ldots,e_n) := e\ \{\psi\}}\quad (\textsc{Assign Array})$$

if $\models \phi \rightarrow (\mathrm{safe}(a(e_1,\ldots,e_n)) \wedge \mathrm{safe}(e) \wedge \mathrm{safe\_range}(a,e))$ and $\models \phi \rightarrow \psi[a(e_1,\ldots,e_n \rhd e)/a]$

Fig. 4: Program Logic for array assignment.

We say a Hoare tripe $\{\phi\}\ S\ \{\psi\}$ is valid iff

$$\forall env \neq \mathbf{error}\ \forall env'.\ [\![\phi]\!](env) = \top \wedge (S, env) \rightsquigarrow env' \Rightarrow env' \neq \mathbf{error} \wedge [\![\psi]\!](env') = \top$$

We use the notation $\models \{\phi\}\ S\ \{\psi\}$ to mean that $\{\phi\}\ S\ \{\psi\}$ is valid.

Fig. 5: Validity of Hoare triples.

Finally, we show how to execute the soundness proof for the running example of array assignment.

$$\frac{}{\{\phi\}\ a(e_1,\ldots,e_n) := e\ \{\psi\}}\quad (\textsc{Assign Array})$$

Proof. if $\models \phi \rightarrow (\mathrm{safe}(a(e_1,\ldots,e_n)) \wedge \mathrm{safe}(e) \wedge \mathrm{safe\_range}(a,e))$ and $\models \phi \rightarrow \psi[a(e_1,\ldots,e_n \rhd e)/a]$

We want to prove $\models \{\phi\}\ a(e_1,\ldots,e_n) := e\ \{\psi\}$, i.e.

Expanding:
$\forall env \neq \mathbf{error}\ \forall env'\ .\ [\![\phi]\!](env) = \top \wedge (a(e_1,\ldots,e_n) := e, env) \rightsquigarrow env' \Rightarrow env' \neq \mathbf{error} \wedge [\![\psi]\!](env') = \top$

Let $env \neq \mathbf{error}$ and env' be environments such that:
(i) $[\![\phi]\!](env) = \top$
(ii) $(a(e_1,\ldots,e_n) := e,\ env) \rightsquigarrow env'$

From (1) and the side condition $\models \phi \rightarrow (\mathrm{safe}(a(e_1,\ldots,e_n)) \wedge \mathrm{safe}(e) \wedge \mathrm{safe\_range}(a,e))$ we know that
(iii) $[\![\mathrm{safe}(a(e_1,\ldots,e_n))]\!](env) = \top$
(iv) $[\![\mathrm{safe}(e)]\!] = \top$
(v) $[\![\mathrm{safe\_range}(a,e)]\!] = \top$

Rule $\mathrm{AssignArr}_{index\_error}$ could have not been applied because (iii) guarantees $[k_1,\ldots,k_n] :: type\_array\_ind(env, a)$.

Rule AssignArr$_{value\_error}$ could have not been applied because (iv) guarantees $[\![\text{safe}(e)]\!] = \top$ and (v) guarantees $k :: type\_array\_val(env, a)$.

So $env' = env[a([k_1 \ldots k_n] \rhd k)/a]$. From (iii), (iv) and (v) it follows immediately that $env' \neq \mathbf{error}$, since env is assumed to be well formed and the substitution $[a([k_1 \ldots k_n] \rhd k)/a]$ is well defined.

From (i) and the side condition $\models \phi \to \psi[a(e_1, \ldots, e_n \rhd e)/a]$, follows $[\![\psi[a([k_1 \ldots k_n] \rhd k)/a]]\!](env) = \top$, but, by a typical substitution lemma (that one can shown by induction on ψ), this is equivalent to $[\![\psi]\!](env[[\![a([k_1 \ldots k_n] \rhd k)]\!](env)/x]) = \top$, i.e. $[\![\psi]\!](env') = \top$.

4 Related Work

The SPARK toolset now includes the SPARKBridge tool which takes the generated VCs of Examiner and translates them into SMT-Lib and also has native support for CVC3 and Yices. This is from the work of Paul B. Jackson on the study and expansion of Examiner and the FDL output of its VCGen[12,13], implemented in the Victor tool.

Although this work aims at proving VCs from SPARK, it does not implement its own VCGen. The aim of the project is to use the Victor tool on top of the SPARK tools, disregarding the correctness (or not) of Examiner, and translating Examiner's output into other widely accepted formats so that other theorem provers can be used.

The Hi-Lite Project from the open-DO initiative has also presented work on deductive verification based on VCs for a SPARK-based subset of Ada[17,18] named Alfa, which intends to be less restrictive than SPARK. Their approach to the VCGen is to translate Ada code (in the Alfa subset) into Why[8] and then to use the VCGen from Why and its interface to the several proof tools it supports, including proof assistants.

Both these works focus on providing translators, either for FDL or for a subset of Ada, and then use existing tools to translate/generate the VCs. In both cases, there is no functional correctness proof of the translator nor for the VCGen implemented in Why.

The approach closest to the work we developed (and are developing) is the work done by Homeier[10] where he proposes the formal verification of the verification tools themselves, although his work is aimed at a *While*[7] language.

5 Conclusions

We have presented a brief outlook on the research work that we have been developing for a subset of the SPARK language. This work provides a firm

[7] While languages are a staple of books on theories of programming languages.

theoretical background on the formal specification of a subset of a programming language aimed at the development of High Integrity Software and for the tools that may be developed for supporting this language and even full SPARK and Ada.

Although the work presented here is directed towards a very specific language, the results can be adapted for programming languages with similar features. Furthermore, the example presented in this paper hints on how semantics and program logics can be adapted to deal with safety restrictions and how to prove the soundness of a program logic in relation to its operational semantics counterpart.

6 Future Work

The program logic for the subset presented on Section 2, although completely specified, is still lacking the soundness proof for the whole subset (in [5] we provided the proof for a large set of constructs), with subprograms being the most important feature that has been left out. Work on specifying a VCGen has been carried out but it is also missing soundness and completeness proofs and has incomplete support for all features of mSPARK. Future work will tackle these theoretical aspects on the formal specification of mSPARK.

Embedding the mSPARK programming language and program logic in a proof assistant, thus providing a framework based on mechanical verification, is one of the most important directions that future work should address. Mechanical verification is less prone to errors in proofs and work on rigorous software development would benefit greatly from having a formally verified workbench for program verification.

References

1. Ada 2012 language reference manual. language and standard libraries - iso/iec 8652:201x(e) (draft 14)
2. Amey, P., Chapman, R., White, N.: Smart certification of mixed criticality systems. In: Vardanega, T., Wellings, A.J. (eds.) Ada-Europe. Lecture Notes in Computer Science, vol. 3555, pp. 144–155. Springer (2005)
3. Amey, P., Hilton, A.J.: Practical experiences of safety- and security-critical technologies. Ada User Journal 25(2), 98–106 (June 2004)
4. Barnes, J.: High Integrity Software: The SPARK Approach to Safety and Security. Addison Wesley, first edn. (March 2003)
5. Brito, E.: A formal approach for a subset of the spark programming language (2010)
6. Chapman, R.: Industrial experience with spark. Ada Lett. XX(4), 64–68 (2000)
7. Common Criteria Recognition Agreement: Common Criteria for Information Technology Security Evaluation. Part 3: Security assurance components (July 2009)
8. Filliâtre, J.C., Marché, C.: The Why/Krakatoa/Caduceus platform for deductive program verification. pp. 173–177 (2007)
9. Hoare, C.A.R.: The emperor's old clothes. Commun. ACM 24(2), 75–83 (1981)

10. Homeier, P.V., Martin, D.F.: Trustworthy tools for trustworthy programs: A verified verification condition generator. In: TPHOLs. pp. 269–284 (1994)
11. Intermetrics, Inc.: Ada Reference Manual, Language and Standard Libraries (1992)
12. Jackson, P.B., Ellis, B.J., Sharp, K.: Using SMT solvers to verify high-integrity programs. In: AFM '07: Proceedings of the second workshop on Automated formal methods. pp. 60–68. ACM, New York, NY, USA (2007)
13. Jackson, P.B., Passmore, G.O.: Proving spark verification conditions with smt solvers (December 2009)
14. Kornecki, A., Zalewski, J.: Software certification for safety-critical systems: A status report. In: Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on. pp. 665 –672 (oct 2008)
15. M.Brosgol, B.: Safety and security: Certification issues and technologies. Crosstalk: The Journal of Defense Software Engineering 21(10) (October 2008)
16. McHale, J.: Upgrade to DO-178B certification - DO-178C to address modern avionics software trends (October 2009)
17. Moy, Y.: Alfa - gnattest - gnatprove (May 2011), http://www.open-do.org/wp-content/uploads/2011/05/Hi-Lite_meeting_ALFA_20110510.pdf
18. Moy, Y.: Alfa & GNATprove: progress and future work (November 2011), http://www.open-do.org/wp-content/uploads/2011/11/gnatprove-status.pdf
19. O'Neil, I.: The formal semantics of SPARK83 (1994)
20. Praxis High Integrity Systems: Tokeneer ID Station. EAL5 Demonstrator: Summary Report (2008)
21. Schonberg, E.: Towards ada 2012: an interim report. Ada Lett. 30, 63–70 (October 2010), http://doi.acm.org/10.1145/1879097.1879080
22. SPARK Team: SPARK Examiner: The SPARK Ravenscar Profile (January 2008)
23. SPARK Team: Generation of VCs for SPARK Programs (February 2009)
24. Taft, S.T., Duff, R.A. (eds.): Ada 95 Reference Manual, Language and Standard Libraries, International Standard ISO/IEC 8652: 1995(E), Lecture Notes in Computer Science, vol. 1246. Springer (1997)
25. Taft, S.T., Duff, R.A., Brukardt, R., Plödereder, E. (eds.): Consolidated Ada Reference Manual. Language and Standard Libraries, International Standard ISO/IEC 8652/1995(E) with Technical Corrigendum 1, Lecture Notes in Computer Science, vol. 2219. Springer (2001)
26. Taft, S.T., Duff, R.A., Brukardt, R., Plödereder, E., Leroy, P.: Ada 2005 Reference Manual. Language and Standard Libraries - International Standard ISO/IEC 8652/1995 (E) with Technical Corrigendum 1 and Amendment 1, Lecture Notes in Computer Science, vol. 4348. Springer (2006)

Improving Logical Clocks in Riak with Dotted Version Vectors: A Case Study

Ricardo Gonçalves

Universidade do Minho,
Braga, Portugal,
tome@di.uminho.pt

Abstract. Major web applications need the partition-tolerance and availability of the CAP theorem for scalability purposes, thus some adopt the eventual consistent model, which sacrifices consistency. These systems must handle data divergence and conflicts that have to be carefully accounted for. Some systems have tried to use classic Version Vectors to track causality, but these reveal either scalability problems or loss of accuracy if pruning is used to prevent growth.
Dotted Version Vectors is a mechanism that deals with data versioning in eventual consistent systems, which allows accurate causality tracking and scalability, both in the number of clients and servers, while limiting vector size to replication degree.
However, theories can abstract too much of the hiding properties which difficult the implementation. We discuss the challenges faced when implementing Dotted Version Vectors in Riak - a distributed key-value database -, evaluate its behavior and performance, discuss the tradeoffs made and provide further optimizations.

Keywords: Databases, Causality, Eventual Consistency, Logical Clocks, Scalability, NoSQL, Riak

1 Introduction

There is a new generation of databases on the rise, which are rapidly gaining popularity due to increasing scalability concerns. Typically these distributed systems have restrictions placed in *Consistency*, *Availability* and *Partition-Tolerance* of the CAP theorem [2]. They were grouped in a new broad class of databases called NoSQL (Not Only SQL). Examples of databases are Google's BigTable, Amazon's Dynamo, Apache's Cassandra (based on Facebook's version) and Basho's Riak [1]. Instead of providing the ACID properties, they focus on implementing what it is called a BASE (Basically Available, Soft state, Eventually consistent) system [9]. A BASE system has weaker consistency model, focuses on availability, uses optimistic replication, thus makes it faster and easier to manage large amounts of data, while scaling horizontally. As the CAP theorem says, we can only have two of the three properties that it describes. Since virtually all

[1] `http://wiki.basho.com/Riak.html`

large-scale systems must have partition tolerance, to account for hardware and networks faults, NoSQL databases focus on the *CP* for strong consistency, *AP* for high availability, or adjustable systems.

Systems like Riak or Cassandra [5] adopt an Eventual Consistency model that sacrifices data consistency to achieve high availability and partition tolerance. This means that eventually, all system nodes will be consistent, but that might not be the true at any given time. In such systems, optimistic/aggressive replication is used to allow users to both successfully retrieve and write data from replicas, even if not all replicas are available. By relaxing the consistency level, inconsistencies are bound to occur, which have to be detected with minimum overhead. This is where Logical Clocks, introduced by Lamport [6], are useful. Logical clocks are mechanisms for tracking causality in distributed systems. Causality is the relationship between two events, where one *could* be the consequence of the other (cause-effect) [7,4]. Due to constraints in global clocks and shared memory in distributed systems, these mechanisms are used for capturing causality, thus partial ordering events. It is partial because sometimes two events cannot be ordered, in which case they are considered concurrent. While some systems have tried to use classic Version Vectors (VV) [1] to track causality, they do not scale well or lose accuracy by pruning the vector to prevent its growth. Dotted Version Vectors (DVV) [8] is a novel logical clock mechanism, that allows both accurate causality tracking and scalability both in the number of clients and servers, while limiting vector size to replication degree.

From a theoretical and abstract point of view, DVV are an evolution of VV, more scalable and flexible. But in engineering, not everything that seems conceptually better, translates into better results. The main purpose of this paper is to implement DVV in a real database, proving that it can be done without major architectural changes. Moreover, we then evaluate this new implementation with the original database and discuss their tradeoffs.

In Section 2 we provide a background in Riak and DVV that is necessary to better comprehend this paper. We explain the basics of Riak, its current logical clock implementation, the classical alternatives, their shortcomings, and finally we describe briefly the DVV mechanism. Next, in Section 3 we present the major changes that had to be done, while implementing DVV in Riak. Section 4 is where the evaluation and benchmarking of this implementation can be found. Finally, we provide conclusion and future work in Section 5.

2 Background

To understand the state of logical clocks in Riak, let us describe its basic components in this context, namely replication, system interface and data versioning. We discuss the current logical clock mechanism (VV), its shortcomings and we then briefly present an alternative: DVV - we describe in which way they differ from traditional VV.

2.1 Logical Clocks in Riak

Riak[2] is developed by Basho Technologies and is heavily influenced by Eric Brewer's CAP Theorem and Amazon's Dynamo [3]. Written mostly in Erlang, with a small amount of Javascript and C, it is a decentralized, fault-tolerant key-value store, with special orientation to document storage. Being heavily influenced by Dynamo, Riak adopts the majority of its key concepts. Being a truly fault-tolerant system, it has no single point of failure, since no machine is special or central. Next, a brief description of some relevant Riak areas.

Replication Inspired by Dynamo, Riak uses the same ring concept for replication. Consistent hashing is used to distribute and organize data. This Riak ring has a 160-bit space size, and by default has 64 partitions, each represented by virtual nodes (vnodes). Vnodes are what manages client's requests, like puts e gets. Each physical node can have several vnodes, depending both on the number of partitions the ring has and the number of physical nodes. The average number of vnodes per node can be calculated by (number of partitions)/(number of nodes). Vnodes positions in the ring are attributed at random intervals, to attempt a more evenly distribution of data across the ring. By default, the replication factor (n_val) is 3 (i.e., 3 replica vnodes per key). Also, the number of successful reads (R) and writes (W) are by default a quorum number (greater than $n\_val/2$), but can be configured depending on the consistency and availability requirement levels.

System Interface In Riak, all requests are perform over HTTP by RESTful Web Services. All requests should include the *X-Riak-ClientId* header, which can be any string that uniquely identifies the client, to track object modifications with Version Vectors (VV). Additionally, every GET request provides a context, that is meant the be given back (unmodified) in a subsequent PUT on that key. A context contains the VV.

Data Versioning Riak has two ways of resolving update conflicts on Riak objects. Riak can allow the most recent update to automatically "win" (using timestamps) or Riak can return/store (depends if it is a GET or PUT, respectively) all versions of the object. The latter gives the client the opportunity to resolve the conflict on its own. This occurs when the *allow_mult* is set to true in the bucket properties. When this property is set to false, there is a silent loss update possibility, e.g., when two clients write to the same key concurrently (almost at the same time), one of the updates is going to be discard. Hereafter, assume that *allow_mult* is always true.

Lets describe two fundamental concepts in Riak:

- Riak Object: A riak object represents the *value* in the *key-value* tuple, i.e., it contains things like metadata, the value(s) itself, the key, the logical clock, and so on. So, from now on, object is the riak object and value or values

[2] The description and tests performed on this paper are based on Riak version 0.13

are the actual data of an object that being stored, e.g., a text, a binary, an image, etc.

- Sibling: A sibling (concurrent object) is created when Riak is unable to resolve the request automatically. There are two scenarios that will create siblings inside of a single object:
  - A client writes a new object that did not come from the current local object (it is not a descendent), conflicting with the local object;
  - A client writes a new object *without* context, i.e., without a clock.

Riak uses VV to track versions of data. This is required since any node is able to receive any request, even if not replica for that key, and not every replica needs to participate (being later synchronized via read-repair or gossiping). When a new object is stored in Riak, a new VV is created and associated with it. Then, for each update, VV is incremented so that Riak can later compare two object versions and conclude:

- One object is a direct descendant of the other.
- The objects are unrelated in recent heritage (the client clock is not a descendent of the server clock), thus considered concurrent. Both values are stored in the resulting object, while both VV are merged.

Using this knowledge, Riak can possibly auto-repair out-of-sync data, or at least provide a client with an opportunity to reconcile divergent objects in an application specific manner.

Riak's implementation of VV tracks updates done by clients instead of tracking updates "written" or handle by nodes. Both are viable options, but what Riak's approach provides is what clients updated the object (and how many times), in contrast to what nodes updated the object (and how many times). Specifically, VV are a list of number of updates made per client (using *X-Riak-ClientId*), like this: $[\{client1, 3\}, \{client2, 1\}, \{client3, 2\}]$. This VV would indicate that client1 updated the object 3 times, client2 updated the object 1 time, and client3 updated the object 2 times. Timestamp data is also stored in the VV but omitted from the example for simplicity. The reason to use client IDs instead of server-side IDs in VV, is because the latter can cause silent update losses, when two or more clients concurrently update the same object on the same node [8].

There is a major difference between this and the traditional approach of using using the node's IDs, because the number of clients tends to be much greater than the actual number of nodes or replicas. Therefore, the VV would grew in size much faster, probably in an unacceptable way (both size and performance). The solution Riak adopted was to prune VV as they grow too big (or to old), by removing the oldest (timestamp wise) information. The size target for pruning is adjustable, but by default it is between 20 and 50, depending on data freshness. Removing information will not cause data loss, but it can create false conflicts. For example, when a client holds an object with an old unpruned VV and submits it to the server, where the clock was pruned, thus creating conflict, where it should not have happened.

In short, the tradeoff is this: prune to keep the size manageable, thus letting false conflicts happen. The probability of false conflict happening should not be neglected. For example, by having a large number of clients interacting with a specific object, VV can rapidly grow, thus forcing the pruning. This can lead to cases where clients have to solve false conflicts, which could be later resolved in a not so correct way, i.e., if the value that “wins”, is in fact the one that would have been removed if pruning was not applied.

2.2 Dotted Version Vectors

While pruning clocks creates false conflicts, DVV prevents unbounded size growth using server-side IDs, thus eliminating the need to prune old data. But we have also said that VV with server-side IDs can cause silent updates. DVV has the same structure as VV, but addresses conflicts in a different way, by having a special case in the vector, when concurrency occurs. If a conflict is detected, the pair that should be updated is transformed to a triple (figure 1 is an example of a possible clock increment when a conflict is detected), which conflicts with the server version. So, what this accomplishes is a representation of concurrency created by two or more writes in the same key on the same node. For more details on this, see the DVV paper [8].

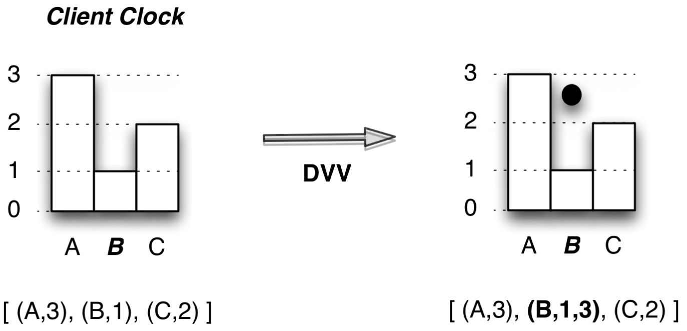


Fig. 1. A possible clock increment using DVV. Unlike VV, DVV allows non-contiguous increments in one ID per clock.

3 Implementing DVV in Riak

The first thing to be done was to implement DVV in Erlang, the programming language that Riak is written. It is a single file, that contains all the required functions. This file was placed in Riak Core, where are all the files that give core function to the system (e.g. Merkle Trees and Consistent Hashing). Then Riak KV (i.e. Riak Key-Value), which has the code for running Riak, was modified to use DDV instead of VV. This required some key changes to reflect the core differences between DVV and VV. One of them was eliminating *X-Riak-ClientId*, since we do not use the client ID anymore to update our clock. These are the main changes, in the following files:

riak_client Here we simply removed the line where the VV was previously incremented in a PUT operation.

riak_kv_put_fsm This file implements a finite-state machine, that encapsulates the PUT operation pipeline. In the initial state, we first see if the current node is a replica, and if not, we forward the request to some replica. Then, when a replica node is the coordinator, we execute the PUT operation locally first. When it is done, this replica provides the resulting object, with the updated clock and value(s), which is sent to the remaining replicas, where they synchronize their local object with this one.

riak_kv_vnode This is where the local put is done. A provided "flag" tells if this node is the coordinator, and thus the one that should do the update/sync to the clock. If this flag is false, the node will only sync the local DVV with the received one. Otherwise, this node is the coordinator, therefore it will run the *update* function with both new and local DVV, and the node ID. Then run the sync function with that resulting DVV and local DVV. Finally, the coordinator sends the results to replicas, but this time not as coordinators, thus they only run the sync function between their local object and the object provided.

riak_object This file encapsulates a Riak object, containing things like metadata, data itself, the key, the clock, and so on. Before, an object only had one clock (one VV), even if there was more than one value (i.e. conflicting values). When conflicts were detected, both VV were merged so that there was only one new VV, which dominated both. This has an obvious disadvantage: the conflicting objects could only be resolved by a newer object. Even if by the gossip between replicas, we found that we could discard some of the conflicting values that were outdated, we could not. With DVV, we change this file so that each value has its own clock. By discarding this redundant values, we are actually saving space and simplifying the complexity of operations, since we manipulate smaller data. It worth noting that this approach to have set of clocks instead of a merged clock, could also be applied to VV. Since DVV was designed to work with set of clocks, it was mandatory to change this aspect, which introduces a little more complexity to the code, but has the advantages stated above.

4 Evaluation

In order to see if the performance was really affected, if there was savings in metadata space (smaller clocks) and if false conflicts were really gone, we had to evaluate this implementation. We executed performance benchmarks comparing the original version and the DVV version. Additionally, other metrics like clock size and conflicts, to provide some insight in what was happening and why. What follows is a description of the benchmark tool, the setup, the results and finally the tradeoffs.

4.1 Basho Bench

Basho Bench is a benchmarking tool created to conduct accurate and repeatable performance and stress tests. This tool outputs the throughput (i.e. total number of operations per second, over the duration of the test) and a range of latency metrics (i.e. 95th percentile, 99th percentile, 99.9th percentile, max, median and mean latency) for each operation. Basho Bench only requires one configuration file. The major parameters that were used are:

- Duration: 20 min;
- Number of concurrent clients: 500;
- Requests per client: 1;
- Types of requests and their relative proportions: various (detailed later);
- Key Space: [0-50000];
- Key Access: Pareto distribution, i.e. 20% of the keys accessed 80% of the time;
- Value Size: fixed 1KB or 5KB;
- Initial random seed was the same for all test, to ensure equal conditions to both mechanisms, while achieve reproducible results;
- Number of replies (R and W for the read and write operations) = 2.

4.2 Setup

For these benchmarks we used seven machines, all in the same local network. A Riak cluster running on 6 similar machines, while another independent machine was simulating the clients. The request rates and number of clients were chosen to try to prevent resource exhausting, since this would create unpredictable results. Resources were monitored to prevent saturation, namely CPU, disk I/O and network bandwidth. We also used the default replication factor $n\_val = 3$.

The following types of requests were issued from clients:

- GET: a simple read operation that returns the object of a given key;
- PUT: a *blind* write, where a value is written in a given key, with no causal context supplied, i.e. without a clock. This operation will increase concurrency (create siblings) if the given key already exists, since an empty clock does not dominate any clock, thus always conflicting with the local node clock;

- UPD: an update, that is expressed by a GET returning an object and a context (clock), followed by a 50 ms delay to simulate the latency between client and server, and finally a PUT that re-supplies the context and writes a new object, which supersedes the one first acquired in the GET. This operation reduces the possible concurrency (object with multiple values) that the GET brought.

From these three core actions we evaluated two benchmarks that considered different workload mixes. The first benchmark was to do a simple generic distribution load, with the proportion of blind puts kept at 10% and interchanged proportions of 30% versus 60% for gets and updates. The size per value was fixed at 1KB.

The second benchmark was to simulate TPC-W [10] workloads, using the "Shopping Mix" (80% reads, 20% writes) with a fixed value size of 5KB, the "Ordering Mix" (50% reads, 50% writes) and the "Browsing Mix" (95% reads, 5% writes), both with 1KB per value. Reads were done with the normal GET operation, while writes were done in UPD.

4.3 Comparison of overall latency

Workload	Clock Type	Get		Put		Update		Clock	Values
		Mean	95th	Mean	95th	Mean	95th	Size	per Key
		(ms)	(ms)	(ms)	(ms)	(ms)	(ms)	(bytes)	(average)
60% GET	VV	7.65	15.9	5.71	10.1	14.4	24.0	790	1.34
10% PUT	DVV	3.16	5.25	4.31	6.27	7.76	10.9	127	1.31
30% UPD	$\frac{DVV}{VV}$	**0.41**	**0.33**	**0.76**	**0.62**	**0.54**	**0.46**	**0.16**	**0.98**
30% GET	VV	10.4	21.6	7.48	13.8	18.8	31.9	859	1.20
10% PUT	DVV	3.45	5.83	4.56	6.59	8.39	11.8	123	1.16
60% UPD	$\frac{DVV}{VV}$	**0.33**	**0.27**	**0.61**	**0.48**	**0.45**	**0.37**	**0.14**	**0.97**

Table 1. DVV and VV benchmarks with a generic approach.

The first, generic, benchmark results are in table 1, while the TPC-W approach benchmark results are in table 2. Both tables show the DVV/VV ratio that helps compare the two mechanisms, values smaller than 1.0 show an improvement and are depicted in **bold**.

In all tests we find that clock size is always (much) smaller in DVV, even with the (default) pruning that occurs with Riak VV. One can also confirm that pruning is occurring, because all the tests reveal that there were more concurrent values in the VV case. The difference in the number of values per key between the two logical clocks, results from false conflicts created by pruning. We recall that since Riak VV resort to pruning they do not reliably represent concurrency, and introduce false conflicts that need to be resolved. Having no pruning, our DVV implementation accurately tracks concurrency, while still allowing an expressive reduction of metadata size. It is easy to see that even if the default pruning activation threshold was lowered in Riak VV case, although it would reduce clock sizes, this would also lead to an increase of false concurrency and higher numbers of values per key.

Regarding performance, the generic benchmark results show that using a value payload of 1KB, the write and read operations were much better then using VV. Having less conflicts, and factoring the smaller clock size, on average operations transfer smaller data (1.8KB versus 1.2KB).

		Get		Update		Clock	Values
Workload	Clock Type	Mean	95th	Mean	95th	Size	per Key
		(ms)	(ms)	(ms)	(ms)	(bytes)	(average)
Browsing Mix	VV	2.15	3.63	5.00	7.70	159	1.00081
	DVV	2.01	3.49	5.70	8.80	89	1.00051
	$\frac{DVV}{VV}$	**0.94**	**0.96**	1.13	1.15	**0.56**	**0.99970**
Shopping Mix	VV	2.84	5.00	6.80	11.0	117	1.00066
	DVV	2.77	4.94	7.70	12.8	82.0	1.00039
	$\frac{DVV}{VV}$	**0.98**	**0.99**	1.13	1.16	**0.70**	**0.99973**
Ordering Mix	VV	7.70	16.2	14.4	24.0	682	1.00549
	DVV	2.95	4.76	7.40	10.0	113	1.00425
	$\frac{DVV}{VV}$	**0.38**	**0.29**	**0.51**	**0.42**	**0.17**	**0.99877**

Table 2. DVV and VV benchmarks with TPC-W approach.

In the TPC-W case, the first thing we can see is that concurrency (rate of conflicts, measured by values per object) is very low, as it would be expected in a more realistic setting (concurrency rates in Dynamo's paper [3] are very similar to these). Read operations were always better, or pretty even between both mechanisms. This is to be expected since the read pipeline was not modified by our implementation, but DVV is usually smaller, thus requiring less data to be transferred.

Write operations were pretty good in the ordering mix, since (like the generic approach) each value was 1KB and the difference in clock size was significant. In contrast, the browsing mix also had 1KB per value, but the difference in clock sizes was not very large (too few writes in 20 minutes for the VV clock to grew significantly, but with time, it would probably grow much larger). Then, on average, values with VV and DVV had 1.16KB and 1.09KB in size, respectively. The same can be said of the shopping mix, in this case 5.12KB and 5.08KB for the VV and DVV, respectively. Therefore, in the shopping mix and browsing mix, the difference in clock size was not sufficient to make up for the changes we had to made in the write pipeline. Simply put, using DVV in Riak, when writing some value, the coordinator has to send every conflicting value to replicas. Moreover, if the coordinator is not a replica for that key, then it has to forward the write request to a new coordinator that is also a replica. In the standard

Riak implementation, the VV case, the write pipeline is simpler, only the new client value is passed to replicas and every node can be a coordinator for any write request.

4.4 Comparison of clock sizes

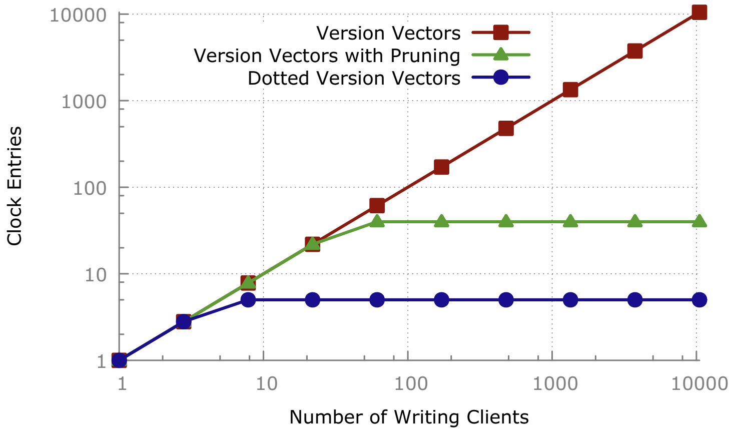


Fig. 2. Theoretical growth in clock size.

Figure 2 illustrates the theoretical effect of the number of writing clients in the number of clock entries, and thus the overall clock size. DVV stabilizes in size when the number of entries reaches the replication factor (usually 3), while VV with id-per-client grow indefinitely. Thus, in practice, systems usually resort to pruning to control its growth. In Riak's case, the imposed limit to the number of kept ids, and the trigger to pruning, is in the 20 to 50 range.

In Figure 3 we depict the evolution of the average clock size per key during the execution of the TPC-W based benchmarks in the Riak cluster. Here we can see that the DVV size becomes constant after a short time, whereas the Riak VV size is constantly increasing until stabilizing somewhere close to 1KB in the Ordering Mix case. Notice that it only stabilizes because of pruning in Riak VV, if not that it would grow linearly. The other workloads using Riak VV did not have enough time to stabilize, but would eventually be similar to the ordering mix. DVV has more or less 3 entries per clock (N=3) and size of 100 bytes, thus 1000 bytes in average for each VV means that it has roughly 30 entries, in line with the pruning range of Riak $[20 - 50]$.

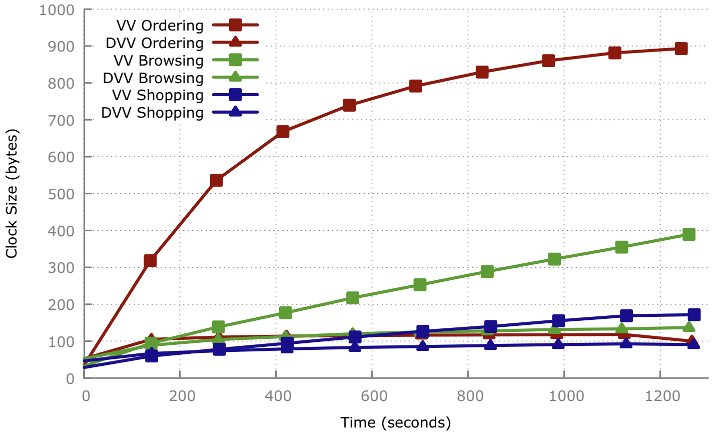


Fig. 3. Real growth in clock size using TPC-W workload mixes.

4.5 Tradeoffs

Lets resume the advantages and disadvantages of using DVV instead of VV. First, the advantages:

- **Simplify API**: since DVV uses the node's internal 160-bit ID, there is no need for clients to provide IDs, thus simplifying the API and avoiding potential ID collisions;
- **Save space**: DVV are bounded to the number of replicas, instead of the number of clients that have ever done a PUT. Since there is a small and stable number of replicas, the size of DVV would be much smaller than traditional VV;
- **Eliminates false conflicts**: clock pruning does not cause data loss, but it does cause false conflicts, where data that could be discard, is viewed as conflicting. Using DVV, the clock is bound to the number of replicas, therefore pruning is not necessary, thus eliminating false conflicts;

And now the disadvantages:

- **More complex write pipeline**: when a *non-replica* node receives a PUT request, it must forward it to a replica node. This overhead can be considerable if the transferred data is big. Which is even worse if the replica is not in the same network as the non-replica. Another thing that may affect negatively the performance is the fact that clock update and synchronization has to first be done in the coordinating replica, and then sent to the remaining replicas, whereas in VV the object goes directly to all replicas simultaneously. This is made worse when in the DVV case, the resulting object of the coordinating replica has siblings, which means that all siblings will be transferred to the remaining replicas. With VV, only the client object is sent to replicas.

5 Conclusions

Logical clocks in Riak are pruned when the number of entries exceeds some threshold, and consequently does not reliably represent concurrency, thus it introduces false conflicts. Having no pruning, DVV accurately tracks concurrency while still allowing an expressive reduction of metadata size. In terms of performance, the results showed that if we have many clients with high reads and low writes, DVV performs much better. On the other hand, fewer clients and more writes tend to mitigate DVV advantages, and in some cases it is worse than VV.

As future work, DVV performance should be addressed. The extra hop for non-replica nodes could be avoid, if we use a partition-aware client library or load balancer that knows which replica to communicate, thus reducing the response time. Another other problem: if the resulting replica coordinator's object has siblings, then it has to transfer all the siblings to the others replicas. This can be somewhat minimized if use a simple LRU cache to store keys and the corresponding clock. Since we often do not have conflicts, the coordinator can check first if the client object is more recent than the local one using the cache. If it is, we can send immediately the client object to all replicas before writing locally.

References

1. Almeida, P., Baquero, C., Fonte, V.: Version stamps-decentralized version vectors. In: Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on. pp. 544 – 551 (2002)
2. Brewer, E.A.: Towards robust distributed systems (abstract). In: PODC '00: Proceedings of the nineteenth annual ACM symposium on Principles of distributed computing. p. 7. ACM, New York, NY, USA (2000)
3. DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., Vogels, W.: Dynamo: amazon's highly available key-value store. In: SOSP '07: Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles. pp. 205–220. ACM, New York, NY, USA (2007)
4. Fidge, C.J.: Partial orders for parallel debugging. In: Workshop on Parallel and Distributed Debugging. pp. 183–194 (1988)
5. Lakshman, A., Malik, P.: Cassandra: a decentralized structured storage system. SIGOPS Oper. Syst. Rev. 44, 35–40 (April 2010)
6. Lamport, L.: Time, clocks, and the ordering of events in a distributed system. Commun. ACM 21(7), 558–565 (July 1978)
7. Mattern, F.: Virtual time and global states of distributed systems. In: Parallel and Distributed Algorithms. pp. 215–226. North-Holland (1989)
8. Preguiça, N.M., Baquero, C., Almeida, P.S., Fonte, V., Gonçalves, R.: Dotted version vectors: Logical clocks for optimistic replication. CoRR abs/1011.5808 (2010)
9. Pritchett, D.: BASE: An acid alternative. ACM Queue 6(3), 48–55 (2008)
10. (TPC)., T.P.P.C.: Tpc benchmark w(web commerce) specification version 1.8 (2002)

List Based Task Scheduling Algorithms on Heterogeneous Systems - An overview

Hamid Arabnejad

Universidade do Porto,
Faculdade de Engenharia, Dep. de Engenharia Informática,
hamid.arabnejad@fe.up.pt

Abstract. Task scheduling is key issue in obtaining high performance in heterogeneous systems. Task scheduling in heterogeneous systems is a NP-problem, therefore several heuristic approaches were proposed to solve it. These heuristics are categorized into several classes, such as list based, clustering and task duplication scheduling. Here I consider the list scheduling approach. In this paper, I will have an overview on six well-known list based scheduling algorithms (HEFT, CPOP, HCPT, HPS, PETS and lookahead) and compare the results of them.

Keywords: task scheduling , static scheduling , heterogeneous system

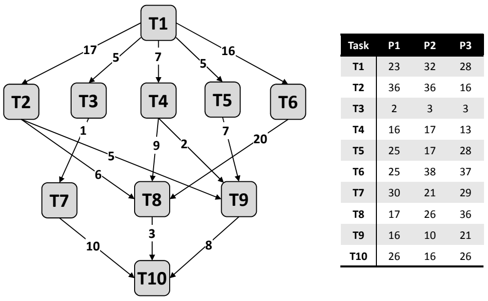
1 Introduction

A heterogeneous system can be defined as a range of different system resources, which can be local or geographically distributed, utilized to executing computationally intensive application. The efficiency of executing parallel applications on heterogeneous systems critically depends on the methods used to schedule the tasks of a parallel application. The objective is to minimize the overall completion time or *makespan*. The task scheduling problem for heterogeneous systems is more complicated than that in the homogeneous computing systems, because of the different execution rates among processors and possibly different communication rates between different processors. The DAG scheduling problem has been shown to be NP-complete [2, 3, 9], even for the homogeneous case, therefore the research effort in this field has been mainly to obtain low complexity heuristics that produce good schedules. The task scheduling problem is broadly classified in two major categories, namely Static Scheduling and Dynamic Scheduling. In Static category, all information about tasks such as execution and communication time for each task and its relation with other tasks are known before hand; in Dynamic category, such information is not available and decisions are made in runtime. In another way, Static scheduling is compile-time scheduling and Dynamic scheduling is run-time scheduling. Static scheduling algorithms are universally classified into two major groups, namely, Heuristic-based and Guided Random Search-based algorithms. Heuristic-based algorithms give near-optimal solutions but with polynomial time complexity and acceptable performance in comparison with Guided Random Search-based algorithms which give optimal solutions with exponential time complexity. The Heuristic-based group is composed by three subcategories that are: list, clustering and duplication scheduling. Clustering heuristics were mainly proposed for homogeneous systems and the aim is to form clusters of tasks that are then assigned to processors. The duplication heuristics produce the shortest *makespans* but they have two disadvantages: one is the higher time complexity, such as cubic in relation to the number of tasks; and second, they have lower efficiency because the main strategy is to duplicate the execution of tasks, resulting in more processor power used. Efficiency is an important characteristic, not only due to the energetic cost but also, in a shared resource, less efficiency means less processors available to run other concurrent applications. List scheduling heuristics, on the other hand, produce the most efficient schedules, without compromising the *makespan* and with a complexity that is, in general, quadratic in relation to the number of tasks. In this paper, I present an overview on list based scheduling algorithm for a bounded number of fully connected heterogeneous processors.

This paper is organized as follows: in Section 2, I introduce the DAG scheduling ; in Section 3, I present an overview on list based scheduling algorithms on heterogeneous systems; in Section 4, I present results of comparison for these algorithms based on several measure parameter and, finally, conclusions in Section 5.

2 DAG Scheduling

The problem addressed in this paper is the static scheduling of a single application on a heterogeneous system. An application can be represented by a *Directed Acyclic Graph* (DAG), $G = (V, E, P, W)$, as shown in Figure 1.



Task	P1	P2	P3
T1	23	32	28
T2	36	36	16
T3	2	3	3
T4	16	17	13
T5	25	17	28
T6	25	38	37
T7	30	21	29
T8	17	26	36
T9	16	10	21
T10	26	16	26

Fig. 1: Application model and computation time matrix of the tasks in each processor

Where V is the set of v nodes, and each node $v_i \in V$ represents an application task, which includes its instruction that must be executed on the same machine. E is the set of e communication edges between tasks, each $e(i, j) \in E$ represents

the task-dependency constraint such that task n_i should complete its execution before task n_j can be started. P is the set of p heterogeneous processors available in the system. W is a $v \times p$ computation cost matrix, where v is the number of tasks and p is the number of processors in the system. $w_{i,j}$ gives the estimate time to execute task v_i on machine p_j. The mean execution time of task n_i can be calculated by $\overline{w_i} = (\sum_{j \in P} w_{i,j})/p$. Each edge $e(i,j) \in E$ is associated with a non-negative weight $c_{i,j}$ representing the communication cost between the tasks n_i and n_j. Note that, when task i and j are assigned to the same processor, the real communication cost is considered to be zero because it is negligible compared to interprocessor communication costs. Additionally, in our model, I consider that processors are connected in a fully connected topology. The execution of tasks and communications with other processors can be done for each processor simultaneously and without contention. Also, the execution of any task is considered nonpreemptive. These model simplifications are common in list scheduling problem [4, 8], and I consider them in order to have a fair comparison to the state of the art algorithms.
Next, I present some of the common attributes used in task scheduling, that I will refer in the following sections.

- $\mathbf{pred(n_i)}$: denotes the set of immediate predecessors of task n_i in a given DAG. A task with no predecessors is called an *entry* task, n_{entry}. If a DAG has multiple entry tasks, a dummy entry task with zero weight and zero communication edges is added to the graph.
- $\mathbf{succ(n_i)}$: denotes the set of immediate successors of task n_i. A task with no successors is called an *exit* task, n_{exit}. Like the entry task, if a DAG has multiple exit tasks, a dummy exit task with zero weight and zero communication edges from current multiple exit tasks to this dummy node is added.
- **makespan** : it is the finish time of the exit task in the scheduled DAG, and is defined by $makespan = AFT(n_{exit})$ where $AFT(n_{exit})$ denotes the Actual Finish Time of the exit task.
- **Critical Path**$(\mathbf{CP})$: the CP of a DAG is the longest path from n_{entry} to n_{exit} in the graph. The length of this path $|CP|$ is the sum of the computation costs of the tasks and intertask communication costs along the path. The $|CP|$ value of a DAG is the lower bound of the makespan.
- $\mathbf{EST(n_i, p_j)}$: denotes the *Earliest Start Time* of a node n_i on a processor p_j
- $\mathbf{EFT(n_i, p_j)}$: denotes the *Earliest Finish Time* of a node n_i on a processor p_j

The *objective function* of the scheduling problem is to determine an assignment of tasks of a given DAG to processors such that the *Schedule Length* is minimized. After all nodes in the DAG are scheduled, the schedule length will be the *Actual Finish Time* of the exit task.

3 List-Based Scheduling Algorithms

In this section, I present a brief survey of task scheduling algorithms, specifically list based heuristics. In the past years, the research on static DAG scheduling has focused on finding suboptimal solutions to obtain a good solution in an acceptably short time. List scheduling heuristics usually generate good quality schedules at a reasonable cost. In comparison with clustering algorithms, they have lower time complexity and in comparison to task duplication strategies, their solutions use less processors, generating more efficient schedules.
A large number of list scheduling algorithms have been developed by researchers in the past. This type of scheduling algorithms has three phases: the prioritizing phase for giving a priority to each task; the selection phase for select the task based on its priority from the ready tasks in current time; and a processor selection phase for selecting a suitable processor that minimizes the heuristic cost function. If two or more tasks have equal priority, then the tie is resolved by selecting a task randomly. The two last phases are repeated until all tasks are scheduled to suitable processors.

Here, I describe the list-based scheduling heuristic algorithms, for scheduling tasks on a bounded number of heterogeneous processors, selected to be compared, namely, CPOP, HEFT, HCPT , HPS, PETS and lookahead.

3.1 CPOP Algorithm

The CPOP (Critical Path On a Processor) algorithm proposed in [8], like other list-based scheduling algorithm, has three phases, namely, *task prioritizing*, *task selection* and *processor selection* phase. In first phase, task prioritizing, the CPOP algorithm used the upward rank ($rank_u$) and downward rank ($rank_d$) to give a priority to each task in DAG. The upward rank ($rank_u$) represents the length of the longest path from the task to exit task, including the computational cost of the task and is given by $rank_u(n_i) = \overline{w_i} + \max_{n_j \in succ(n_i)}\{\overline{c_{i,j}} + rank_u(n_j)\}$ for exit task $rank_u(n_{exit}) = \overline{w_{exit}}$, and the downward rank ($rank_d$) represents the length of the longest path from a start task to the task and is given by $rank_d(n_i) = \max_{n_j \in pred(n_i)}\{rank_d(n_j) + \overline{w_j} + \overline{c_{j,i}}\}$ for entry task $rank_d(n_{entry}) = 0$

The CPOP algorithm, after calculating the upward and downward rank value for all tasks in the DAG, the priority of each task is equal to $rank_d + rank_u$. In CPOP algorithm, tasks are categorized into critical path tasks and non-critical path tasks. The CPOP algorithm defined a *CPprocessor* as the processor that minimizes the overall execution time of the critical path assuming all the critical path nodes are mapped onto it. In task selection phase, the CPOP algorithm select the task with highest priority from ready task. In the next phase, processor selection phase, if the selected task is a CP task, it is mapped to *CPprocessor*, otherwise it should be mapped to the processor that minimize its earliest finish time.

3.2 HEFT Algorithm

The HEFT (Heterogeneous Earliest Finish Time) algorithm [8] is highly competitive in that it generates a comparable schedule length to other scheduling algorithms, with a low time complexity. Like most of list-based scheduling algorithm it hast three phases. In *task prioritizing* phase, it used $rank_u$ (described before in CPOP algorithm) to assign priority to the task. In *task selection* phase, the HEFT algorithm select the task with highest priority from the ready list as the selected task. And in the *processor selection* phase, the HEFT select the processor that allows the EFT (Earliest Finish Time) of the selected task. However, the HEFT algorithm uses an insertion policy that tries to insert a task in an earliest idle time between two already scheduled tasks on a processor, if the slot is large enough to accommodate the task.

3.3 HCPT Algorithm

The HCPT (Heterogeneous Critical Parent Trees) algorithm [4] uses a new mechanism to construct the scheduling list L, instead of assigning priorities to the application tasks. HCPT divides the task graph into a set of unlisted-parent trees. The root of each unlisted-parent tree is a critical path node (CN). A CN is defined as the node that has zero difference between its $AEST$ and $ALST$. The $AEST$ is the *Average Earliest Start Time* of the task and it is equivalent to $rank_d$ as shown by $AEST(n_i) = \max_{n_j \in pred(n_i)}\{AEST(n_j) + \overline{w_j} + \overline{c_{j,i}}\}$ and for entry task $AEST(n_{entry}) = 0$. The *Average Latest Start Time* $(ALST)$ of the task can be computed recursively by traversing the DAG upward, starting from the exit task and given by $ALST(n_i) = \min_{n_j \in succ(n_i)} \left\{ALST(n_j) - \overline{c_{i,j}}\right\} - \overline{w_i}$ and for exit task $ALST(n_{exit}) =$.
The algorithm has also two phases, namely *listing tasks* and *processor assignment*. In the first phase, the algorithm starts with an empty queue L and an auxiliary stack S that contains the CNs pushed in decreasing order of their ALSTs, i.e. the entry node is on top of S. Consequently, *top*(S) is examined. If *top*(S) has an unlisted parent (i.e. has a parent not in L), then this parent is pushed on the stack S. Otherwise, *top*(S) is popped and enqueued into L. In the processor assignment phase, the algorithm tries to assign each task $n_i \in L$ to a processor p_j that allows the task to finish its execution as earlier as possible.

3.4 HPS Algorithm

The HPS (High Performance task Scheduling) [6] algorithm has three phases, namely, *level sorting*, *task prioritization* and *processor selection* phase. In the level sorting phase, the given DAG is traversed in a top-down fashion to sort tasks at each level in order to group the tasks that are independent of each other. As a result, tasks in the same level can be executed in parallel. In the task prioritization phase, priority is computed and assigned to each task using the attributes *Down Link Cost* (DLC), *Up Link Cost* (ULC) and *Link Cost* (LC) of the task. The DLC of a task is the maximum communication cost among all the immediate predecessors of the task. The DLC for all tasks at level 0 is 0. The ULC of a task is the maximum communication cost among all the immediate successors of the task. The ULC for an exit task is 0. The LC of a task is the sum of DLC, ULC and maximum LC of all its immediate predecessor tasks.

At each level, based on LC values, the task with highest LC value receives the highest priority followed by the task with next highest LC value and so on in the same level. In the processor selection phase, the processor that gives the minimum EFT for a task is selected for executing that task. It has an insertion-based policy, which considers the insertion of a task in an earliest idle time slot between two already scheduled tasks on a processor.

3.5 PETS Algorithm

The PETS (Performance Effective Task Scheduling) algorithm [5] has the same three phases as HPS. In the level sorting phase, like HPS, tasks are categorized in levels so that in each level the tasks are independent. In the task prioritization phase, priority is computed and assigned to each task using the attributes *Average Computation Cost* (ACC), *Data Transfer Cost* (DTC) and the *Rank of Predecessor Task* (RPT). The ACC of a task is the average computation cost on all the p processors. The DTC of a task n_i is the amount of communication costs incurred to transfer the data from task n_i to all its immediate successor tasks; for an exit node $DTC(n_{exit}) = 0$. The RPT of a task n_i is the highest rank of all its immediate predecessor tasks; for an entry node $RPT(n_{entry}) = 0$. The rank is computed for each task n_i based on its ACC, DTC and RPT values and is given by $rank(n_i) = round\{ACC(n_i) + DTC(n_i) + RPT(n_i)\}$.

At each level, the task with highest rank value receives the highest priority followed by the task with next highest rank value and so on. A tie is broken by selecting the task with a lower ACC value. As some of the other task scheduling algorithms, in the processor selection phase, it selects the processor that gives the minimum EFT value for executing the task. It also uses a insertion-based policy for scheduling a task in an idle slot between two previously scheduled tasks on a given processor.

3.6 Lookahead Algorithm

The Lookahead scheduling Algorithm [1] uses a new methodology to select the suitable processor for selected task in each step of scheduling. In Lookahead algorithm, first, calculate the upward rank for all tasks in a given DAG as same as HEFT. But in Processor selection phase, unlike the HEFT algorithm that select the processor based on earliest finish time for current task, the Lookahead algorithm for test each processor, first assign the selected task to the processor and then schedule the selected task children and save the maximum EFT of children as EFT selected task on the processor. After test all processor for assigning selected task to them, the lookahead select the processor with minimum EFT based on its children.

4 Experimental Result and Discussion

This section presents performance comparison of the DMCP algorithm with the algorithms presents above. For this purpose, I consider randomly generated application graphs. I first present the comparison metrics used for the performance evaluation.

4.1 Comparison Metrics

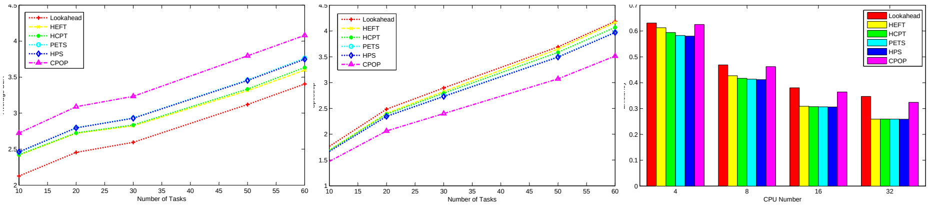
The comparison metrics are Scheduling Length Ratio, Speedup, Efficiency.

$$SLR = \frac{makespane(solution)}{\sum_{n_i \in CP_{MIN}} \min_{p_j \in P} (w_{(i,j)})} \qquad Speedup = \frac{\min\limits_{p_j \in P} \big[\sum\limits_{n_i \in V} w_{(i,j)} \big]}{makespane(solution)}$$

The denominator in SLR is the minimum computation of tasks on critical path. With any algorithm, there is no makespane less than the denominator of SLR equation. Therefore, the algorithm with lower SLR is the best algorithm. Average SLR values over several task graphs are used in our results. In *Speedup*, the sequential time is obtained by the sum of the processing time on the processor that minimizes the total computation cost [8].
For the general case, Efficiency is defined as the Speedup divided by the number of processors used $Efficiency = Speedup/\{Number\ of\ processors\ used\}$.
The DAGs used in this simulation setup were randomly generated using the program in [7] which consider the following parameters: *width* as the number of tasks on the largest level; *regularity* is the uniformity of the number of tasks in each level; *density* is the number of edges between two levels of the DAG. These parameters may vary between 0 and 1. An additional parameter, *jump*, indicates that an edge can go from level l to level $l + jump$. In this paper, I used this synthetic DAG generator for making the DAG structure which includes the specific number of nodes and their dependencies. To obtain computation and communication costs, two extra parameters are used: *CCR* and *beta*. The first parameter, CCR(Communication to Computation Ratio) is ratio of the sum of the edge weights to the sum of the node weights in a DAG; and beta(Range percentage of computation costs on processors) is the *heterogeneity factor* for processors speed. A higher value for β implies higher heterogeneity and very different computation costs among processors and a low value implies that the computation costs for a given task is almost equal among processors [8]. The average computation cost of a task n_i in a given graph $\overline{w_i}$ is selected randomly from a uniform distribution with range $\left[0, 2 \times \overline{w_{DAG}}\right]$, where $\overline{w_{DAG}}$ is the average computation cost of the given graph. The computation cost of each task n_i on each processor p_j is randomly set from the range of $\overline{w_i} \times \left(1 - \frac{\beta}{2}\right) \leq w_{i,j} \leq \overline{w_i} \times \left(1 + \frac{\beta}{2}\right)$.
In this paper, I consider DAGs with 10, 20, 30, 40, 50 and 60 tasks; the number of processors equal to 4, 8, 16 and 32; CCR of 0.1, 0.5, 0.8, 1, 2, 5 and 10; width equal to 0.1, 0.4, 0.8; regularity equal to 0.2,0.8 ; density equal to 0.2, 0.8; Beta equal to 0.1, 0.2, 0.5, 1 and 2 ; and jumps of 1, 2, and 4. These combinations

give 30,240 different DAG types. Since 10 random DAGs were generate for each combination, the total number of DAGs used in our experiment was 302,400.

Figure 2 shows the results of SLR, Speedup and Efficiency produced by the Lookahead, HEFT, HCPT and CPOP algorithm. We can see that Lookahead has the lower SLR for all DAG sizes. Consequently, Lookahead achieved better Speedups. The second best algorithm in terms of SLR is HEFT, as was referred before to be the state of art algorithm so far. Attending to Efficiency, Lookahead is also the best one. CPOP that follows very closely Lookahead in terms of Efficiency, has the poor SLR.



(a) (b) (c)

Fig. 2: (a) Average SLR; (b) average SpeedUP and (c) Efficiency comparison for Random Graphs

If I want to discuss about the result with respect to hardware feature, I should compare results for different values for CCR and beta (heterogeneity factor).

Figure 3 shows the SLR values for each algorithm with respect to different values of CCR. As shown, in lower CCR values, the improvement is not significant and HCPT, HEFT and Lookahead algorithm have close SLR values but in higher CCR, Lookahead algorithm shows better SLR than the other algorithms.

Also, Figure 4 show the same condition in improvement like as CCR for beta (heterogeneity factor); In lower beta, all algorithms have very closely same SLR values ; But with increasing beta parameter, as shown, Lookahead shows better improvement in terms of SLR and then HEFT and HCPT.

5 Conclusion

In this paper, we had an overview on the most well-know list-based scheduling algorithms on heterogeneous systems. My result shows, among of all these algorithms, Lookahead shows better performance in terms of SLR, SpeedUp and efficiency in overall But if we want review them based on hardware feature, as shown in Figure 3 for lower CCR all algorithms have same performance in term of SLR and also in Figure 4, as like CCR factor, shown for lower heterogeneity factor, all algorithms have same SLR values.

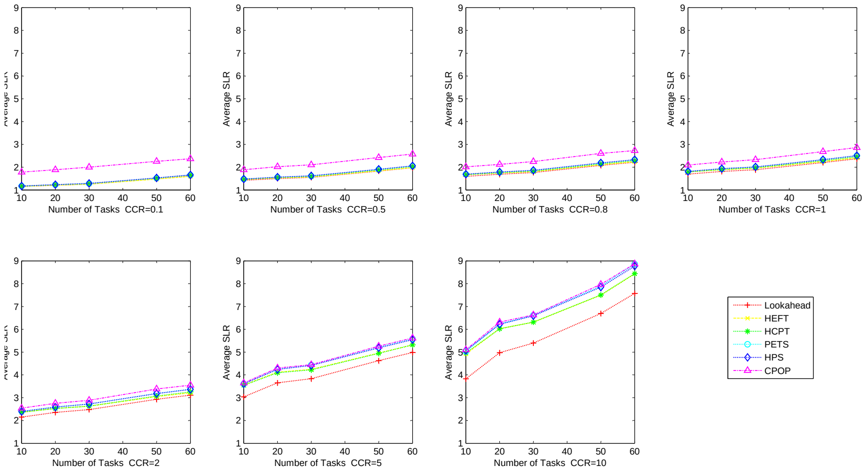


Fig. 3: Average SLR with respect to different CCR values

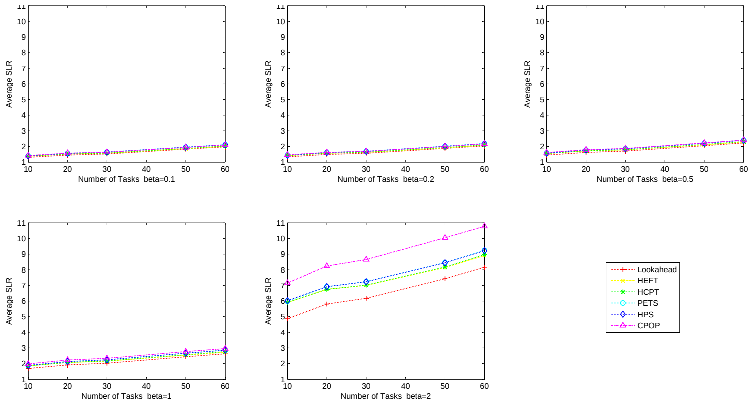


Fig. 4: Average SLR with respect to different beta (heterogeneity factor values

References

1. Luiz F. Bittencourt, Rizos Sakellariou, and Edmundo R. M. Madeira. DAG scheduling using a lookahead variant of the heterogeneous earliest finish time algorithm. In *PDP*, pages 27–34. IEEE Computer Society, 2010.
2. E. G. Coffman, editor. *Computer and Job-Shop Scheduling Theory*. Wiley, 1976.
3. M. R. Garey and D. S. Johnson. *Computers and intractability; a guide to the theory of NP-completeness*. W.H. Freeman, 1979.
4. Tarek Hagras and Jan Janecek. A simple scheduling heuristic for heterogeneous computing environments. In *ISPDC*, pages 104–110. IEEE Computer Society, 2003.
5. E. Ilavarasan and P. Thambidurai. Low complexity performance effective task scheduling algorithm for heterogeneous computing environments. *Journal of Computer Sciences*, 3(2):94–103, 2007.

6. E. Ilavarasan, P. Thambidurai, and R. Mahilmannan. High performance task scheduling algorithm for heterogeneous computing system. volume 3719 of *Lecture Notes in Computer Science*, pages 193–203. Springer, 2005.
7. DAG Generation Program. http://www.loria.fr/˜suter/dags.html, 2010.
8. Haluk Topcuoglu, Salim Hariri, and Min-You Wu. Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, PDS-13(3):260–274, March 2002.
9. J.D. Ullman. Np-complete scheduling problems. *Journal of Computer and System Sciences*, 10(3):384–393, 1975.

Examining Checkpoint and Storage Schemes for Fault Tolerance in Computing Clusters

Tiago Santos and Jorge Barbosa

Faculdade de Engenharia da Universidade do Porto, Porto
{pro09027,jbarbosa}@fe.up.pt
http://www.fe.up.pt

Abstract. This paper presents several methods used in the storage of checkpointing images adopted by fault tolerant systems. Some problematic issues associated with many of these methods are also indicated. Not only clusters but any computing system suffers from unexpected failures which can cause the loss of valuable computations results and data. Periodic checkpoints of both computations and data and their safeguard in reliable storage provide an easy way to provide fault tolerance. However each method brings challenges and drawbacks that depend not only on the application but also on the underlying hardware. By exploring the major methods, this paper provides a light characterization of the most important checkpoint and storage schemes available. From this characterization we conclude that for clusters of workstations the best checkpoint/storage choice is a group-based non-blocking checkpoint and a hybrid storage scheme.

Keywords: Fault tolerance, checkpoints, storage, overheads, parity

1 Introduction

Failures represents a serious problem and users demand means to ensure not only the continuity of an application in case of failures but also to be able to avoid the loss of all computations done just prior to the fault. The design and creation of fault tolerance schemes and policies reveals itself as an important issue, especially when dealing with about large-scale and/or long-running applications, where the loss of any work can cause unbearable costs and delays. Among the many existing basic checkpointing algorithms, coordinated checkpointing and uncoordinated checkpointing with message logging are the most commonly adopted. In both cases, the checkpoint and its respective storage is a crucial element, that should be carefully analyzed, since it have a major impact in the execution of an application. Due the limited I/O (Input/output) bandwidth of both the storage facilities and the underlying hardware of both the clusters and of the storage facilities, scalability and efficiency represents a constant problem. Typical installations follow a centralized storage approach, where during the checkpoint process, all the process nodes take checkpoints at the same time (coordinated checkpointing), causing a high bandwidth stress on the I/O nodes and on the centralized storage devices. In today's applications, huge data contexts must potentially be transferred through the network into reliable storage. For that reason, the times related with checkpoints are critical, since they have a direct impact on the application execution time.

In this paper several fundamental architectures are analyzed, classified in order to to be able to cross advantages and disadvantages so to decrease the overall costs. To understand the importance of the storage and checkpointing schemes, one checkpoint per hour in petascale / exascale systems can cause a 20% overheads for current scientific applications and [1] showed that on today's systems, the I/O generated by checkpoints consume almost 80% of the total I/O usage of a system. Also in [2] they state that a 1-petaFLOPS system can potentially take a performance hit of over 50% unless the I/O bandwidth of storage nodes are increased. Currently there are a lot of checkpoint and storage schemes in existence, and many of them have a very limited number of users, since usually they are selected in terms of convenience without any regard to their impact on the running applications. A group of the most common schemes of checkpointing and storage were selected, in order to try to analyze their features and classify them. The checkpoint scheduling problem, the message logging storage and special network storage possibilities, e.g., network overlay are not mentioned and are left for future work.

The rest of the paper is organized as follows: Section 2 describes the two storage groups: disks and memory. Section 3 reviews several main checkpointing and storage methods. Section 4 ends the paper with a table summarizing the main characteristics of the methods and some final remarks.

2 Checkpoint and Storage Classes

All methods studied can be separated into two possible classes: disk-based and memory based. However, this classification could be extended to incorporate the hybrid approach: disk and memory-based, which is discussed in the next section.

2.1 Disk-Based Checkpointing

It is the most common and easiest approach available. The application periodically stores in a disk the checkpoint image. As the size of the cluster increases, so does the cost of these schemes, mainly due the memory footprint of many scientific applications, which prevents in many cases the overlap of computations and communications. This means that the cache is almost non-existent, leading to a rate of data writing as fast as the storage capability. Additionally with the increase of the number of nodes, also increases the probability of failures, causing an increase in the number of checkpoints [3]. The combined effects of image size and checkpoint frequency easily generates not only excessive contention problems but also a significant overhead in the execution time of the application. The main cause for the absurd overheads in the disks approaches comes from the fact that the hard disk drives are usually mechanical devices, so it is extremely difficult to scale its bandwidth as the rotation speed and the seek latency are limited by their physical constraints. For that reason, it is not feasible to use HDD technology to meet checkpoint interval requirement that might be as low as a few seconds.

In order to understand the problem of disk-based approaches lets consider this example: lets suppose we have a petascale architecture composed with computing nodes, I/O nodes and disk-based storage devices (see Fig. 1). The total memory of the system is around 150 TB and the storage has a 3 PB capacity and the IO bandwidth is about 20 to 300 GB/s. From these values we can calculate the minimum mean time to store 150TB: 150TB / 300 GB/s = 500 s or 8.3 minutes. If we consider a checkpoint rate of one each hour, this means that the system will spend in a good day at least: 14% of its time in checkpoint transfer operations, which is a very large overhead for time-sensitive applications. Even with special file systems (e.g.: PLFS [4]) and hardware (e.g.: Sun Fire X4500) which can significantly decrease these overheads, the costs are still too high, for long running applications.

Disk based approaches may have as the main limitation their narrow I/O bandwidth and speed. However there are several technologies that can improve their performance, namely SSDs and ioDrives. SSDs are electronic data storage instead of their electromechanical counterpart: HDDs. The SSD uses solid-state memory to persistently store data and is several order of magnitude superior to HDDs. Be that as it may, SSDs are limited by the interface speed which is the same one for the HDDs. IoDrives are a more recent technology that uses PCIe flash cards for solid state storage, providing a high read and write performance and the advantage of using a faster interfaces than HDDs and SSDs. Several experiments were made to compare these three storage devices, and the results can be seen in [5], where the performance superiority of ioDrives are demonstrated. The limited bandwidth of HDDs together with its poor access time make it not suitable for fast checkpointing On the other hand SSD technology can provide some relief, but still inadequate when compared with the memory approaches. Several experiments were made to compare these three storage devices, and the results can be seen in Table 1.

Table 1. Performance comparison of SSD, ioDrives and HDD (from [5]).

	HP server SSD	HP SFF 15K 6G SAS HDD	80 GB ioDrive
Random reads	¿20,000 IO/s	340 IO/s	119,790 IO/s
Random writes	¿5,000 IO/s	300 IO/s	89,549 (75/25 r/w mix) IO/s
Sequential reads	230 MB/s	160 MB/s	750 MB/s
Sequential writes	180 MB/s	160 MB/s	500 MB/s

2.2 Memory-based checkpointing

In these schemes, the nodes save and/or encode the checkpoints locally (in their own memory) or remotely (on others nodes' memory or in special memory devices), and since network and memory bandwidths are faster than the bandwidth of storage systems, this approach significantly reduces the overheads [6, 7]. Memory-based checkpoint, namely using DRAM technology, spite its lower latency in relation with disks, it is a constrained approach due its volatile nature and limited availability. Also these approaches has some others significant drawbacks [6]:

- most algorithms require spare or dedicated nodes;
- most encoding techniques can tolerate only 1 or 2 simultaneous failures;
- encoding algorithms capable to tolerate a higher number of simultaneous failures are highly time and memory consuming;
- if the checkpoints are store only on volatile memory, these approach is unable to restart the execution after a whole system failure;
- since it stores encoded data in main memory it has a huge memory footprint and since memory is still a very limited commodity, these techniques are impractical in large-scale systems.

Recently, non-volatile memory technologies, like the Phase-Change RAM (PCRAM) [8] are becoming more available and soon will compensate to use them instead of physical disks, providing an extraordinary improvement both in latency as in power characteristics. Research have already demonstrated the ability of these memories combined with hybrid approaches to be scalable up to the exascale.

3 Main Methods

Choosing a method or scheme to perform the checkpoint and its respective storage is not an easy task. Each method bring challenges and drawbacks which depends not only on the application but also on the underlying hardware. With so many factors to consider, it is easy to find fault tolerance mechanisms with execution overheads superior to 50% which is in many cases intolerable. Spite the active search by the industry and research groups for methods to reduce the failures rates of systems, it is still too impractical to create fail-safe components. For that reason, presently the best solution is to make checkpointing techniques more efficient. However, due the limited I/O bandwidth of both the storage facilities and the backbone, scalability represents a constant headache, whose solution is not easy. Following we present the main methods that are used today on the hope to shed some light on them.

3.1 Checkpointing Strategies

The process to proceed to the transfer of the checkpoint image is a complex problem and has already been theoretically study in several researches like [9].

The choice of the method should be done considering three aspects: concurrent writing to a disk can interfere with a process execution, controlling network contention problems slows down processes execution and the network traffic is proportional to the size of concurrently transmitted images [10]. The most effective in terms of reliability and performance is perhaps the two-level checkpoint [11] where each node or process has its own local storage, memory or disk (see Fig. 1), so that the process have the possibility to recover from transient failures without resorting to remote storage.

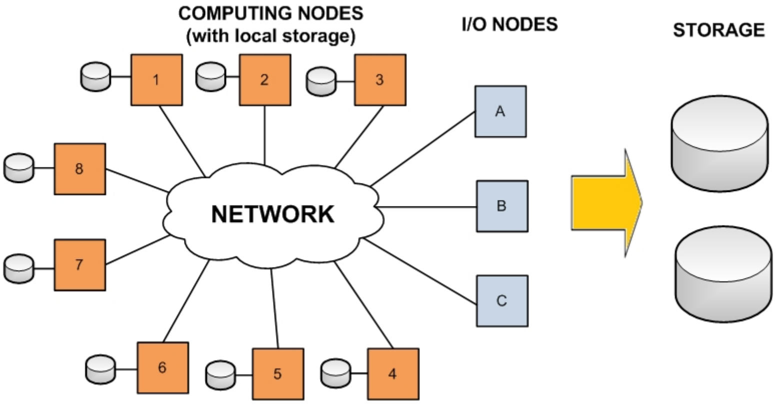


Fig. 1. Every computing node process has some sort of storage, be it memory or disk.

The existence of remote storage facilities, where a global image of the application can be stored, is used in case of node failure or catastrophic system failure. While at first sight, this approach seem expensive, by maintaining multiple checkpoints it is possible to achieve a significant overhead reduction. In Table 2 we summarize some representatives of transfer scheduling.

Incremental checkpoint [12, 13] aims to reduce the size of the checkpoint data by saving only altered memory pages (dirty pages) since the last full checkpoint. It reduces the checkpoint bandwidth and the storage space requirements, leading to a lower rate of full checkpoints. However this method requires support both from the operating system and hardware. Also, they do not scale well specially for massively parallel programs, also its effectiveness reduces with the increase in checkpoint interval in case of large checkpoint intervals, where the incremental checkpoint size can be almost the same as the full checkpoint size. Wang et al. in [12] were the first ones to experimentally derive an optimal balance between the two: 1 full checkpoint for 9 incremental. [13] present us with a mathematical approach for determining the number of consecutive incremental checkpoints, depending on the probability of failure after a checkpoint, the full checkpoint overhead and the recovery cost of each incremental checkpoint:

Table 2. Four important checkpoint image transfer scheduling.

Method	Observations
Local Disk	Protects against processes failures Does not protect against node failures Low overhead
Direct Remote storage	The greatest performance penalty Memory bounded
Local disk and remote storage	Protects against nodes failures Useful for migration schemes Better performance than direct remote storage Low overhead if transfer is made concurrently
Memory to remote storage	Memory bounded Faster than all previous

$$m = \left\lceil \frac{(1-\mu)O_F}{P_I \delta} - 1 \right\rceil \tag{1}$$

With O_F as the full checkpoint overhead, P_I the probability that a failure will occur after the second full checkpoint and before the next incremental checkpoint, μ as the incremental checkpoint overhead ratio and δ, the additional recovery cost per incremental checkpoint.

According to a performance study [14] between *blocking*and *non-blocking* checkpointing algorithms, for high-speed networks like Infiniband and Myrinet, the blocking algorithm gives the best performance for sensible or small checkpoint frequencies. However on clusters of workstations and computational grids, the costs of synchronization of the blocking protocol introduces a high overhead that does not appear with the non-blocking implementation. It was demonstrated that the checkpoint frequency impacts more on the performance than the number of nodes involved in a checkpoint synchronization for both non-blocking and blocking protocols.

Grouping or teaming processes and nodes [15, 16] according to their communication patterns and the underlying hardware can lead to a significant overhead reduction. By scheduling the group checkpoints to be taken at different times, it is easy to avoid storage bottlenecks. The coordination can be done in a way to avoid the need for message logging.

3.2 Reliability and Checkpointing Preservation

Encoding techniques are usually utilized in the checkpoint context to increase redundancy and by that the reliability of the system. There a myriad of variants, from parity-based (XOR [7], PG-LCP and ICPD [17]) to encoding-based like Reed-Solomon [18, 21]. In the XOR parity approach the checkpoints images are saved in each node's local storage. The parity of all checkpoints are calculated and saved in extra checkpointing processors so to make the application recoverable in the event of failures. The n bit of the encoded checkpoint is calculated as the exclusive-OR (XOR) operation between the n bit of the checkpoint of all computing nodes. After a failure the checkpoint of the failed node is calculated from the encoded checkpoint and the checkpoints of non-failed processes. Its main drawbacks are its inability to tolerate more than single simultaneous failures. If more occurs, the encoded checkpoint and the checkpoint of the non-failed nodes will form a system of the one equation and two or more unknowns, from which is impossible to recover any data. But separating the processes into groups, where each group can tolerate one failure, increases the reliability of this technique. However if the size of local checkpoint or stripe of data are different, the size of each checkpoint has to get virtually enlarged to the size of the biggest checkpoint, and so it is not adequate for heterogeneous executions.

To solve the single fail problem, a two dimensional parity can be used. In this case the nodes are organized into a matrix, which allows the system to tolerate several failures, but if the number of failures is superior to 2 then the problem maintains. Another more scalable algorithm, the Localized Weighted Checksum encoding, was proposed by Chen and Dongarra [19], capable of tolerating more than 3 simultaneous failures. Their algorithm relies on a chain-pipeline to encode the checkpoints. In here a binary tree is used to encode the checkpoints with the bit parity technique. Considering p processes, this method can survive the catastrophic failure of k processes, and the overhead does not increases as the p increases. In fig. 2 we can see a parity scheme variant called Parity Grouping of Local Checkpoints (PG-LCP), where the parity are locally calculated and then stored on a separated node.

Be $CP_{k,i}$ representing the bit i of the checkpoint image of process k, each parity bit is calculated by:

$$p_i = CP_{0,i} \otimes CP_{1,i} \otimes ... \otimes CP_{N-1,i} \tag{2}$$

There are others variations like the Intra-Checkpoint Distribution (ICPD), where each checkpoint is split blockwise into stripes, and the Reed Solomon, which spite time consuming, is the most tolerable encoding technique. RS is similar to the the parity technique but uses a distribution matrix, creating a system of m equations, and as long the number of failures is lower than the number of equations then its possible to recover the system.

Parity-based's main advantage is the introduction of less memory overhead, but it incurs a higher performance overhead that the neighbor scheme. Also parity schemes are usually associated with the several RAID storage layouts, destined to efficiently storage checkpoint data in a distributed reliable manner [22]. In RAID schemes the degree of redundancy can be held relatively low when compared to replication.

Copy-on-Write algorithms takes the advantage of the memory's low latency. The data is copied to a separated memory address space via virtual-memory,

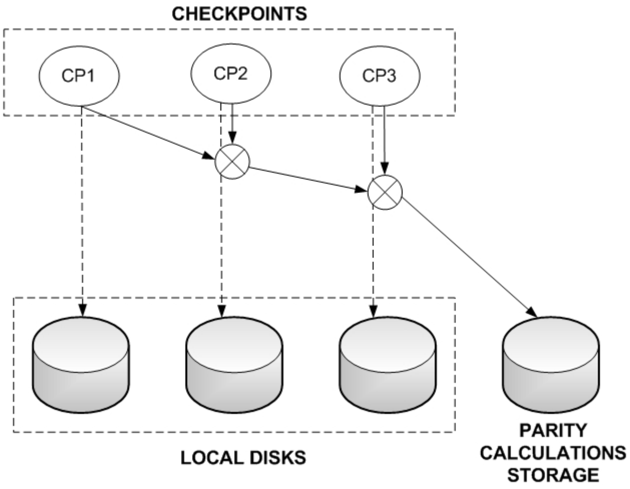


Fig. 2. Parity grouping of local checkpoints. In this case, parity is calculated stepwise along a reduction tree [20].

page protection hardware. Once complete, the data is asynchronously saved to a stable storage while the application is execution. Although Copy-to-Write can increase the checkpoint latency, it also decreases the checkpoint overhead [23]. These techniques can be greatly improved with the use of buffering capabilities [24], to enable the overlapping of memory-to-memory transfer of data and its writing to a stable storage. Since it is a memory approach it is also bounded by the available memory.

In the *Neighbor-based* approach the main memory of neighbors nodes [25]. Processes are organized in a virtual ring topology, where each process saves its checkpoint in his own memory and in the memory of the following process in the ring. The degree of replication is only one, so the system can only tolerate single failures, unless the failures are not adjacent. It is not robust against failures during checkpoint, since to tolerate these ones, each process must be able to allocate 2 checkpoint images in its memory The transmission of the image to its neighbor can be done asynchronously with the process computation.

The *Content Addressable Storage* or CAS reduces and eliminates duplication data within the CAS file system by storing common data once and referencing it as needed and any common data that exists from one checkpoint period to the next can be stored once, meaning a reduction of the disk I/O and consumed disk space. Its main drawback is the introduction of single points of failure: if one common block of data corrupts can cause many unreadable checkpoints.

At-least-k semantics [26] allows a checkpoint to be stored in the volatile memory of k nodes simultaneously, which increases the checkpoint survivability while maintaining the advantages of using only volatile memory. Since each subsequent checkpoint can be stored in different locations, this information needs to be managed during free-fault execution causing a usually insignificant overhead, but nonetheless introducing a single point of failure.

4 Conclusions and Final Remarks

This study addresses several checkpointing and storage schemes for fault tolerance in a distributed or parallel computing environment. Together with Table 2, Table 3 attempts to aggregate the major characteristics of each one of them.

From the performed analysis we can verify that the main sources of bottlenecks are the bandwidth and latency of the storage facilities. Additionally the overheads are dependent on how the checkpoint image transfer scheduler is implemented and the best performance is achieved when the image transfer is handled concurrently and asynchronously, since the storage access time can represent the largest percentage of the total checkpoint time. Each option is best suited according to a particular situation (application, hardware characteristics and usage).

As an illustration, consider a long-running application on a large system. The user should prefer the use on a non-blocking checkpoint approach and, depending on the expected failure frequency (probability) it may also chose between coordinate checkpointing with central storage together with a team-based scheme. Still, depending on the fault probability and on the storage scheme, a choice between an incremental or only full checkpointing schemes could be made. Clusters or other computing system should be provided with highly flexible fault tolerant systems, easily adaptable by the users to their respective applications.

In general terms we can say that since both a disk-based and a volatile memory approach have serious limitations, a two-level approach (local and remote storage) is advisable due its flexibility where the ratio reliability and performance can easily be tuned to the demands of the application. For clusters of loosely coupled workstations and benefiting from the availability of local storages the best checkpoint/storage scheme is a group-based non-blocking checkpoint with a hybrid storage scheme (local and remote).

In the future, we pretend to perform simulations to empirically characterize and adapt some of these schemes to be part of scalable checkpointing and storage algorithms. These will be used in distributed fault tolerance systems for applications following a message passing programming paradigm running on modern clusters, characterized by multi-core, virtualization and loosely coupled workstations.

Table 3. General characteristics of the different methods, indicating their major advantages and disadvantages.

Method	Observations
Local + Remote	◇ protects against processes failures ◇ protects against node failures ◇ low checkpoint overhead
Content addressable storage	◇ reduces the disk i/o and disk space consumed ◇ introduces single points of failure
Incremental Checkpoint	◇ reduces checkpoint bandwidth and storage space requirements ◇ leads to a lower rate of full checkpoints ◇ the overhead of inc check is smaller than for full checkpoint ◇ checkpoint time is reduced
Copy-on-Write	◇ low checkpoint overhead ◇ memory bounded ◇ complex implementation with low portability
Neighbor-based	◇ can not support multiple failures (if adjacent) ◇ higher memory overhead if compared with others approaches
At-least-k Semantic	◇ high memory overhead ◇ some overhead during fault-free executions ◇ central point of failure in replicator manager
General encoding	◇ most encoding techniques can tolerate only 1 or 2 simultaneous failures ◇ not very adequate where data distribution is heterogeneous because many techniques virtually increases checkpoint sizes so to match the largest block of data ◇ lower memory overhead ◇ higher performance overhead that Neighbor-based.
PG-LCP	◇ fast collective rollback ◇ performance comparable to RAID4
ICPD	◇ provides fast stable checkpointing ◇ parity calculations are made in parallel ◇ high number of communications ◇ fast recovery if single failure ◇ adequate for clusters of workstations
Reed Solomon	◇ most tolerable encoding technique ◇ time consuming
Localized Weighted Checksum	◇ tolerates more than 3 simultaneous failures ◇ the overhead does not increases as the p increases
Teams/Groups	◇ the checkpoint time if noticeably less than in global coordinate reaching up to 78% [15] ◇ performance improvements up to 70% ◇ reduces checkpoint delays ◇ reduces checkpoint bottlenecks ◇ computing intensive applications with large communication blocks can benefit a lot from this technique
Special hardware	◇ superior performance ◇ low portability between systems ◇ higher costs
Non-blocking	◇ the execution time remains constant ◇ as the number of central storages increases, so does the time for image transfer ◇ non-blocking is better suitable for clusters of workstations and computing grids ◇ blocking has better performance for sensible checkpoint frequency ◇ in the choice between the two the checkpoint frequency has more impact than the number of nodes

References

1. Kei Davis, Fabrizio Petrini. Achieving Usability and Efficiency in Large-Scale Parallel Computing Systems. In European Conference on Parallel Computing (Euro-Par 2004), Pisa, Italy, August 2004.
2. Ron A. Oldfield, Sarala Arunagiri, Patricia J. Teller, Seetharami Seelam, Maria Ruiz Varela, Rolf Riesen, and Philip C. Roth. 2007. Modeling the Impact of Checkpoints on Next-Generation Systems. In Proceedings of the 24th IEEE Conference on Mass Storage Systems and Technologies (MSST '07). IEEE Computer Society, Washington, DC, USA, 30-46.
3. Ron A. Oldfield, Sarala Arunagiri, Patricia J. Teller, Seetharami Seelam, Maria Ruiz Varela, Rolf Riesen, and Philip C. Roth. 2007. Modeling the Impact of Checkpoints on Next-Generation Systems. In Proceedings of the 24th IEEE Conference on Mass Storage Systems and Technologies (MSST '07). IEEE Computer Society, Washington, DC, USA, 30-46.
4. John Bent, Garth Gibson, Gary Grider, Ben McClelland, Paul Nowoczynski, James Nunez, Milo Polte, and Meghan Wingate. 2009. PLFS: a checkpoint filesystem for parallel applications. In Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis (SC '09). ACM, New York, NY, USA
5. A Comparison of SSD, ioDrives, and SAS rotational drives using TPC-H Benchmark. Technical Report White Paper, HP Development Company, 2010.
6. Leonardo Arturo Bautista Gomez, Naoya Maruyama, Franck Cappello, and Satoshi Matsuoka. 2010. Distributed Diskless Checkpoint for Large Scale Systems. In Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGRID '10). IEEE Computer Society, Washington, DC, USA, 63-72
7. James S. Plank, Kai Li, and Michael A. Puening. 1998. Diskless Checkpointing. IEEE Trans. Parallel Distrib. Syst. 9, 10 (October 1998), 972-986.
8. Moinuddin K. Qureshi, Vijayalakshmi Srinivasan, and Jude A. Rivers. 2009. Scalable high performance main memory system using phase-change memory technology. SIGARCH Comput. Archit. News 37, 3 (June 2009), 24-33.
9. Elmootazbellah N. Elnozahy and James S. Plank. 2004. Checkpointing for Peta-Scale Systems: A Look into the Future of Practical Rollback-Recovery. IEEE Trans. Dependable Secur. Comput. 1, 2 (April 2004), 97-108.
10. Hyungsoo Jung, Hyuck Han, Heon Y. Yeom, and Sooyong Kang. 2011. Athanasia: A User-Transparent and Fault-Tolerant System for Parallel Applications. IEEE Trans. Parallel Distrib. Syst. 22, 10 (October 2011), 1653-1668.
11. M. Lotfi, S. A. Motamedi, and M. Bandarabadi, Lightweight blocking coordinated checkpointing for cluster computer systems, in Proceedings of the 41st Southeastern Symposium on System Theory, 2009, pp. 144-147.
12. Chao Wang, Frank Mueller, Christian Engelmann, and Stephen L. Scott. 2010. Hybrid Checkpointing for MPI Jobs in HPC Environments. In Proceedings of the 2010 IEEE 16th International Conference on Parallel and Distributed Systems (ICPADS '10). IEEE Computer Society, Washington, DC, USA, 524-533. DOI=10.1109/ICPADS.2010.48
13. Nichamon Naksinehaboon, Yudan Liu, Chokchai (Box) Leangsuksun, Raja Nassar, Mihaela Paun, and Stephen L. Scott. 2008. Reliability-Aware Approach: An Incremental Checkpoint/Restart Model in HPC Environments. In Proceedings of the 2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID '08). IEEE Computer Society, Washington, DC, USA, 783-788.

14. Camille Coti, Thomas Herault, Pierre Lemarinier, Laurence Pilard, Ala Rezmerita, Eric Rodriguez, and Franck Cappello. 2006. Blocking vs. non-blocking coordinated checkpointing for large-scale fault tolerant MPI. In Proceedings of the 2006 ACM/IEEE conference on Supercomputing (SC '06). ACM, New York, NY, USA, , Article 127
15. Qi Gao, Wei Huang, Matthew J. Koop, and Dhabaleswar K. Panda. 2007. Group-based Coordinated Checkpointing for MPI: A Case Study on InfiniBand. In Proceedings of the 2007 International Conference on Parallel Processing (ICPP '07). IEEE Computer Society, Washington, DC, USA, 47-.
16. Ho, J., Wang, C.-L., and Lau, F. (2008). Scalable Group-based Checkpoint/Restart for Large-Scale Message-passing Systems. In 22nd IEEE International Parallel Distributed Processing Symposium.
17. Peter Sobe. 2003. Stable Checkpointing in Distributed Systems without Shared Disks. In Proceedings of the 17th International Symposium on Parallel and Distributed Processing (IPDPS '03). IEEE Computer Society, Washington, DC, USA, 214.2-.
18. T.K. Moon, Error Correction Coding: Mathematical Methods and Algorithms, Hoboken, NJ: John Wiley & Sons, Inc., 2005.
19. Zizhong Chen and Jack Dongarra. 2008. A Scalable Checkpoint Encoding Algorithm for Diskless Checkpointing. In Proceedings of the 2008 11th IEEE High Assurance Systems Engineering Symposium (HASE '08). IEEE Computer Society, Washington, DC, USA, 71-79.
20. Peter Sobe. 2003. Stable Checkpointing in Distributed Systems without Shared Disks. In Proceedings of the 17th International Symposium on Parallel and Distributed Processing (IPDPS '03). IEEE Computer Society, Washington, DC, USA, 214.2-.
21. James S. Plank, Jianqiang Luo, Catherine D. Schuman, Lihao Xu, and Zooko Wilcox-O'Hearn. 2009. A performance evaluation and examination of open-source erasure coding libraries for storage. In Proccedings of the 7th conference on File and storage technologies (FAST '09), Margo Seltzer and Ric Wheeler (Eds.). USENIX Association, Berkeley, CA, USA, 253-265.
22. J. S. Plank. 1996. Improving the performance of coordinated checkpointers on networks of workstations using RAID techniques. In Proceedings of SRDS '96. IEEE Computer Society, Washington, DC, USA, 76-.
23. Elnozahy, E. N., Johnson, D. B., & Zwaenepoel, W. (1992). The Performance of Consistent Checkpointing. 1992 Proceedings 11th Symposium on Reliable Distributed Systems, pp, 39-47. IEEE Comput. Soc. Press.
24. K. Li, J. F. Naughton, and J. S. Plank. 1994. Low-Latency, Concurrent Checkpointing for Parallel Programs. IEEE Trans. Parallel Distrib. Syst. 5, 8 (August 1994), 874-879.
25. Tzi-Cker Chiueh and Peitao Deng. 1996. Evaluation of checkpoint mechanisms for massively parallel machines. In Proceedings of the The Twenty-Sixth Annual International Symposium on Fault-Tolerant Computing (FTCS '96) (FTCS '96). IEEE Computer Society, Washington, DC, USA, 370-.
26. Justin Rough, Andrzej M. Goscinski. A Group Communications Facility for Reliable Computing on Clusters. In Edwin Hsing-Mean Sha, editor, Proceedings of the ISCA 14th International Conference on Parallel and Distributed Computing Systems, August 8-10, 2001, Richardson, Texas, USA. pages 19-24, ISCA, 2001

NS-3 EmuNetDevice evolution to support binding to PPP interfaces

Helder Fontes, Gustavo Carneiro and Manuel Ricardo

INESC TEC (Formerly INESC Porto) and Faculdade de Engenharia, Universidade do Porto, Portugal
{hfontes, gjc, mricardo}@inescporto.pt

Abstract. Network Simulator 3 (NS-3) enables the interaction of simulated network nodes with real ones through the use of special emulated network devices (EmuNetDevices) and real time execution. A virtual node can use an EmuNetDevice to bind its network device to a real network device of the computer hosting the simulation; however, the real network device must operate at MAC layer and be persistent. This paper presents an evolution of the ns-3 EmuNetDevice module enabling the support for binding simulated nodes to PPP network interfaces, which operate at IP layer and only exist when the connection is active, e.g. 3G client network interfaces. This EmuNetDevice evolution is expected to implement the new features without introducing a significant network delay. Simulation results are presented, validating the new module correct operation, and is shown that the introduced delay is negligible and can be despised, taking into account the high delays that usually characterize the 3G operator networks.

Keywords. ns-3, network simulator 3, EmuNetDevice, simulation, interface binding, network protocols.

1 Introduction

In the networking research and development field one recurring problem faced is the duplication of effort to write first simulation and then implementation code. In [1] the authors posit an alternative development process that takes advantage of the built in network emulation features of Network Simulator 3 (ns-3) [2] and allows developers to share most code between simulation and implementation of a protocol. Sharing code allows developers to 1) save time, 2) reduce the introduction of errors and 3) easily deploy the code to production environment after careful examination and safe operation in simulation mode. The results of such method are very positive [1], validating the fast prototyping of network protocols through ns-3 simulation model reuse.

This method is only possible due to the existence of a key ns-3 component called EmuNetDevice. Although, some limitations regarding operation with IP layer link types were identified throughout the use of this component in a scenario where operation above 3G operator links is needed. An example of such scenario is the one presented by WiMetronet [3] that was further addressed and improved by Helder Fontes in his MSc thesis called "Multi-Technology Router for Mobile Networks: Layer 2 Overlay Network over Private and Public Wireless Links"[4]. This work was motivated by the possibility of implementing such complex scenarios as the ones presented in [3] and [4].

The main goal of this work is to address the EmuNetDevice identified limitations, creating an improved version of this module. This new version is expected to overcome those limitations, being capable of intelligently adapt to the operation above the already supported Layer 2 and, now, the Layer 3 network links. This is intended while 1) maintaining retro compatibility with previously developed simulation scenarios and 2) without introducing any significant delay or bottleneck to the network connection.

This paper presents our work following this order: first, the current version of the EmuNetDevice is introduced (Section 2) with special focus to its features and to the identified limitations; second, a new version of the EmuNetDevice is proposed (Section 3), presenting in detail the new features developed and explaining how they overcome the related limitations; third, the test scenarios are introduced (Section 4) and the obtained results evaluated and discussed; and, finally, some conclusions about this work are drawn (Section 5) and possible future research work topics are suggested and explained.

2 Current EmuNetDevice

Fast prototyping of network protocols through ns-3 simulation model reuse [1] is only possible due to the existence of a key component from ns-3 called EmuNetDevice. This concept is now introduced.

Such as real network nodes, simulated ones can also have network interfaces. Those network interfaces usually exist only within the boundaries of the simulation scenario. The EmuNetDevice, on the other way, is a special network interface (NetDevice) that can be bound to a real network interface. This feature allows the simulated node to interact with a real network through the use of one or more real network interfaces. These real network interfaces are the ones present on the machine hosting the simulation software and running the simulation in real time. Real time simulations imply that the virtual nodes operate synchronized with a real world clock instead of running the simulation as fast as it is possible[2].

Although EmuNetDevice exists, it has a limitation: binding to real interfaces is only supported where Layer 2 (MAC layer) of the OSI model [5] is available, such as Ethernet links. It is, in fact, hardcoded to read and write Ethernet frames from an interface. Based on this, any Layer 3 (IP layer) network link, such as the PPP [6] links of 3G operator networks, cannot be used by simulated nodes as a communication medium. This happens because the lower level supported by these interfaces is layer 3, therefore, only IP packets can be read/wrote from/to these interfaces.

Another limitation of the EmuNetDevice is the lack of a feature for cloning the MAC [7] address of the real interface. In Ethernet networks not managed by the user this could present a problem. Without this feature the ns-3 will be injecting Ethernet frames with the automatically assigned "fictitious" MAC addresses. These "fictitious"

MAC addresses are only intended to be used inside the simulation, therefore, they can be filtered or cause network problems.

3 New EmuNetDevice

The EmuNetDevice module was evolved to support a new set of important features, further improving the capability of running simulated or hybrid (real and simulated nodes interacting) environments over different types of real network interface links. The development was made in C++, the language used for the source code of ns-3. Each new feature is now listed and explained. The new features are not expected to introduce significant network delay and throughput bottleneck.

3.1 MAC Address Cloning

Virtual nodes running inside ns-3 use fictitious MAC addresses by default. This does not present a problem running simulations limited to a virtual simulated environment, but could be a problem when operating in real networks. Sometimes the MAC address used by the EmuNetDevice has to match the MAC address of the host computer's real interface to allow communication in the real network. Because it is not easy nor practical to edit the MAC addresses of the simulation code to mach each real interface, an EmuNetDevice configuration option was introduced allowing to clone, automatically, and in run time, the MAC address of the real interface to which that specific EmuNetDevice is bound to.

3.2 Intelligent detection of Layer 2 or 3 interfaces

Original EmuNetDevice supports only Layer 2 network interfaces, leaving aside the possibility of using Layer 3 interfaces such as the PPPs from 3G operator networks. Layer 3 interfaces do not operate with Ethernet frames. The lower level units they can read and write are the IP packets. The ns-3 was hardcoded to write the Ethernet frames' MAC header. Now it auto detects if the interface only supports IP level, allowing to communicate with Layer 3 interfaces. This intelligent adaptation allows the EmuNetDevice to remain retro compatible with previous coded scenarios.

4 Tests and Results

This Section describes each executed test and its objectives. In the end, the results are also presented and interpreted.

Before defining the tests, it is important to decide what is important to measure. It was decided to measure 1) if the EmuNetDevice can establish communication, 2) the connection delay and 3) the throughput achieved. Through this measurements it is possible to acknowledge if the new module is operating correctly and without introducing excessive delay nor throughput bottlenecks.

Those measurements were taken in two different scenarios. Scenario 1 represents a situation where we need to use 3G operator links, previously unsupported by the simulator. The correct communication between a real node and a virtual node will validate the correct detection of PPP links, with the simulator being now capable of reading and writing IP packets from/to an interface. Scenario 2 uses LAN Ethernet links to connect a real and a virtual node. This last scenario was created to validate the correct operation of the simulator in a previous supported scenario, and also to validate the MAC cloning feature.

4.1 Scenario 1 - Using 3G operator links

Due to the instable delay and throughput variation that characterizes the nature of a 3G connection, the correct operation of the simulator using PPP links was the only thing possible to check. A virtual ns-3 node was used to communicate with a real node. ICMP[8] echo requests and replies were exchanged between nodes and were checked with Wireshark[9], a packet sniffing program. UDP echo operations were also performed, running an echo server inside ns-3 and inside the real node. Each one of the tests was performed with success, showing good operation of the new evolved version of the EmuNetDevice.

4.2 Scenario 2 – Using LAN links

Using LAN links the same tests used in Scenario 1 were performed, validating the good operation of the new version of the EmuNetDevice. Some tests measuring the delay introduced by the simulation software were also performed. The bottleneck represented by the network operations inside the simulator was also measured. The delay introduced by the simulator was approximately 1ms, which is considered insignificant for most types of possible network traffic. Related to the throughput, for big packets, with an MTU of 1400Bytes, the simulator was able to perform network operations at a rate of 94Mbps, corresponding to the real throughput of a 100Mbps Ethernet link. For small packets, with an MTU of 160Bytes, the throughput was limited to approximately 10Mbps, which is 6x slower than the expected 60Mbps throughput in kernel space. This is explained due to the increased CPU load caused by processing each packet at user space. Although 10Mbps is considered a very good result, since small packet traffic is mostly related to VOIP traffic and such high bandwidth per node is not expected to be achieved.

In this scenario was used the same code as in scenario 1. Running the same code helped validating the introduction of low delay and the low impact in the network throughput. This would not be possible to test in scenario 1 due to the instable delay and throughput variation of the 3G connection used.

5 Conclusions and Future Work

Finished the development of the new version of the EmuNetDevice and after all the tests executes, it is possible to conclude that the new version of the EmuNetDevice can be easily used and all the *a priori* objectives were achieved. Now, a virtual node inside the ns-3 can intelligently operate above real 3G operator network links and clone the MAC address of the real Ethernet interfaces. The introduced delay was demonstrated to be despised and the throughput bottleneck only exists for intensive traffic of small packets, which is improbable to occur.

As future work, other evolutions to the EmuNetDevice are expected to be introduced. 3G operator links are known for its instability, leading to losses of connectivity and the extinction of the temporary PPP interface. This leads to the end of the simulation because ns-3 is not expecting a network interface to disappear. A good evolution will be to enable the EmuNetDevice to resist to connection drops, automatic reestablish the connectivity and updating the IP address of the simulated node.

References

[1] Gustavo Carneiro, Helder Fontes, e Manuel Ricardo, «Fast prototyping of network protocols through ns-3 simulation model reuse», *Simulation Modelling Practice and Theory*, 10-Jul-2011.

[2] «ns-3». [Online]. Available: http://www.nsnam.org/. [Accessed: 15-Set-2011].

[3] M. Ricardo, G. Carneiro, P. Fortuna, F. Abrantes, e J. Dias, «WiMetroNet—A Scalable Wireless Network for Metropolitan Transports», *2010 Sixth Advanced International Conference on Telecommunications*, 2010.

[4] Helder Fontes, «Multi-Technology Router for Mobile Networks: Layer 2 Overlay Network over Private and Public Wireless Links», MSc, Faculdade de Engenharia da universidade do Porto, 2010.

[5] H. Zimmermann, «OSI reference model--The ISO model of architecture for open systems interconnection», *Communications, IEEE Transactions on*, vol. 28, n. 4, pp. 425-432, 1980.

[6] W. Simpson, «The Point-to-Point Protocol (PPP)». [Online]. Available: http://tools.ietf.org/html/rfc1661. [Accessed: 19-Jan-2012].

[7] «IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications». 2007.

[8] J. Postel, «Internet Control Message Protocol». [Online]. Available: http://tools.ietf.org/html/rfc792. [Accessed: 19-Jan-2012].

[9] «Wireshark · Go deep.» [Online]. Available: http://www.wireshark.org/. [Accessed: 19-Jan-2012].

Session IV

Information Retrieval

Chairman: Sameh Eisa

Connecting DSpace and Protégé : Applying ontologies to research data curation
João Rocha Da Silva

Twitter event detection: combining wavelet analysis and topic inference summarization
Mário Cordeiro

Classification of Sentiment Polarity of Portuguese On-line News
Inês Coimbra Morgado

A Comparative Study of Hierarchical Clustering Algorithms for Tagging Systems
Anisa Allahdadi

Connecting DSpace and Protégé : Applying ontologies to research data curation

João Rocha da Silva

Faculdade de Engenharia da Universidade do Porto FEUP-DEI
Rua Dr. Roberto Frias, s/n, 4200-465 Porto PORTUGAL
joaorosilva@gmail.com,
WWW home page: http://joaorosilva.no-ip.org

Abstract. Research data management is presently assuming great importance as it becomes increasingly essential to the research workflow. Regardless of the policies that institutions decide to adopt, they must be supported by adequate tools such as online research data repositories. One of such tools is DSpace, a web-based repository platform designed for the preservation of research publications that has been recently customized at U.Porto to support a research data management workflow. In this paper, a further improvement to this workflow is proposed, which has the goal of improving the metadata richness of the datasets by representing them in an ontology format and storing these representations at the repository. To achieve this, a plugin for the well-known Protégé ontology editor (*Mapping Master*), was customized to work together with a DSpace repository to assist curators in the dataset translation and deposit process. The end result is an integrated solution for converting tabular datasets into ontologies that can be easily shared or automatically processed and contain richer representation information than their annotated table counterparts.

1 Introduction

According to the Data Management Association (DAMA), data management is the development, execution and supervision of plans, policies and practices that control, predict, deliver and enhance the value of data and information assets [1]. Research data management, in particular, is a problem with specific issues that range from the political and ethical issues concerning data sharing to the technical aspects of how to make data reuse possible. The debate on Open Data is complex and closely related to this work, but our focus is to make it technically possible, not determining if it should be general practice.

Research data is among the most valuable types of data, since research datasets may be useful for entire research communities and also because research findings can have an impact on society as a whole [2]. The emerging "Fourth Paradigm of Science", which comes from the increasingly widespread access to powerful computing capabilities in networked environments [3] is now allowing researchers to manipulate and produce larger and larger datasets [4]. As a consequence, research data management is one of the greatest challenges of the information age—this “data deluge” [5] needs to be addressed to ensure the reproducibility of research findings, placing new research questions and increasing the visibility of researchers and institutions [6].

An essential part of data curation is capturing a dataset’s *representation information*, which is the term used to designate all the relevant information that is required for a researcher to correctly interpret and reuse a dataset [7]. Capturing the semantics of research data is especially difficult because of its inherent complexity and specifics—representation information may use domain-specific terms which can only be correctly understood by researchers working on that dataset’s particular research domain.

Representation information is usually represented by metadata (“data about data”) that can be formalized in the form of controlled vocabularies which are “established lists of standardized terminology for use in indexing and retrieval of information. An example of a controlled vocabulary is subject headings used to describe library resources” [8]. Metadata can also be generic—author, creation date and location, associated publications—or specific—in the case of a chemistry dataset, a metadata element may refer to the chemical agents used for the experiment that yielded the data. Dublin Core is a widely used generic metadata profile for describing resources [9].

A specific type of controlled vocabulary are ontologies, which are in short, “specifications of conceptualizations” [10]. Ontologies are closely related to the semantic web [11] and are excellent tools for defining relevant concepts of a domain (the types of things and relationships between those things). They also providing a higher-level representation of datasets because they do not rely on the structure or syntax features of documents such as nesting or numbering of elements [12]. They not only provide richer semantic representations but can also be used to infer knowledge over existing sets of relationships between different entities, something that can play an important role in the retrieval of research information (“semantic search”). Ontologies evolve through their reuse, making it it one of the goals to aim for when designing one [13]. The common understandings that arise from that reuse are the basis for interoperability [14].

This paper will begin with an introduction of the research data management domain. Some related work will be presented, followed by a description of the implemented data curation solution, which will include a technological analysis and architectural description. A step-by-step demonstration of the software will follow. Finally, some conclusions will be drawn and some future work prospects will be discussed.

2 Data curation in DSpace

In order to establish the context of this work, a brief description of a prior data curation will now be described. This experiment, called UPData, took place throughout 2011 at FEUP with the cooperation of the Rectorate of U.Porto. During this experiment, several research groups and individual researchers from diverse domains were interviewed to assess their practices and to elicit their data management needs. When inquired about data reuse, all the interviewed researchers stated that they were not comfortable using data that they knew little about (its collection method, date, instruments used, etc.). Thus, the decision to reuse data depends on the richness of the metadata models used to capture its representation information. Another aspect to point out is that researchers tend to judge the adequacy and usefulness of a repository primarily by the data access features that it offers.

As an approach to this problem, the experiment included the customization of a DSpace instance to better support the handling of research data. The changes included storing XML representations of regular data at the repository to enable easier querying and exploration of the data. The chosen XML format for storing this data contains limited semantics—it uses elements from existing XML Schemas created by specialists from each research domain to establish common meanings to the the dimensions of each table, a first step in establishing direct connections between datasets from different domains.

The UPData project was an important first step in implementing a research data management workflow, since it made it possible to store the data in a consistent and queriable format, a prerequisite for building higher-level, richer representations of that data.

3 Building ontologies from tabular data

Research data often carries domain-specific, complex semantics, requiring high-level representations such as ontologies to comprehensively represent its representation information. However, it can be difficult to devise a method to translate real-world data representations to ontology-based representations, so this translation step must be supported on tools designed to assist the curators in this process. These tools must be flexible enough to allow for the translation of arbitrary subsets of tabular data into ontologies without relying on any aspect of the tables' structure, while reusing concepts from domain-specific ontologies. The translated datasets should also be made available on a web-based interface to make them simple to retrieve.

The goal of this work is to build such a solution by combining two open-source software solutions: a web-based data repository and an ontology editor. The repository will be used to preserve the datasets' original files, the ontologies that contain the concepts of each domain and the translated versions of datasets (in ontology format). The ontology editor will be connected to the repository to retrieve the data and any reusable domain ontologies during the translation process. After the dataset is translated into an ontology, the editor must be capable of sending the translated version to the central repository as well as another ontology that establishes the concepts used in the mapping process.

3.1 Selecting a repository platform

There are several open-source repository platforms currently available, most of them designed for the preservation of research publications. Some prominent examples are DSpace, ePrints and Fedora. DSpace was originally selected for UPData because it has a large user base and also because it is currently used by U.Porto to support its Open and Thematic Repositories [15].

The improvements made to DSpace focused primarily on tabular data curation, using XML as the internal format to reduce technology dependencies to a bare minimum. Excel spreadsheets were selected as the means to exchange information with the end-users because most are familiar with this software and use it on a regular basis.

In DSpace, an `Item` is the smallest-granularity object to which Dublin Core metadata can be added. An `Item` is usually a set of files (called `Bitstreams`) that constitute a publication or a dataset. The modifications to DSpace introduced a finer-granularity entity—the `Datatable` level. A `Datatable` represents each of the tables contained in a file and has its own parametrizable metadata, which can also include Dublin Core elements. `Items` are grouped into `Collections` which are used to group them according to their domain, origin or other relevant aspects—for example, an institutional repository may host several collections, one for each of its working research groups.

The result was a platform that offers data deposit features designed for tabular data as well as simple data sub-setting and dataset retrieval functionality. These additions were built on XML technology—XML documents are used for data storage, XQuery is used for querying the deposited data and XSLT style sheets are used to handle the visualization of the data on the client's Web Browser.

3.2 Selecting an ontology editor

An ontology editor is a tool designed to assist users in building an ontology. It must support operations such as the specification of classes or properties and support the addition of class instances/individuals to the ontology.

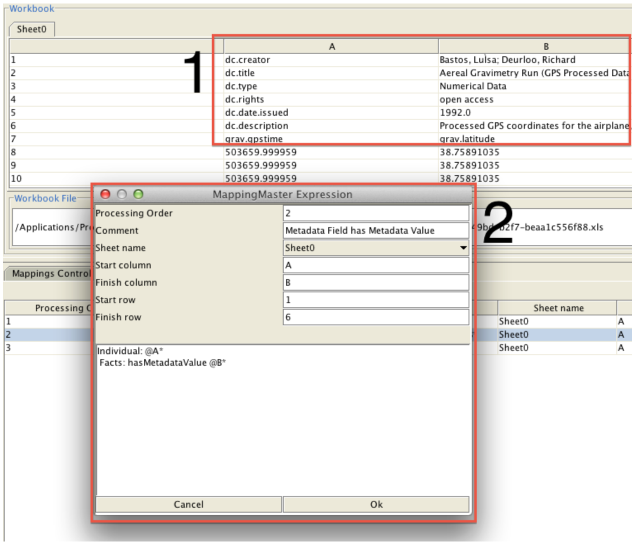
The *Protégé* ontology editor [16] is a cross-platform open-source software that supports all these features. Moreover, it is extensible through its plug-in architecture, allowing third parties to add features to the base platform. For the purpose of this work, a particularly interesting plug-in is the *Mapping Master* extension, which can be used to transform the contents of Excel spread sheets into OWL[1] ontologies [17]. It allows users to map arbitrary sections of Excel workbooks into ontology concepts (regardless of the workbook's structure) and can also establish relationships between the results of the mapping operation. The mappings are specified by the user through series of statements in the Mapping Master (or M^2) language [18].

[1] Web Ontology Language

3.3 The M^2 language and the Mapping Master plugin

The Mapping Master plug-in for Protégé assists users in building ontology representations of their Excel-formatted data by mapping sections an Excel workbook to ontology concepts. Using the interface, the user can specify a series of mapping statements in a DSL[2] designed by the creators of the Mapping Master extension—this language is called M^2 and is based on the Manchester Syntax [19].

The M^2 DSL was designed to allow the mapping or arbitrary sections of a workbook to ontology concepts, thus eliminating the need for these workbooks to adhere to a specific convention regarding its structure. These sections can also be specified using *wildcards*, making it easier for users to select large parts of a workbook when specifying a mapping. An example of a mapping statement using wildcards is shown in Figure 1.



	A	B
1	dc.creator	Bastos, Luìsa; Deurloo, Richard
2	dc.title	Aereal Gravimetry Run (GPS Processed Data
3	dc.type	Numerical Data
4	dc.rights	open access
5	dc.date.issued	1992.0
6	dc.description	Processed GPS coordinates for the airplane
7	grav.gpstime	grav.latitude
8	503659.999959	38.75891035
9	503659.999959	38.75891035
10	503659.999959	38.75891035

MappingMaster Expression

Processing Order	2
Comment	Metadata Field has Metadata Value
Sheet name	Sheet0
Start column	A
Finish column	B
Start row	1
Finish row	6

```
Individual: @A*
 Facts: hasMetadataValue @B*
```

	Sheet name	
1	Sheet0	A
2	Sheet0	A
3	Sheet0	A

Fig. 1. Editing a mapping expression in Protégé Mapping Master

This interface allows users to specify the M^2 statements of each mapping—“`Individual: @A* (MetadataField) Facts: hasMetadataValue @B*`” is the example shown in the figure—as well as other parameters, namely the Start/End rows and Start/End columns. In this particular case, the statement makes use of an ontology class, `MetadataField` and a data property, `HasMetadataValue`.

When this mapping shown in Figure 1 is run against this workbook, six `MetadataField` instances will be added to the ontology currently being edited

[2] Domain Specific Language

in the Protégé editor due to the use of wildcards—let $i \in 1..6$ (Start Row...End Row). For each A_i cell, a `MetadataField` instance X_i will be created, being A_i the value of the cell in column A, row i. Also, $\texttt{hasMetadataValue}(X_i) = B_i$, being `hasMetadataValue` a data property of the currently open ontology and B_i the element contained in the cell in column B and row i.

3.4 The translation step as part of the curation workflow

A high-level representation of the new data curation process is shown in Figure 2. In this figure, section 1 shows the process that a dataset must go through before it can be translated into an ontology—it must be deposited in the DSpace repository and undergo the tabular curation process implemented in UPData.

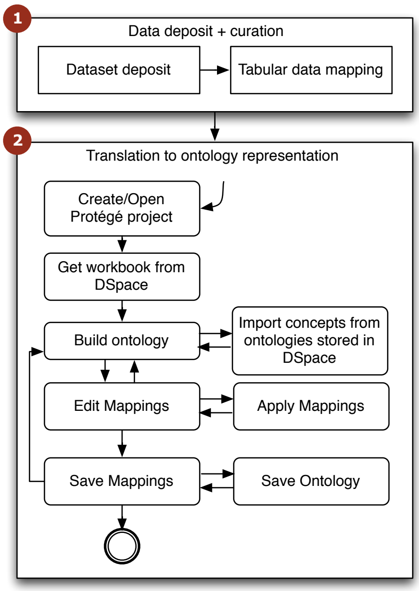


Fig. 2. The curation workflow

Section 2 shows the steps in the translation of a dataset's tabular data into an ontology-based representation. First, the user must create a blank project in the Protégé ontology editor and select the workbook that is to be retrieved from DSpace for annotation. The ontology building process follows—the user must specify the concepts that are contained in the workbook. This ontology can then be built "manually"—by using standard editing interface—and also by importing existing ontologies stored in the DSpace repository. Figure 2 shows the workbook selection step before the ontology building step, but these steps can be carried out in reverse order.

The ontology importing feature is interesting to allow ontologies stored in the DSpace repository to be imported directly into the Protégé knowledge base, thus encouraging ontology reuse.

The next step in the process is to specify the mappings to be applied to the workbook, using the knowledge base currently loaded in the editor. When the user executes the mapping operation, these are validated by the Mapping Master system and the current ontology is updated to reflect the results of the process. The user has the option of saving the list of mappings to the DSpace repository at any time, but since these mappings are established between sections of the current workbook and concepts of the current Protégé knowledge base, it must be submitted to DSpace along with the mappings on every save operation. Only when the mappings are consistent with the knowledge base may both be saved to the repository—such validations are performed on the Protégé client.

4 Integrating DSpace and Protégé for data curation

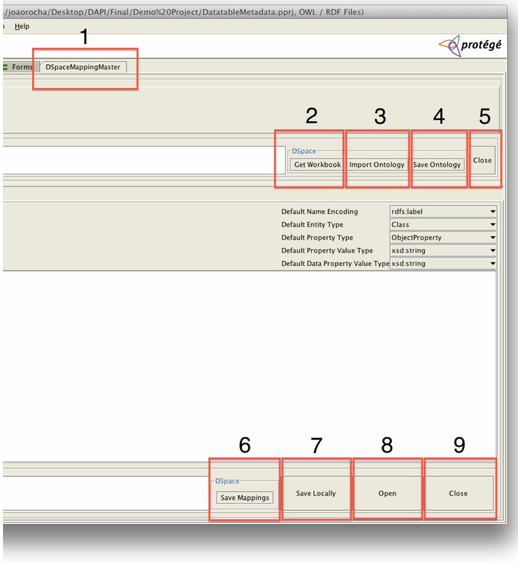
In this section we will show a full mapping workflow using the plugin's functionality. In this simple example, the metadata fields of a workbook will be translated into ontology format, but more sophisticated mappings could be performed due to the flexibility of the M^2 mapping language.

4.1 Main window

Figure 3 shows the main window of the plug-in, which we called "DSpaceMappingMaster". The highlighted interface elements in the figure are described as follows:

1. The plugin is shown as a tab that can be activated in the project configuration settings area of the interface
2. Retrieving workbook from DSpace—used for importing a workbook from the remote repository and showing it in the Protégé interface, as well as importing any ontologies previously associated to it.
3. Importing an ontology—this option is used retrieve a domain concepts (not dataset) ontology from DSpace and import it into the active knowledge base
4. Saving the current knowledge base to DSpace and associate it to the workbook
5. Closing the currently open workbook
6. Saving mappings ontology to DSpace
7. Saving the mappings ontology to the local file system
8. Loading a mappings ontology from the local file system
9. Closing the current mappings list

Saving the currently open knowledge base to the local file system is performed through the "Save" option in Protégé, since it does not depend on the plugin.


/joaorocha/Desktop/DAPI/Final/Demo%20Project/DatatableMetadata.pprj, OWL / RDF Files)

Default Name Encoding: rdfs:label
Default Entity Type: Class
Default Property Type: ObjectProperty
Default Property Value Type: xsd:string
Default Data Property Value Type: xsd:string

Fig. 3. The DSpace Mapping Master main window

4.2 Validating the repository location

When Protégé first tries to interact with DSpace, the user must specify the location of the remote repository. To ensure that the location is correct and that the remote repository supports the required operations, the location must be validated. Thus, a service for validating the repository address was implemented to allow the client to send a request to the specified address and—if the repository is valid—receive a reply with an XML message. This message includes a version number, which is a simple way to tell clients which functionalities they can expect from a server when communicating with it for the first time. Currently it is not used, but was included for added future-proofing.

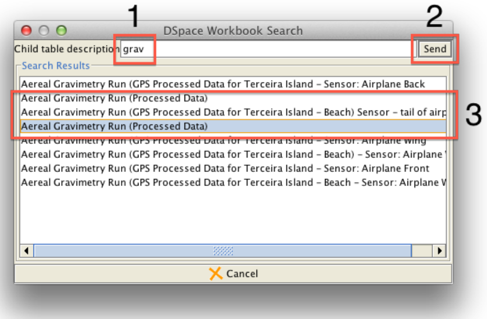
After the repository is successfully validated, the client will save it to the local file system to avoid requesting this information every time the program is opened.

4.3 Importing a workbook

When the user selects this option, a prompt similar to Figure 4 will be shown.

To search for a workbook, the user must first specify part of the title of a `Datatable` in that workbook. This is done by typing the query in text area 1 and sending it by selecting button 2.

The querying is performed in DSpace against `Datatable` titles because the platform does not include Dublin Core metadata at the `Bitstream` (file) level.



Child table description grav

Aereal Gravimetry Run (GPS Processed Data for Terceira Island – Sensor: Airplane Back
Aereal Gravimetry Run (Processed Data)
Aereal Gravimetry Run (GPS Processed Data for Terceira Island – Beach) Sensor – tail of airp
Aereal Gravimetry Run (Processed Data)
Aereal Gravimetry Run (GPS Processed Data for Terceira Island – Sensor: Airplane Wing
Aereal Gravimetry Run (GPS Processed Data for Terceira Island – Beach) – Sensor: Airplane
Aereal Gravimetry Run (GPS Processed Data for Terceira Island – Sensor: Airplane Front
Aereal Gravimetry Run (GPS Processed Data for Terceira Island – Beach – Sensor: Airplane W

Fig. 4. Workbook search dialog

As a consequence, `Bitstreams` have no title, so we must use their associated `Datatables`'s metadata to find datatables and return the parent `Bitstreams` as the query results.

After the user selects one of the results shown in list number 3, the corresponding workbook will be loaded in the Protégé interface. The system will also try to retrieve any mappings ontologies already associated to the workbook, as well as the concept ontologies that they depend on—when a mapping ontology is imported, the matching concepts ontology has to to be imported into the knowledge base, or else it would not be possible to execute the mapping operation. The results of the import are shown to the user in a dialogue message. In this case, only the workbook was imported because it had no associated ontologies.

4.4 Editing ontology

The user can edit the knowledge base using the standard editor features. In this case, we have added a class named `MetadataField`, which represents all metadata fields of a `Datatable`. After the classes and properties have been added to the knowledge based they can be used in mapping statements.

4.5 Importing an ontology

Importing an ontology is performed via the button number 3 shown in Figure 3. After selecting this button, the system will show a search window similar to the one used in the workbook importing feature. The user can specify part of the description of the ontology to be imported in the top text box. After the query is submitted to the DSpace repository, the matches will be displayed a list. After the user selects the ontology to be imported from that list, the ontology will be downloaded from DSpace and imported into the current Protégé knowledge base. The results of the import are shown in Figure 5.

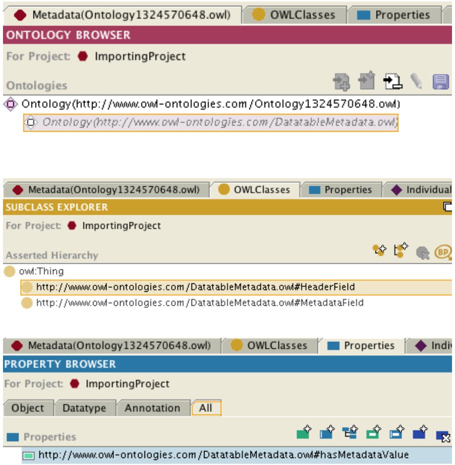

Metadata(Ontology1324570648.owl)
OWLClasses
Properties
ONTOLOGY BROWSER
For Project: ImportingProject
Ontologies
Ontology(http://www.owl-ontologies.com/Ontology1324570648.owl)
Ontology(http://www.owl-ontologies.com/DatatableMetadata.owl)
Metadata(Ontology1324570648.owl)
OWLClasses
Properties
Individual
SUBCLASS EXPLORER
For Project: ImportingProject
Asserted Hierarchy
owl:Thing
http://www.owl-ontologies.com/DatatableMetadata.owl#HeaderField
http://www.owl-ontologies.com/DatatableMetadata.owl#MetadataField
Metadata(Ontology1324570648.owl)
OWLClasses
Properties
Indi
PROPERTY BROWSER
For Project: ImportingProject
Object
Datatype
Annotation
All
Properties
http://www.owl-ontologies.com/DatatableMetadata.owl#hasMetadataValue

Fig. 5. Ontology import results

4.6 Saving mappings to DSpace

Mappings are saved to DSpace in an OWL representation which adheres to the ontology used by the Mapping Master, and only consistent knowledge base/mappings pairs may be saved on the repository—the validation is performed before sending the save request.

After selecting the “Save Mappings” option (area 6 in Figure 3), the user will be prompted to confirm the submission to DSpace. After the description is entered, the mappings and project ontologies are saved to DSpace in OWL format and associated to the workbook—if the operation is successful, a message will be presented to the user.

Saving new mappings will overwrite any existing ones as well as the concepts ontology for this dataset. If the user confirms the submission, another prompt will be shown so that the user can enter the description of the new ontology.

5 Conclusions and Discussion

Research data preservation is a complex issue that poses many technical challenges, particularly when devising flexible and rich data models to represent the data’s *representation information*. In spite of these concerns, there are great benefits in the adoption of these practices, especially in nowadays’ research environment, in which information and powerful computing capabilities assume prominent roles. Some of these benefits are: improved data reuse, research reproducibility, improved citation and also the discovery of interesting new research topics through meta-research or “data mashups”.

We have demonstrated how two open-source software solutions (an ontology editor and a publication repository) can be integrated to translate tabular datasets into ontology representations. Ontology representations are much more preservation-friendly than the original Excel spreadsheets in which the data were originally stored because they carry much richer semantic content. At the same time, they can be represented in established syntaxes, do not rely on specific technologies and can be used to provide better information retrieval mechanisms for research data through the inference process.

Such repositories can help encourage the reuse of the original research data and also of the metadata standards used to catalogue the data. It remains to be determined if such repository functionalities can effectively contribute to mitigating the issues that arise from the existence of many—often conflicting—metadata “standards”.

5.1 Future work

We now list some future development and research opportunities that may stem from this work.

One of the planned features of the system is to integrate with the authentication system of DSpace, since at the present time the communication with the repository is performed without validating the identity of the client. This is still under active development and it is expected that the system will filter the results according to the current user’s permissions, returning only datasets that belong to DSpace `Collections` that the user can curate. Another feature that could be implemented in the future could be to use metadata contained in ontologies themselves as a way to retrieve them (for example through `rdf:Description` elements contained in the XML), instead of using a description that is currently stored as a field in the DSpace database.

Protégé includes a built-in feature that allows the standard application to connect to remote ontology repositories. It may be interesting to study a possible integration between DSpace and this system for sharing concept ontologies instead of relying on a “tailor-made” web service. Improvements to the client application could also include a dedicated option for selecting a different repository at any time, since that at the present status of development the application only requests a new repository location when the repository on the current address does not respond to a request.

5.2 Acknowledgements

The author would like to acknowledge that this work has been performed under research grant SFRH/BD/77092/2011, provided by the FCT (Fundação para a Ciência e Tecnologia).

References

1. Mark (Editor) Mosley. Data management body of knowledge - introduction and project status. URL: `http://www.dama.org/files/public/DI_DAMA_DMBOK_Guide_Presentation_2007.pdf`, November 2007.
2. Francine Berman. Got data?: a guide to data preservation in the information age. *Commun. ACM*, 51(12):50–56, 2008.
3. Anthony Hey, Stewart Tansley, and Kristin Tolle. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research, 2009.
4. A J G Hey and A E Trefethen. The data deluge: An e-science perspective. *Grid Computing - Making the Global Infrastructure a Reality*, pages 809–824, 2003.
5. Philip Lord, Alison Macdonald, Liz Lyon, and David Giaretta. From data deluge to data curation. In *In Proc 3th UK e-Science All Hands Meeting*, pages 371–375, 2004.
6. Christine L. Borgman. The conundrum of sharing research data. URL: `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1869155#`, 2011.
7. Digital Curation Centre. What is representation information? URL: `http://www.dcc.ac.uk/resources/curation-lifecycle-model/lifecycle-model-faqs#node-1082-question-5`, December 2011.
8. OECD. Controlled vocabulary. URL: `http://stats.oecd.org/glossary/detail.asp?ID=6260`, November 2005.
9. Dublin Core Metadata Initiative. Dublin core metadata element set, version 1.0: Reference description. URL: `http://dublincore.org/documents/1998/09/dces/`, October 2011.
10. Tom Gruber. Ontology. URL: `http://tomgruber.org/writing/ontology-definition-2007.htm`, December 2009.
11. Ian Horrocks. Ontologies and the semantic web. *Commun. ACM*, 51:58–67, December 2008.
12. Elin K. Jacob. Ontologies and the semantic web. *Bulletin of the American Society for Information Science and Technology*, 29(4):19–22, 2003.
13. Elena Paslaru Bontas, Malgorzata Mochol, and Robert Tolksdorf. Case studies on ontology reuse. In *In Proceedings of the 5th International Conference on Knowledge Management IKNOW05*, 2005.
14. Muthukkaruppan Annamalai and Leon Sterling. Guidelines for constructing reusable domain ontologies, 2003.
15. University of Porto. Open and thematic repositories. URL: `http://repositorio.up.pt/repos.html`, November 2011.
16. Stanford Center for Biomedical Informatics Research. The protégé ontology editor and knowledge acquisition system. URL: `http://protege.stanford.edu/`, October 2011.
17. Protégé Wiki. Protégé wiki: Mapping master. URL: `http://protege.cim3.net/cgi-bin/wiki.pl?MappingMaster`, October 2011.
18. Martin J. Oconnor and Christian Halaschek-Wiener. Mapping master: a flexible approach for mapping spreadsheets to owl. In *9th International Semantic Web Conference (ISWC2010)*, November 2010.
19. W3C. Owl 2 web ontology language - manchester syntax. URL: `http://www.w3.org/TR/owl2-manchester-syntax/`, October 2011.

Twitter event detection: combining wavelet analysis and topic inference summarization

Mário Cordeiro

Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto PORTUGAL
pro11001@fe.up.pt

Abstract. Today streaming text mining plays an important role within real-time social media mining. Given the amount and cadence of the data generated by those platforms, classical text mining techniques are not suitable to deal with such new mining challenges. Event detection is no exception, available algorithms rely on text mining techniques applied to pre-known datasets processed with no restrictions about computational complexity and required execution time per document analysis.
This work presents a lightweight event detection using wavelet signal analysis of hashtag occurrences in the twitter public stream. It also proposes a strategy to describe detected events using a Latent Dirichlet Allocation topic inference model based on Gibbs Sampling. Peak detection using Continuous Wavelet Transformation achieved good results in the identification of abrupt increases on the mentions of specific hashtags. The combination of this method with the extraction of topics from tweets with hashtag mentions proved to be a viable option to summarize detected twitter events in streaming environments.

Keywords: event detection, topic detection, text stream mining, twitter

1 Introduction

Twitter differs from other social networks by being a micro-blogging service that limits the size of messages. This feature that allow twitter users to publish short messages, in a faster and summarized way, make it the preferred tool for the quick dissemination of information over the web. In March 2011, the estimated number of twitter users was 200 million [22] and the amount of messages published in a single day totalized 177 million tweets sent on the March 11, 2011 [23]. People use twitter to share advice, opinions, news, moods, concerns, facts, rumors, and everything else imaginable. Corporations use twitter to make announcements of products, services, events, and news media companies use twitter to publish near real-time information about breaking news.

From the point of view of data mining, tweets can be seen as a source of data enabling users and corporations to stay informed of what is happening now or what's being said about them and their brands. Sentiment Analysis and Opinion Mining performed subsets of tweets mentioning the person or product keywords [14] are common text mining problems applied to twitter corpus.

Being Twitter the “what’s-happening-right-now” tool [21] and given the nature of it’s data – an real-time flow of text messages (tweets) coming from very different sources covering varied kinds of subjects in distinct languages and locations – makes the twitter public stream an interesting data set for event detection based on text mining techniques.

In fact the use and extension of text retrieval and clustering techniques for event detection has long been a research topic [33]. Examples of specific applications to the twitter stream are the first story detection proposed by Petrovic et al. [16] that tries to detect whether users discuss any new event that have never appeared before in Twitter, Weng et al. [27] proposed the detection of generic events using signal analysis on the Singapore General Election 2011, and Sakaki et al. [20] exploit tweets to detect critical events like earthquake .

In twitter event detection, the underlying assumption that some related words would show an increase in the usage when an event is happening is not a viable method [12] . In comparison to traditional event detection from news wire, the twitter stream include a much higher volume of data flooded by high amounts of meaningless messages. According to a study by Pear Analytics [19], about 40% of all the tweets are pointless “babbles” and 37% conversational. Such tweets, that some authors call noise [16], are important to build a user’s social presence [11] and may help to understand the impact an event had or how people reacted to it, but normally they affect negatively the performance of event detection algorithms.

2 Related work

Early first story detection systems where based on the representation of documents as vectors in a term space using term frequencies [1, 32]. Applying a distance measure, new documents are compared to their nearest neighbors and if it’s distance exceeds an predefined maximum value the document is considered to be a first story. This method implies to have all the document term frequencies in memory, moreover, finding the nearest neighbor for new documents even for optimized solutions don’t provide much improvement over a simple linear search [10]. To overcome this, Petrovic et al. [16] proposed an modified locality sensitive hashing (LSH) [7] used as a nearest neighbor search optimization that fulfills the data stream mining requirements by using constant size buckets.

Authors Chen and Roy [5] and Weng et al. [27] assume that the occurrence of an event may be detected by observing abrupt increases on the use keywords related with the event. Weng et al. [27] proposed an wavelet-based twitter event detection. Initially, based on the number of word occurrences over the time, individual signals for each of the words are constructed. Signals are then filtered per wavelet analysis to reveal bursts in the word’s appearance and therefore compute the cross-correlation between signals. Finally events are detected by applying a modularity-based graph partitioning clustering algorithm to the signals. Using wavelet analysis it is compatible with the stream mining requirements, but the cross-correlation between signals or the modularity-based graph partitioning may not meet those requirements in particular the use of limited resources and the capability of working in real-time.

Using topic distributions rather than bags of words to represent documents reduces the lexical variability and retain the overall semantic structure of the corpus [34]. Topic models discover the abstract "topics" that occur in a collection of documents and are able to identify low sets of representative characteristics of the documents in very high dimensional data. Inference for topic models remain computationally expensive even with the recently advances in fast inference latent Dirichlet allocation (LDA) algorithms Blei et al. [3]. SparseLDA [34], FastLDA [17] and O-LDA [4] are sampling-based inference LDA methods using Gibbs sampling that propose efficient topic inference to text streaming collections. Moving from document topics to event detection requires to introduce other attributes such as spatial and temporal parameters to the LDA model. This lead Pan and Mitra [15] to combine the LDA model with temporal segmentation and spatial clustering in two distinct methods: the proposed 3S-LDA is a LDA done in three step, the document topic assignment, a temporal segmentation and finally spacial clustering; the Space-Time LDA is a spatial latent Dirichlet (SLDA) allocation [26] adapted from the detection of segments in images to the detection events in text corpus. Those algorithms present good event detection in the TDT3 [9] and Reuters news dataset but they were not tested in text streaming.

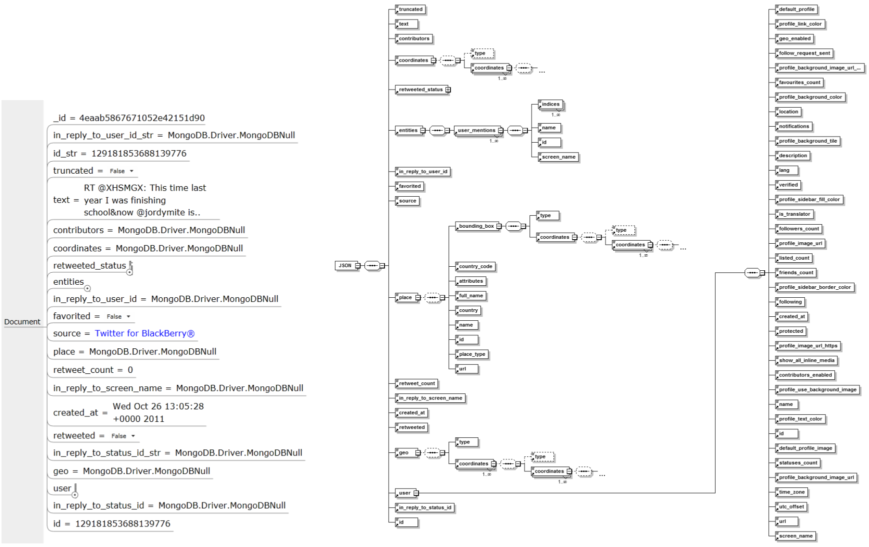
3 The twitter stream

The data set object of analysis will be retrieved using the twitter streaming API (`statuses/sample` method) that, using the default access level ('Spritzer'), returns a random sample of all public tweets [24]. This access level provides a small proportion of all public tweets (1%) [25]. The twitter stream API gives also access the firehose: 100% and gardenhose: 10% streams [24] using special accounts.

The data returned is a set of documents, one per tweet, in the JavaScript Object Notation [6]. These documents (Figure 1a), apart from the text of the tweet, contain additional data like tweet information i.e.: date, source of tweet, type; user information i.e.: profile, location and counters for favourites, friends, followers, etc.; entities mentioned in the tweet text i.e.: urls, hashtags and user, among other information.

Given the average number of 140 million tweets sent per day referred by Twitter [23], it is expected that the size of the data retrieved by the Streaming API (1%), in a 24 hour time span, will be roughly 1.400.000 tweets. The 140-character limit of tweets give an expected 196 MBytes per day or 2269 Bytes per second data stream.

The json tweet document contains attributes describing the tweet, user information, tweet relations with other tweets, a lists of urls, hashtags and user mentions contained in the tweet. In some cases information related with the lo-


Document

_id = 4eaab5867671052e42151d90
in_reply_to_user_id_str = MongoDB.Driver.MongoDBNull
id_str = 129181853688139776
truncated = False
text = RT @XHSMGX: This time last year I was finishing school&now @Jordymite is..
contributors = MongoDB.Driver.MongoDBNull
coordinates = MongoDB.Driver.MongoDBNull
retweeted_status
entities
in_reply_to_user_id = MongoDB.Driver.MongoDBNull
favorited = False
source = Twitter for BlackBerry®
place = MongoDB.Driver.MongoDBNull
retweet_count = 0
in_reply_to_screen_name = MongoDB.Driver.MongoDBNull
created_at = Wed Oct 26 13:05:28 +0000 2011
retweeted = False
in_reply_to_status_id_str = MongoDB.Driver.MongoDBNull
geo = MongoDB.Driver.MongoDBNull
user
in_reply_to_status_id = MongoDB.Driver.MongoDBNull
id = 129181853688139776

(a) tweet json sample (b) tweet structure

Fig. 1: Tweet document structure and json sample

cation of the user is also provided in the document. The Figure 1b shows the list of attributes and the relations between entities in the document.

4 Data stream event detection

Given the volume of data and the real-time nature of the twitter stream, event detection should be processed in an online manner. It's objective is to identify abrupt rises in the use of sets of words that could point to the occurrence of events. Unlike Sentiment Analysis and Opinion Mining algorithms, analysis is not based on a restricted set of tweets mentioning predefined keywords, in this case events are not known *a priori*, so data analysis cannot be confined to filtering techniques using only tweets mentioning sets of keywords. The event detection should address the maximum number of tweets and therefore instead of using classical text mining algorithms it should consider mining algorithms adapted to the processing of data streams.

In conventional machine learning algorithms the training data is available as a whole set. On the opposite, when the training set is a potentially endless flow of data arriving in an order that cannot be controlled, we consider that we are in the presence of a data mining algorithm that is able to learn from a stream. Data stream mining algorithms can be addressed by classical data mining algorithms that met the following requirements [2]:

- **Process an example at time, and inspect it only once:** No random access to the data being supplied. Examples are accepted as they arrive and in the arriving order. Algorithms can remember previous examples but should keep the used resources at a minimum level. Algorithms that require more than one pass to operate are topically not suitable for data streaming mining;
- **Use a limited amount of memory:** Memory usage can be divided in two parts the memory used to store running statistics and the memory used to store the current model;
- **Work in a limited amount of time:** To be capable of working in real-time, it must process the examples as fast or fastest than they arrive;
- **Be ready to predict at any point:** It should be able to produce the best model after seeing any number of examples. The process of generation of the model should be as efficient as possible. Final model generation should be direct and avoiding the re-computation of the model based on running statistics.

5 Twitter event detection based on wavelet analysis of hashtag mentions

None of the event detection methods described in section 2 met all the requirements for data stream mining outlined in section 4, therefore they are not suitable to perform real-time event detection in the twitter stream. Hashtag mentions of each tweet are provided in the tweet document returned by the public stream in node `entities hashtags` list (Figure 1b) and replace word occurrences in tweet text. Building occurrence signals from that hashtags it is more cost effective when compared to the building of tweet text word co-occurrences that require word tokenization and pre-preprocessing of the tweets text to remove irrelevant words and other entities like user mentions, urls and hashtags. Given the amount of data needed to be processed, topic detection applied in real-time to the Spritzer twitter stream [25] may not be feasible even with the SparseLDA [34] or O-LDA [4] that are described as real-time approaches to detect latent topics data streams. The proposed approach relies in the monitoring of hashtag mention signals using wavelet signal analysis. Because topic inference models are computationally expensive and require to be retrained each time a new tweet arrives, they will be only used in a later stage of the processing stage. In the current proposal LDA is applied only to hashtags signals that were identified as events at a given time interval and are used to estimate topics that help to describe the event itself.

Figure 2 shows the proposed workflow to perform the detection of events in the twitter stream. The following sections describe each one of the steps in detail.

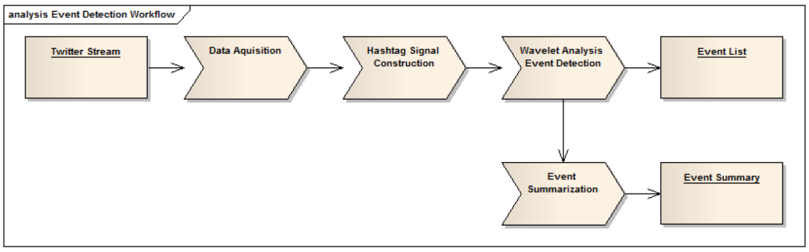

analysis Event Detection Workflow
Twitter Stream
Data Aquisition
Hashtag Signal Construction
Wavelet Analysis Event Detection
Event List
Event Summarization
Event Summary

Fig. 2: Event Detection Workflow

5.1 Data acquisition

The data set acquisition was done getting the tweet documents in json format and inserting them in a document-oriented database. Two tools were used in this task, cURL [28] to access the twitter stream and MongoDB [29] to store the data set in a way that can be queried. The Code 1.1 shows the command to insert live twitter data in the database. All the tweets retrieved by the cURL were stored in a collection named `tweets` in a database called `twitter`.

```
curl -k https://stream.twitter.com/1/statuses/sample.json -uUSER:PASS |
    mongoimport -d twitter -c tweets
```

Code 1.1: Data acquisition of twitter data set

5.2 Hashtag signal construction

The event detection of the twitter stream was made by analyzing the evolution of hashtag mentions over the time in the twitter's public stream. There are several transformations necessary to be performed prior to the event detection using wavelet analysis. Event detection is based on the peak analysis of individual hashtags mentions over the time (hashtag signal now on). To build such signals, for each one of the hashtags mentioned in tweets, it was necessary to build individual mention counts for each one of the hashtags over the time.

The data preparation process was made using map reduce transformations [8]. MongoDB supports map reduce transformations using javascript functions [13]. Therefore, all data preparation algorithms were performed on the database side. Processing the twitter stream data source using this approach removed unnecessary data transfers involving high amounts of data transfers across external components and lead to an increase in the performance compared to other approaches. To build the individual hashtags signals it was necessary to perform two map reduce transformations:

Extraction of hashtags from tweet documents The first step was the extraction of the hashtags mentioned in the the tweet documents. Using a map reduce transformation all hashtags were retrieved from tweets and then grouped in intervals of 5 minutes. The Code 1.2 shows the json document resulted after that transformation for a single 5 minute interval. The json document contains all the hashtags mentioned in the tweets posted between 21:45:00 and 16:50:00 in the 12th November 2011 (Code 1.2 line 1). The `hashtags` list show each one of the hashtags mentioned in the time period (Code 1.2 lines 4 to 9). In the `totaltext` list is shown the tweet text where the hashtag was mentioned. In order to improve the topic inference process described in section 5.4, all hashtags, URLs and user mentions were removed from the tweets text (Code 1.2 lines 11 to 16,). The `totalcount` shows the total number of hashtags mentions in the given time period, Code 1.2 line 17 shows that were used 596 hashtag mentions in this 5 minute period.

```
1  { _id: ISODate(2011-11-12T21:45:00Z),
2    value: {
3      hashtags: [
4        ‘currucucu’,
5        ‘kappennu’,
6        ‘thingslongerthankimsmarriage’,
7        ‘eribelieberbbb12’,
8        [...],
9        ‘stilldontliketheguy’  ],
10     totaltext: [
11        ‘jajja sii lo poco jaja  porfa saludes a de’,
12        ‘roept geesten op en ze wilt niet stoppen’,
13        ‘the time a black person gets to live in a horror film’,
14        ‘gente usa a tag’,
15        [...],
16        ‘dont know how i feel a bout todd halley’   ],
17     totalcount: 596  } }
```

Code 1.2: Map reduce result of hashtag mentions in an interval of 5 minutes

Construction of individual hashtag mention signals The final map reduce transformation is made to build hashtag mention signals over the time. Two simultaneous transformations were done in this map reduce. It is necessary to count the hashtag mentions in each 5 minute interval and grouping them in separated time series one for each hashtag. In Code 1.3 is shown an example were for each hashtag the `timeline` node contains pairs of timestamps (unix epoch [30]) associated with a count of the number of times the hashtag was mentioned over the time. In Code 1.3 line 1 is shown the time series for the “np” hashtag. The `textline` node contains a concatenation of all the tweets text that had the hashtag mentioned in the given time interval (Code 1.3 lines 11 to 16). This text will be used in section 5.4 to summarize the event with related topics once that hashtags itself may not be self explanatory. Line 18 in Code 1.3 shows the total number of times the hashtag “np” was mentioned in the whole time series (all 5 minutes intervals).

```
1  { _id: ‘np’,
2    value: {
    timeline: {
      1320183900000: 2,
      1320184200000: 17,
      1320184500000: 21,
      1320184800000: 13,
      [...],
      1320188400000: 17  },
    textline: {
      1320183900000: 'favor wale illest btch',
      1320184200000: 'i do no days off long walk jill [...]' ,
      1320184500000: 'twisting stank curreny [...]' ,
      1320184800000: 'trey songz missing you remix [...]' ,
      [...],
      1320188400000: 'marsha ambrosius late nights and [...]'
    },
    totalcount: 20648  } }
```

Code 1.3: Map reduce result of the evolution of hashtag "np" over the time ("np" signal)

5.3 Event detection using wavelet hashtag signal analysis

Signals built from Hashtag count mentions were assumed as being the basis for twitter event detection. It was also assumed that they represent the evolution of trends in the twitter stream. The map reduce process described in section 5.2 produced one signal for each of the hashtags mentioned in the defined time interval. Those signals represent time series in the evolution of topic mentions in the tweet stream. It was considered that an abrupt increase on the mention of a hashtag may result in a possible event that is happening at a given time. Wavelet analysis is a well know signal processing method to detect changes and peaks in signals [5]. The continuous wavelet transform (CWT) construct a time-frequency representation of a signal that offers very good time and frequency localization. Two wavelet tools where used to detect events in the twitter stream: the peak analysis was used to detect peaks in the hashtag signal; local maxima detection was used to detect changes in the hashtag signal. Because of the noisy nature of the tweet stream – some hashtag signals have a high variance between consecutive time intervals – signals were preprocessed using Kolmogorov-Zurbenko Adaptive Filters [31] to retrieve the trend of the hashtag signals (Code 1.4 line 3). This step removed some noise from hashtag signals and lead to a better wavelet peak and local maxima detection. Prior to the continuous wavelet transform (CWT) (Code 1.4 line 6), the signal was transformed in a time series (Code 1.4 lines 4 and 5). Extrema locations (in time and in scale) are calculated based on the CWT of the signal (Code 1.4 line 7). Finally, the peak detection in the time series was done and found the local maxima in the each one of the time series via a CWT tree (Code 1.4 line 8). The event detection algorithm was implemented in R (programming language) [18] and results inserted back into the MongoDB database using the resulting json documents.

```
1 foreach (hashtag_signal in twitter_stream)
2 {
3     kzsignal <- kz ( hashtag_signal )
```

```
    x <- getTimeInterval (hashtag_signal)
    y <- signalSeries(kzsignal, x))
   W <- wavCWT(y, wavelet="gaussian2")
   W.tree <- wavCWTTree(W)[1:100]
    p <- wavCWTPeaks(W.tree)
}
```

Code 1.4: Pseudocode for event detection in R programming

5.4 Event summarization with LDA topic inference

By monitoring the evolution in hashtag mentions, detected events are a list of hashtags that had a peak in mention counts at a given time interval. Some hashtags may be self explanatory of the event itself like "7bilhoesdepessoasnomundo" that revel tweets related with the 7th Billionth Child Born in 31th October 2011 or "papandreou" in tweets related with the greek PM Georgios Papandreou resignation the 9th November 2011. Other hashtags names representing events may not be self explanatory and may need additional information to describe human perceptible event descriptions. To archive this, in parallel when an event is detected, an topic inference algorithm is applied on all the tweets text related with the hashtag in each one of the time series 5 minutes interval. The idea is to extract latent topics using Latent Dirichlet Allocation (LDA) [3] from the tweets text. This method improved the hashtag description summarize the event with a set of topics inferred from tweets belonging to the time interval were the event occurred.

The process of extracting latent topics is done for each hashtag signal obtained from the process described in section 5.2. Each time interval of the `textline` node is considered as document used to train the LDA model. The Code 1.3 (lines 11 to 16) show the input text documents used in the the topic inference process for the "np" hashtag signal. In each one one of the time intervals was created a document-term matrix to be passed to the LDA algorithm. Topic were estimated by the LDA model using Gibbs Sampling, where each document represents the text of all the tweets referring that hashtags in individual intervals of 5 minutes. After the LDA estimation, in each 5 minute interval of the hashtag signal were retrieved 5 topics representative of the hashtag at that given moment. Code 1.5 line 11 shown the 5 extracted topics (t1 to t5) for hashtag "np" between 22:40 and 22:45 of the 1st November of 2011. Note that in this case the topics vary in time depending on the considered time interval (Code 1.5 line 6 show the extracted topics for the previous 5 minute interval and line 15 with topics for the following 5 minutes). In conjunction with the timestamp of the event detection this process will retrieve the 5 topics relevant to the hashtag in using tweets of the timespan where the event occurred.

```
1 [...],
2 { hashtag: 'np',
3   time: 1320186900000,
4   count: 11,
5   text: 'onetimelt this song never gets [...]',
  t1: ‘nirvana’, t2: ‘andando’, t3: ‘aidonia’, t4: ‘anastacia’, t5: ‘np’
        },
{ hashtag: ‘np’,
  time: 1320187200000,
  count: 19,
  text: ‘make me proud special k placebo ok [...]’,
  t1: ‘nirvana’, t2: ‘ambrosius’, t3: ‘anything’, t4: ‘afraid’, t5: ‘
      chainz’ },
{ hashtag: ‘np,
  time: 1320187500000,
  [...],
  t1: ‘np’, t2: ‘afraid’, t3: ‘anything’, t4: ‘ashanti’, t5: ‘better’
      },[...]
```

Code 1.5: Summarization of hashtag “np” over the time using Latent Dirichlet Allocation (LDA)

6 Experimental results

The the Spritzer twitter stream [25] was monitored between 00:00 of the 10th of November and 23:59 of 18th of November of 2011 totalizing more than 192 hours of data acquisition. In this time period were retrieved 13.651.464 tweets, this gives almost 72.000 tweets per hour and an average of 1.7 million tweets per day. These values were far above that ones expected by estimations made in section 3. In this time interval were mentioned 493.050 distinct hashtags meaning that were constructed 493.050 hashtags signals to be monitored. Table 1 lists the top 20 most mentioned hashtags in the time interval considered.

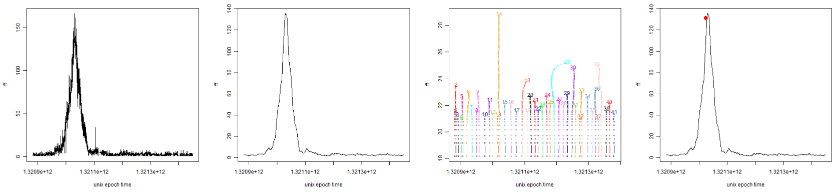
Table 1: Top 20 popular hashtags and mentions counts

#	hashtag	mentions
1	ff	23872
2	np	20648
3	teamfollowback	19088
4	oomf	13687
5	useatwitternameinasentence	13640
6	nowplaying	11249
7	nf	7979
8	rt	7741
9	fb	7719
10	thingspeopleshouldntdo	6189
11	xfactor	5929
12	ows	5800
13	99fm	5511
14	iwannabe	5225
15	followback	5134
16	bahrain	4981
17	tfb	4726
18	jobs	4632
19	myweddingsong	4386
20	peopleschoice	4160

Results of event detection and topic summarization are presented in the following subsections.

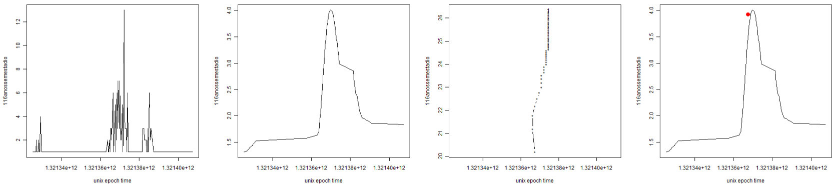
6.1 Event detection results

The event detection was done in 4 steps: building of hashtag signals, Kolmogorov-Zurbenko filtering, extrema detection using the continuous wavelet transformation and peak detection. Figure 3 and Figure 4 show a visual detail of each one of those steps for two hashtags ("ff" and "116anossemestadio"). The red dots shows the detected events at a given timestamp: Figure 3d shows the detection of an event in timestamp 1321029000000 (Friday, 11 Nov 2011 16:30:00 GMT) and Figure 4d an event in timestamp 1321372200000 (Tuesday, 15 Nov 2011 15:50:00 GMT). The "ff" event is related to the FollowFriday twitter trend that occurs every friday and where some twitter users suggest other twitter users to be followed. The "116anossemestadio" is related to the 116th anniversary of the Clube de Regatas do Flamengo Brazilian sports club in November 15, 2011. Remarks that Figure 3c and Figure 4c shown the results of the calculation of all local maxima and local minima for a sliding windows across the signal evolution. Each branch represent a change in the signal evolution. This continuous wavelet transform tree will be the basis used to detect peaks in the signal described in Code 1.4.



(a) original (b) filtered (c) cwt extrema (d) peak detection

Fig. 3: Peak detection for "ff" hashtag



(a) original (b) filtered (c) cwt extrema (d) peak detection

Fig. 4: Peak detection for "116anossemestadio" hashtag

Results of detection of events are presented in Table 2. This table contains a list of hastags and respective event datetime converted from the unix epoch timestamp.

6.2 Event summarization results

Event summarization results are presented in Table 2. The last column of the table shows a summary of the event using 5 topics inferred from tweets that had mentions to the event hashtag in that specific 5 minute interval. Additionally to the hashtag name the infered 5 topics help to summarize the occurrence of the event.

7 Conclusions

The twitter public stream mining using wavelet signal analysis of hashtag occurrences proved to be a valid lightweight option to event detection in this specific real-time and high throughput data stream. The data acquisition of the stream using a document based database, the use of twitter hashtag names included in the stream and the data preprocessing using map reduce transformations performed inside the database, led to an efficient hashtag signal construction. Event detection using peak detection on continuous wavelet transformed hastag signal provided good results in the identification of hashtags mentions bursts independently of being isolated events or events repeated over the time. The proposed technique to describe the detected events using a Latent Dirichlet Allocation topic inference model led to an better description of the event. With this method, detected events by wavelet analysis are enriched with 5 inferred topics from tweets occurred at that given time. Based on the experimental results, in most cases the combination of both hashtag names and inferred topics gave useful description information about the event. For future work and improvements there were identified 3 main topics: hashtag manipulation with the objective to group distinct hashtags related with the same event (“f1jp”, “f1nijigen”, “f1”, “f1chat”, “f1gp”); event repetition learning algorithms to identify and ignore periodic events like the follow friday (“ff”) that result in an event detected every friday; faster topic inference using sampling-based LDA methods applied to text streaming collections like the SparseLDA, FastLDA or O-LDA real-time implementations.

Table 2: List of events detected and respective summarization with topics

hashtag	datetime	5 main topics
116anossemestadio	15 Nov 15:50	flamenguista, 116anossemestadio, estdio, dentro, 116anossemestadio
49ers	13 Nov 23:35	49ers, giants, harbaugh, conversion, touchdown
argentina	11 Nov 21:15	moreno, bolivia, demichelis, jugando, mataaar
berlusconi	12 Nov 20:55	silvio, coglioni, berlusconi, boggles, career
blacksabbath	11 Nov 21:20	bezetting, meninos, blacksabbath, aaaeeewwww, blacksabbath
boosie	14 Nov 17:15	charger, loaded, loaded, better, hunnids
calle13	11 Nov 04:00	actitud, hahahha, celebraron, akabara, cambio
carrierclassic	12 Nov 00:05	correction, basketball, should, uniforms, student
cmaawards	10 Nov 01:40	natasha, watching, watching, taylor, country
cmas	10 Nov 01:50	absolutely, country, country, delicious, aldean
f1	13 Nov 13:05	vettel, actually, forward, campeonato, chequerd
ff	11 Nov 16:30	follow, akexwbc, asegura, ff, excuse
fimdomundoinforma	11 Nov 16:10	sobrevivi, sextafeira, dinheiro, evento, acabar
flamengo116anos	15 Nov 13:05	parabns, estdio, planeta, eterna, parabns
latingrammys	11 Nov 01:20	shakira, franco, premios, bolivar, ascendencia
lilwaynewackestpunchlines	10 Nov 06:45	lilwaynewackestpunchlines, phenomenal, should, phenomenal, almost
mareoflores	13 Nov 22:50	pretenden, siguen, libead, mareoflores, detencin
pacquiao	13 Nov 05:50	marquez, booing, pacquiao, boxing, fighting
raw	15 Nov 01:20	tengok, watching, followers, appearance, boston
tvoh	11 Nov 21:25	degene, kijken, tvoh, ahhaha, geniaal
vergonharecord	14 Nov 00:10	outras, arrependase, tendenciosa, atacando, comprar
veteransday	11 Nov 14:30	country, alabama, served, events, respect
walkingdead	14 Nov 02:45	stupid, walkingdead, alucinaciones, andrea, stupid
wish111111	10 Nov 15:20	tomorrow, person, better, awards, closed
wwe	15 Nov 01:20	cpeatt, welcome, attacked, hometown, wwe
xfactor	13 Nov 17:55	technical, amelia, blowing, devlin, beautiful

References

[1] Allan, J., Lavrenko, V., Malin, D., Swan, R.: Detections, Bounds, and Timelines: UMass and TDT-3 (2007), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.2547

[2] Bifet, A., Kirkby, R.: Data stream mining: a practical approach. Tech. rep., The University of Waikato (Aug 2009)

[3] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)

[4] Canini, K.R., Shi, L., Griffiths, T.L.: Online inference of topics with latent dirichlet allocation. Journal of Machine Learning Research - Proceedings Track 5, 65–72 (2009)

[5] Chen, L., Roy, A.: Event detection from flickr data through wavelet-based spatial analysis. In: Cheung, D.W.L., Song, I.Y., Chu, W.W., Hu, X., Lin, J.J. (eds.) CIKM. pp. 523–532. ACM (2009)

[6] Crockford, D.: RFC 4627 - The application/json Media Type for JavaScript Object Notation (JSON). Tech. rep., IETF (2006), http://tools.ietf.org/html/rfc4627

[7] Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S.: Locality-sensitive hashing scheme based on p-stable distributions. In: Snoeyink, J., Boissonnat, J.D. (eds.) Symposium on Computational Geometry. pp. 253–262. ACM (2004)

[8] Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Commun. ACM 51(1), 107–113 (2008)

[9] Graff, D., Cieri, C., Strassel, S., Martey, N.: The tdt-3 text and speech corpus. In: in Proceedings of DARPA Broadcast News Workshop. pp. 57–60. Morgan Kaufmann (1999)

[10] Indyk, P., Motwani, R.: Approximate nearest neighbors: Towards removing the curse of dimensionality. In: STOC. pp. 604–613 (1998)

[11] Kaplan, A.M., Haenlein, M.: The early bird catches the news: Nine things you should know about micro-blogging. Business Horizons (Oct 2010), http://dx.doi.org/10.1016/j.bushor.2010.09.004

[12] Kleinberg, J.: Bursty and hierarchical structure in streams. Data Min. Knowl. Discov. 7, 373–397 (October 2003), http://dl.acm.org/citation.cfm?id=861097.861114

[13] MongoDB: Mapreduce — mongodb manual (2011), http://www.mongodb.org/display/DOCS/MapReduce, [Online; accessed 29-December-2011]

[14] Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) LREC. European Language Resources Association (2010)

[15] Pan, C.C., Mitra, P.: Event detection with spatial latent dirichlet allocation. In: Newton, G., Wright, M., Cassel, L.N. (eds.) JCDL. pp. 349–358. ACM (2011)

[16] Petrovic, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to twitter. In: HLT-NAACL. pp. 181–189. The Association for Computational Linguistics (2010)

[17] Porteous, I., Asuncion, A., Newman, D., Smyth, P., Ihler, A., Welling, M.: Fast collapsed gibbs sampling for latent dirichlet allocation. In: In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 569–577 (2008)

[18] R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2010), http://www.R-project.org, ISBN 3-900051-07-0

[19] RyanăKelly: Pearanalytics - twitter study - august 2009 (2009), http://www.pearanalytics.com/blog/wp-content/uploads/2010/05/Twitter-Study-August-2009.pdf, [Online; accessed 21-November-2011]

[20] Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web. pp. 851–860. WWW '10, ACM, New York, NY, USA (Apr 2010), http://dx.doi.org/10.1145/1772690.1772777

[21] Schonfeld, E.: Techcrunch: Mining the thought stream (2009), http://techcrunch.com/2009/02/15/mining-the-thought-stream, [Online; accessed 14-November-2011]

[22] Shiels, M.: BBC News: Twitter co-founder Jack Dorsey rejoins company. http://www.bbc.co.uk/news/business-12889048 (2011), [Online; accessed 01-November-2011]

[23] Twitter: Twitter Blog: #numbers. http://blog.twitter.com/2011/03/numbers.html (2011), [Online; accessed 01-November-2011]

[24] Twitter: Twitter Developers: Streaming API. https://dev.twitter.com/docs/streaming-api (2011), [Online; accessed 01-November-2011]

[25] Twitter: Twitter Developers: Streaming API Methods. https://dev.twitter.com/docs/streaming-api/methods (2011), [Online; accessed 01-November-2011]

[26] Wang, X., Grimson, E.: Spatial latent dirichlet allocation. In: Platt, J.C., Koller, D., Singer, Y., Roweis, S.T. (eds.) NIPS. MIT Press (2007)

[27] Weng, J., Yao, Y., Leonardi, E., Lee, F.: Event Detection in Twitter. Tech. rep., HP Labs (2011), http://www.hpl.hp.com/techreports/2011/HPL-2011-98.html

[28] Wikipedia: curl — Wikipedia, the free encyclopedia (2011), http://en.wikipedia.org/wiki/CURL, [Online; accessed 01-November-2011]

[29] Wikipedia: Mongodb — Wikipedia, the free encyclopedia (2011), http://en.wikipedia.org/wiki/MongoDB, [Online; accessed 01-November-2011]

[30] Wikipedia: Unix time — Wikipedia, the free encyclopedia (2011), http://en.wikipedia.org/wiki/Unix_time, [Online; accessed 29-December-2011]

[31] Yang, W., Zurbenko, I.: Kolmogorov-zurbenko filters. Wiley Interdisciplinary Reviews: Computational Statistics 2(3), 340–351 (2010), http://dx.doi.org/10.1002/wics.71

[32] Yang, Y., Pierce, T., Carbonell, J.G.: A study of retrospective and on-line event detection. In: SIGIR. pp. 28–36. ACM (1998)

[33] Yang, Y., Pierce, T., Carbonell, J.: A study of retrospective and on-line event detection. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. pp. 28–36. SIGIR '98, ACM, New York, NY, USA (1998), `http://dx.doi.org/10.1145/290941.290953`

[34] Yao, L., Mimno, D.M., McCallum, A.: Efficient methods for topic model inference on streaming document collections. In: IV, J.F.E., Fogelman-Soulié, F., Flach, P.A., Zaki, M.J. (eds.) KDD. pp. 937–946. ACM (2009)

Classification of Sentiment Polarity of Portuguese On-line News

Inês Coimbra Morgado

Department of Informatics Engineering,
Faculty of Engineering, University of Porto,
Porto, Portugal
coimbra.ines@fe.up.pt

Abstract. Sentiment polarity analysis is a subset of text classification aiming to classify text according to its sentiment polarity: positive, neutral and negative. This paper proposes an approach to classify Portuguese on-line news according to their sentiment polarity, in order to verify the usefulness of a non-supervised learning approach. To do so, each sentence of each news is assigned a likelihood of belonging to each class. This likelihood depends both on the content of the sentence and on the influence of nearby sentences. Once analysed, each news may be visualised as an *HTML* document with each sentence highlighted according to its sentiment polarity. Forty news (two hundred and fifty sentences) were manually classified to enable results analysis. It is concluded that it is important to take the influence of nearby sentences into account even though this influence should be as little as possible and that an accurate automatic classification is possible.

Keywords: Information Retrieval; Sentiment Polarity; Text Classification

1 Introduction

People everywhere are trying to figure out new ways of obtaining information as quickly as possible. It is fundamental to be able to collect and analyse relevant information about the subject at hand. Without any automatic, or semi-automatic, way of accomplishing this, the analysis becomes a hand-made task, which disables the possibility of handling large amounts of data in a short period of time.

One of the areas that tries to be of assistance to this problem is the information retrieval field of study, whose major goal is to analyse large amounts of data, as automatically as possible, extracting any relevant information it may contain. Some questions arise from this, such as which data to analyse, how to analyse non-structured text, how to distinguish relevant from non-relevant information and what to do with the extracted information.

Depending on the interests of who is collecting the information, the subject varies tremendously. An application of these studies is, for example, analysing

Twitter messages in pre-election periods to extract public opinion on each candidate, in order to anticipate vote intentions.

Text classification is a quite explored field of study. Black [1] and Hearst [2] developed some work that led to the definition of the Naive-Bayes Classifier by Gale *et al.* [3], which is one of the most used methods for text classification. Other studies, like the one of Tong *et al.* [4], use the well known Support Vector Machines (SVM) [5].

In this paper, the problem at hand is to identify the polarity of sentiments of Portuguese on-line news, *i.e.*, if they are positive, negative or simply neutral. An example of an application of this is the analysis of how the type of news available may affect the state of mind of the readers, such as matching a huge amount of negative news to the increase of depression in the community. However, this will not be subject of further study in this paper, being its focus on the techniques used to classify the news.

Sentiment analysis is a subset of text classification. This way, the Naive-Bayes Classifier and Support Vector Machine (SVM) are two widely used methods. Examples of such are the works of Pang *et al.* [6] and Kennedy *et al.* [7]. These are explained further in Section 2.2.

The approach presented in this paper divides each of the news into sentences and each sentence is analysed according to its sentiment polarity, based on the number of positive and negative words in the sentence and on the polarity of the nearby sentences. The results can be visualised in the format of an *HTML* document, having each sentence highlighted according to its sentiment polarity. In the end, the approach will be evaluated in terms of accuracy, precision and recall, through a comparison between the results obtained by the approach and a manual classification of some news.

This paper is organised as follows. Section 2 presents and analyses the already produced work on this field. Section 3 characterises the dataset that was used to validate the approach. Section 4 describes the methodology used in this work and how the results were evaluated. Section 5 presents the results obtained. Section 6 presents the conclusions drawn from this study and some future work.

2 Related Literature

Information retrieval is a broad field of research and, therefore, the number of related studies is immense. As such, it is necessary to focus the state of the art on text classification studies, as the developed work was, after all, a text classification problem.

2.1 Text Classification

Text classification is usually achieved through supervised learning, *i.e.*, it is necessary to have a sample on which to base the analysis. This way, text classification is usually divided in two phases: a learning or training phase and a classification phase. The first receives a previously hand classified set of documents and, based on these, it defines a function which, given a document d classifies it in a class c. The latter, receives a document d and, using the previously defined function, outputs its classification [8].

There are two main ways of representing text for classification, through a list of words or a bag of words. The difference between these two cases is that in the first, the order in which the words appear in the list matters, whilst in the latter, the only concern is the frequency of each word in the document. In both cases, it is usual to pre-process the text, such as stemming it or removing stop words.

The two main approaches to text classification problems are Naive-Bayes and SVM.

Naive-Bayes Classifier The Naive-Bayes Classifier was first introduced by Gale *et al.* [3] as a response to word disambiguation problems. This approach is based on hand-annotated approaches, such as the ones of Black [1, 9] or Hearst [2]. However, Gale *et al.* [3] replaced the hand-annotated text with parallel text, *i.e.*, they collect this information from large amounts of text like the Canadian Hansards [10]. This way, their approach was divided in two phases: a training phase, in which sentences of both senses of the word to disambiguate are selected and a classifier is defined, and a testing phase, in which they use the previously defined classifier to classify each document.

The idea is to build a probabilistic model based on the previously classified dataset, *i.e.*, given a certain document, what is the probability of it belonging to each category. Defining c as each of the categories in the set of categories and d as a certain document, the idea is to calculate $P(c|d)$ [11]. In order to accomplish this, it is necessary to calculate the probability of each word of the document belonging to each category, regardless of its position on the sentence. Naive-Bayes functions under the assumption that each feature is independent of the others. Even though this assumption is, in most cases, incorrect, the approach usually works correctly. The probability $P(c|d)$ can be calculated by:

$$P(c|d) = \frac{P(c)\prod_{w\in d} P(w|c)^{n_{wd}}}{P(d)},$$

where n_{wd} is the number of times the word w occurs in the document d, $P(w|c)$ is the probability of observing w if the class is c, $P(c)$ is the probability of the class being c, $P(d)$ is a constant that makes the probabilities for the different classes add up to one [11] and

$$P(w|c) = \frac{1+\sum_{d\in D_c} n_{wd}}{k+\sum_{w'}\sum d\in D_c n_{w'd}},$$

where D_c is the collection of all documents of the class c on the training dataset and k is the number of distinct words in the training dataset.

The classifier is a combination between the probabilistic model and a decision rule. The decision rule usually assigns the document to the category it is more likely to belong to.

Support Vector Machine Text classification (or data classification) is a recurrent problem in machine learning. SVM is a technique commonly used to overcome the problem of dividing data into two categories. Alike Naive-Bayes, SVM is a supervised learning technique and, therefore, it requires a pre-classified dataset. The main idea is to associate a vector in a p-dimensional space and a classification (1 or -1, according to the category it is inserted in) to each document and then define a $(p-1)$-dimensional hyperplane[1] which separates both categories [5]. The vector associated to each document is based on the different features the document may present. Ideally, this hyperplane should be as far apart from each category as possible, *i.e.*, the distance between the hyperplane and the closest point of each category should be maximised. This is called the maximum-margin hyperplane. The instances that lie most close to the hyperplane are called support vectors [4]. Figure 1 presents the maximum-margin hyperplane for a categorised dataset.

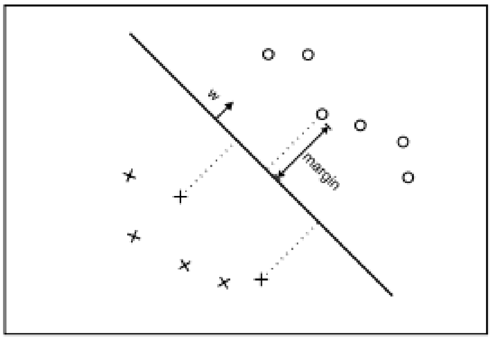


Fig. 1. Example of a maximum-margin hyperplane [4]

The choice of the maximum-margin hyperplane is a p-dimensional optimisation problem and, therefore, solving it offers computational challenges. However, there are already some approaches to solve this kind of problems like Ferris *et al.* [12] and John Platt [13]. In some cases the hyperplane may be a non-linear one. An example of such is depicted in Figure 2.

After defining the maximum-margin hyperplane, a vector (point) is assigned to each of the unclassified documents. Then, depending on which side of the hyperplane they are, they are considered as being of the corresponding category.

2.2 Sentiment Polarity Analysis

Sentiment analysis is a subset of text classification. Instead of trying to categorise a certain document, its goal is to understand the sentiment that document conveys. Moreover, it tries to identify which part of the document should be

[1] generalisation of a mathematical plane into a different number of dimensions

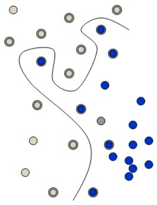


Fig. 2. Example of a non-linear hyperplane [14]

considered as positive and which should be considered negative. Therefore, this question has to be handled in a different way than simple text categorisation.

Minimum Cut Approach Pang *et al.* [6] describe a process of classifying movie reviews as positive or negative. They divided the analysis in two phases: first they were to identify the relevant sentences, *i.e.*, the prone to subjectivity sentences and then, using only the subjective sentences, they divided them in positive and negative. This way, they would obtain 3-way categorisation: positive, negative and neutral. In order to accomplish this, Pang *et al.* defined an undirected graph with n + *2* nodes, where n is the number of sentences (v) in the document and the other two are a source node (s) and a sink node (t), representing the classes to which each sentence should be associated (C_1 and C_2, respectively).

The graph has edges connecting s to every v_i and every v_i to t. The weight of an edge connecting s to a v_i is the probability of v_i being classified as C_1 (ind_1) and the weight of an edge connecting a v_i to t is the probability of v_i being classified as C_2 (ind_2). As each v_i is either C_1 or C_2, $ind_2(v_i) equalsto 1 - ind_1(v_i)$. The weight of an edge connecting a v_i to a v_k is the probability of the classification of v_k influencing the classification of v_i, *i.e.*, how likely it is for both sentences having the same classification. In this paper this will be referred to as association score ($assoc(v_i, v_k)$).

In the first phase C_1 is objective and C_2 is subjective whilst in the second phase C_1 is positive and C_2 is negative.

In order to decide to which node a sentence must be associated, Pang *et al.* [6] associate a cost to each cut of the graph and they try to minimise it, applying a minimum cut algorithm. A cut is a set of nodes forming a path from s to t and its cost the sum of the weights of all the edges crossed in that path.

The minimum cut is the one with the minimum cost. Figure 3 depicts the representation of the first phase of the process.

In order to test their approach, Pang *et al.* [6] used a dataset of previously classified movie reviews. They divided their evaluation into two sections, one testing the efficiency of the subjectivity separation (first phase) and another one testing the importance of analysing the context in order to classify the text.

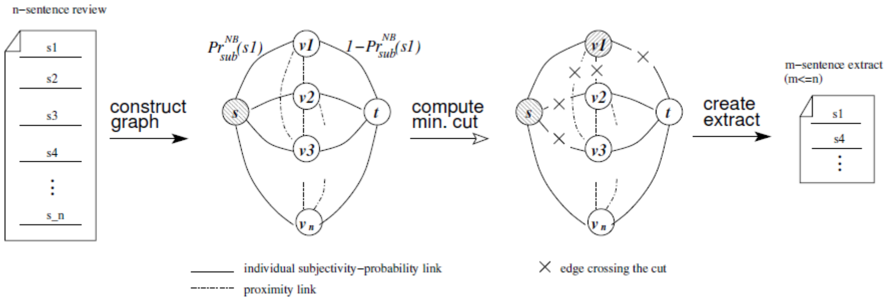


Fig. 3. Graph-cut-based creation of subjective extracts [6]

Contextual Valence Shifters Approach Kennedy *et al.*'s goal [7] is, similarly to Pang *et al.* [6], to automatically classify movie reviews' sentiment polarity. However, they use a different approach.

Kennedy *et al.* [7] try two different approaches. The first one strictly considers the number of positive and negative terms in a review: a review is positive/negative if it has more positive/negative terms than negative/positive ones. If the quantity of positive and negative terms are equal (or if the difference is within a defined margin), the review is considered neutral. In order to identify the polarity of each term, the *General Inquirer* [15] is used. It is yet taken into consideration the existence of terms that may change the polarity (or intensity of the polarity) of another term. These terms are denominated valence shifters (*not* is an example of a negative valence shifter).

The second approach uses a machine learning approach based on SVM, whose features are single words. Kennedy *et al.* [7] improve this approach by adding, as features, pairs of terms of the type *valence shifter + (positive/negative) term*, replacing the valence shifter by a value representative of its intensity.

In order to test their approach, Kennedy *et al.* [7] used the same dataset used by Pang *et al.* [6]. They present results for both approaches and for a third one which combines the two. Their thesis was that the introduction of valence shifters in sentiment analysis would improve it and the obtained results validate this.

3 Dataset Characterisation

In this work, a dataset of Portuguese on-line news was analysed in order to validate the approach. This dataset was collected by Sapo[2] and provided by Sapo Labs at the Faculty of Engineering of the University of Porto[3], as a comma separated values (*csv*) file.

[2] http://www.sapo.pt

[3] http://labs.sapo.pt/up/

The news were collected between the Ninth of September 2011 and the Seventeenth of November of 2011, enclosing approximately two hundred thousand news.

Each line of the file represents a document (a news) with six fields, as described in Table 1, where only the fields *tag* and *geo* are optional.

Table 1. Description of each field of the dataset

Field	**Description**
pubdate	The date the news was published in the format DD-MM-YYYY HH:MM.
title	The title of the news.
content	The content of the news.
tags	A word or words related to the news or to the topic of the news. Example: *Economia* (*Economy*, n English).
geo	Xml with geographical information like the city and district where the event described on the news took place
page_url	The url of the news

Even though this is the dataset to classify, another one was used as support for the classification. The dataset *SentiLex-PT 02* [16] is a sentiment lexicon for Portuguese, including adjectives, nouns, verbs and some idiomatic expressions. For each sample, the dataset indicates the target of the sentiment, the sentiment polarity and the polarity assignment. This way, each entry has the following fields:

- *PoS* (Part of Speech): *ADJ* for adjectives, *N* for nouns, *V* for verbs and *IDIOM* for idiomatic expressions;
- *TG* (Target of Polarity): the target of the sentiment which corresponds to human noun (*HUM*), functioning as the subject (*N0*) and/or the complement (*N1*) of the predicate;
- *POL* (Polarity): 1 for positive, 0 for neutral or -1 for negative;
- *ANOT* (Polarity Annotation): indicates the annotation was performed manually (*MAN*) or automatically (*JALC*).
- *REV*: an optional field which has specific notes about the annotation. If it is *REV=AMB* it means the word is ambiguous, *i.e.*, its polarity depends on the context and if it it *REV:POL* it indicates the polarity of this entry was modified in relation to the previous version of the dataset (*SentiLEx-PT 01*).

There are two versions of the *SentiLex-PT*: one with just the lemma of the words and another with the inflected forms of the words. In this work, the one containing just the lemma of the words was used.

4 Methodology

The approach followed was an adaptation of the one presented by Pang *et al.* [6], described in Section 2.2. Pang *et al.* [6] divided their approach in two phases: classification of sentences as objective or subjective and classification of the subjective sentences as positive or negative. Both phases were also divided in two phases: calculation the probability of a sentence being of one class, regardless of its context and calculation that same probability but taking into consideration the influence of the nearby sentences.

The followed methodology was yet a simplificationwas a simplification of this approach, as instead of dividing the classification in two phases, there was only one phase, *i.e.*, the probabilities of a word being neutral, positive or negative are calculated all at the same time. In order to do so, each news was divided into sentences, each sentence into words and the lemma of each word was obtained and its polarity evaluated. A word was considered positive if the polarity was 1, neutral if 0 and negative if -1. However, if the word was not present in the sentiment dataset, it would not be considered. There were also some words whose polarity was not taken into consideration. These were the so called stop words, which were mainly articles, like *a* or *the*. The list of Portuguese stop words that was used in this project can be found in [17]. Alike Kennedy *et al.*'s work [7], negative valence shifters have also been also considered, *i.e.*, if a word was near a negative valence shifter, its polarity would become the opposite (negative became positive and vice-versa). A word was considered to be near to another if they were at most six words apart. The negative valence shifters considered in this work were the Portuguese versions of *no*, *never*, *none* and *nor*.

Alike Pang *et al.*'s approach [6], this one also divided the calculation of the probability of each sentence being positive, neutral or negative in two phases. First, the probability which did not consider the context of the sentence was calculated. Then, the surroundings were taken into consideration. In order to do so, counters for the number of positive, neutral and negative words were set, being updated each time a word was found in the sentiment dataset. As the end of the sentence was reached, three probabilities (positive, neutral and negative) were associated to each sentence. These corresponded to the ratio between the number of words of a certain class and the sum of all the classified words. For example, the probability of a sentence being positive, regardless of its surroundings would be:

$$P_{sentence}(positive) = \frac{positivewords}{positivewords+negativewords+neutralwords} * 100.$$

After all the sentences of a certain news had been analysed, there was a second round, which would again assign three probabilities to each sentence. However, at this point, the calculation would take into consideration the previously calculated probabilities for the sentence itself and for the ones next to it. Two sentences were considered to be next to each other if they were at most two sentences apart. Four alternative influences of the nearby sentences were used in order to draw some conclusions about which was the best alternative: no influence, constant influence, influence decaying linearly with the distance and influence decaying exponentially with the distance. After the whole news had been analysed, it was classified according to the highest probability.

In order to understand if this approach was valid, the obtained results had to be evaluated. The most accurate way to do so was to perform the test on a previously hand-classified dataset. As the dataset to be analysed had not yet been evaluated and due to time restrictions, forty news, which corresponds to about two hundred and fifty sentences, from the dataset were classified manually. Using this manual classification, values of accuracy, precision and recall were obtained for each of the three classes (positive, neutral and negative).

4.1 Details of the approach

Before the analysis process took place, the date files containing the stop words and the polarities were processed, creating an array for the stop words and a hash for the words in the polarity dataset and the corresponding polarity. During the analysis, in order to verify whether or not a word was a stop word, the word itself would be searched in the stop words array and, in order to determine the polarity of a word, its lemma would be searched in the polarity hash.

Even though the polarity dataset is supposed to contain the lemma of the words, in reality there are some words which do not correspond to the actual lemma. For example, the dataset contains the word *demitido* (*fired* in English) when the corresponding lemma is *demitir* (*fire* in English). This way, before inserting a word to the polarity hash, the corresponding lemma was obtained and is inserted to the hash instead of the original word.

When working with lemma it was necessary to be careful with the negative forms, *i.e.*, even though *responsible* and *irresponsible* have the same lemma, they have opposite polarities. In order to solve this issue, it was verified if the initial letters of a word corresponded to a negative preposition ('i' 'r' for example). If so and if the original word and the word without the preposition had the same lemma (like *irresponsible* and *responsible*), then the polarity would be set as of the opposite of the polarity of the lemma.

5 Results

After processing all the data, it was necessary to offer some way of visualisation of the results. Having the content of the news, it was possible to create an *HTML* document for each news, containing its content. The tags *<font color="green"></font>* or *<font color="red"></font>* were added before and after each positive and negative sentence, respectively. This way, each sentence would be presented as black if neutral, green if positive and red if negative.

As only forty news were classified manually, there was no need to analyse all the news of the dataset, *i.e.*, only the manually classified news were processed. Even though the more news were evaluated, the more valid the obtained results would be, two hundred and fifty sentences is enough to take some valid conclusions.

The whole process took about 1 minute, which is an average of 1,5 seconds per news.

As there are three different categories to which each sentence may be assigned, the accuracy, precision and recall values were calculated to each category.

Table 2 presents the results for a constant influence, Table 3 presents the results for a linear decay in influence and Table 4 presents the results for an exponential decay in influence.

Table 2. Accuracy, Precision and Recall for each category with constant influence of nearby sentences

	Accuracy (%)	**Precision (%)**	**Recall (%)**
Positive	62.698	16.484	45.455
Neutral	61.111	4.494	23.529
Negative	42.063	88.889	33.014
Average	55.291	36.622	33.556

Table 3. Accuracy, Precision and Recall for each category with linear decreasing influence of nearby sentences

	Accuracy (%)	**Precision (%)**	**Recall (%)**
Positive	63.095	16.484	46.875
Neutral	59.127	7.865	25.000
Negative	43.651	84.722	31.771
Average	55.291	36.357	34.549

Table 4. Accuracy, Precision and Recall for each category with exponential decreasing influence of nearby sentences

	Accuracy (%)	**Precision (%)**	**Recall (%)**
Positive	63.492	18.681	48.571
Neutral	62.302	12.360	39.286
Negative	44.048	83.333	31.746
Average	56.614	38.125	39.868

The results from the experiment with no influence from the nearby sentences are displayed in Table 5. This experiment was revealed to be useful to verify whether or not it was important to take into consideration the influence of these sentences.

Table 5. Accuracy, Precision and Recall for each category with no influence of nearby sentences

	Accuracy (%)	**Precision (%)**	**Recall (%)**
Positive	61.508	17.582	42.105
Neutral	61.905	20.225	41.860
Negative	46.429	75.000	31.579
Average	56.614	37.602	38.515

6 Conclusions and Future Work

There are two kinds of results to analyse: the accuracy, precision and recall of each category and the average of these values, for each influence. The latter is usually more significant as it provides an overview of the results, enabling more accurate conclusions.

From the results presented in Section 5, it is observable that, as the influence of the nearby sentences decays, the better the obtained results. However, without any influence whatsoever, the results' quality decreases. These results are logical as, in order to determine the polarity of a sentence, its context must be accounted for but the most important factor is its content.

The majority of values for accuracy obtained are above 50%, which represents a quite accurate classification, even though the values of precision and recall tend to be lower. One aspect that it is important to notice is the discrepancy between the values of precision for the negative sentences and the positive or neutral sentences. This may mean it would be advisable to re-examine the probabilities calculations.

Taking all this into consideration, it is possible conclude that the automatic classification of news according to their polarities is achievable, even though the approach presented needs some adjustments.

As for future work, the first topic of discussion should be the calculation of the different probabilities as the classification relays completely on this. It would also be interesting to enable the analysis of sets of words as a way to identify idiomatic expressions and their polarity. At this moment, the news are analysed word by word which makes this impossible. Some improvements can also be done to the visualisation. Ideally, it should extract the source code of the news and modify it, adding the tags for green and red text. However, due to the amount of different structures of the *HTML* documents, such was not possible. It would be interesting to tackle this question in the future, allowing the results to be presented with the original layout slightly modified.

References

1. Ezra W. Black. An experiment in computational discrimination of english word senses. *IBM J. Res. Dev.*, 32:185–194, March 1988.
2. Marti A. Hearst. Noun homograph disambiguation using local context in large text corpora. In *University of Waterloo*, pages 1–22, 1991.
3. W. A. Gale, K. W. Church, and D. Yarowsky. A method for disambiguating word senses in a large corpus. In *Computers and the Humanitzes*, volume 26, pages 415–439, 1993.
4. Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, March 2002.
5. William S Noble. What is a support vector machine? *Nature Biotechnology*, 24(12):1565–1567, 2006.
6. Bo Pang and Lillian Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.
7. Alistair Kennedy and Diana Inkpen. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2):110–125, 2006.
8. Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34:1–47, March 2002.
9. Ezra W. Black. *Towards computational discrimination of English word senses.* City University of New York, 1987.
10. Information Service of the Parliament of Canada. Parliament of canada. `http://www.parl.gc.ca/ParlBusiness.aspx?Language=E`. Online on November 2011.
11. Eibe Frank and R R Bouckaert. Naive bayes for text classification with unbalanced classes. *Knowledge Discovery in Databases PKDD 2006*, 4213:503–510, 2006.
12. Michael C. Ferris and Todd S. Munson. Interior-point methods for massive support vector machines. *SIAM J. on Optimization*, 13:783–804, August 2002.
13. John C. Platt. Using analytic qp and sparseness to speed training of support vector machines. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 557–563, Cambridge, MA, USA, 1999. MIT Press.
14. CDTRG. Svm - support vector machines. `http://www.dtreg.com/svm.htm`. Online on November 2011.
15. Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. *The General Inquirer: A Computer Approach to Content Analysis.* MIT Press, 1966.
16. Paula Carvalho, Mário J. Silva, and João Ramalho. Sentilex-pt 02 in english. `http://dmir.inesc-id.pt/reaction/SentiLex-PT_02_in_English`, 2011. Online on November 2011.
17. Stanley Loh. Listas de stopwords - stoplist (portugues, ingles, espanhol). `http://miningtext.blogspot.com/2008/11/listas-de-stopwords-stoplist-portugues.html`, 2011. Online on December 2011.

A Comparative Study of Hierarchical Clustering Algorithms for Tagging Systems

Anisa Allahdadi

BIHE, Iran
anisa.allahdadi@bihe.org

Abstract. With the rapid growth of information on the web, the so-called web2.0 services provide users with a simple way of managing a collection of resources. The collaborative nature of social bookmarking systems allows users to annotate their resources easily and explore other people resources in the network. However, data exploration in such large and complex networks is not always easy, due to lack of organizational structure. Recently several clustering approaches have been introduces to improve the data navigation and information exploration in such systems. In this paper we proposed a framework for the purpose of clustering and visualizing tag spaces to enhance data exploration capabilities. We presented two hierarchical clustering methods and applied them to the del.icio.us data set as one of the most popular social bookmarking systems. We finally made a comparative study on these two approaches regarding the clustering accuracy and time performance.

Keywords. web2.0, tags, collaborative, exploration, navigation, clustering, visualizing

1 Introduction

As the amount of information on the web grows rapidly, the issue of data exploration and retrieval becomes more crucial. Bookmarking systems allow users to annotate their resources with some keywords, called tags, which they consider most appropriate for the content. Users can also share both resources and tags with other users in the network. Tagging system normally consists of three main entities which make a tripartite network of a folksonomy [2]: users, resources, and tags. When a user applies a tag to a resource in the system, a tripartite relationship between the user, the resource and the tag is formed. The study of these relationships among the tripartite entities provides valuable knowledge to obtain:

- Semantic of tags will be easier to understand, as a tag can be defined or explained by the other tags in the group.
- Clusters of similar tags can be utilized for tag recommendation, tag ambiguity alleviation, and tag bundles suggestion.
- Clusters of related resources can be quite useful in the time of searching and recommending similar resources to users.
- The best topic for a resource and some Meta information about it can be offered automatically from representatives of tag bundles.

- Community of Like-minded users is provided by putting similar users in a group. The users in each group can share their resources, tagging behavior and so many similar experiences with each other.

The collection of user-generated tags is usually presented in the *tag cloud*, which is a visual display of tags. Tag clouds take the place of tag lists in which attributes of the text such as font size, color or weight are used to represent the importance or frequencies of the tags. Schrammel et al. [15] performed a research to investigate which layout for tag clouds is the most appropriate one for searching purposes. However there are some inherent limitations in tagging context. Most important of all is ambiguity. While different users apply different words to a resource, and it causes difficulty to find out what the user exactly mean by this word. Acronyms and synonyms are other sources of potential ambiguity as well.

A number of articles [12, 13] claimed that data mining techniques such as clustering, which finds groups of related tags, provide a mean to overcome these limitations mentioned above. With the help of clustering the effect of ambiguity can be alleviated. Gemmell et al. [12] expressed that tags can be aggregated in clusters and their ambiguity can be diminished since the ambiguous tags can be covered by the majority of tags in a cluster. Sbodio and Simpson [13] stated that clustering plays an important role in identifying the concepts of folksonomies and it might also help in automating the process of vocabularies extraction from unstructured folksonomies.

In this work, we focused on investigating the possibility of improving search and data exploration in collaborative tagging systems. We concentrated mainly on clustering like-minded users and similar bookmarks. Further works present that organizing bookmarks according to user interest and tagging manner may lead to even more interesting results. To achieve the mentioned objectives we design and implement a framework in Java called Tag Clustering and Visualizing Framework (TCV). The framework consists of four main sections.

- Data Extraction: allocated for fetching data from del.icio.us tagging system and forming three main matrices of user correlation, bookmark similarity and user similarity.
- Clustering: to cluster obtained data in the previous section based on two hierarchical clustering methods, greedy agglomerative and betweenness divisive.
- Visualization: to demonstrate both original data in the mentioned matrices and clustered data.
- Statistics: for creating excel charts according to the original data and clustering results.

The rest of this paper is organized as follows: Section 2 lists some related works. Section 3 presents the TCV framework and its structure and modules. In Section 4 the experiment results have been reported. And the conclusion and future works have been addressed in Section 5.

2 Related Work

Central to this work is the quality and performance of clustering. Previously in [14] Begelman et al. presented several clustering techniques and provide some results on del.icio.us and RawSugar to prove that clustering can improve the tagging experience. In another direction of work, Gemmell et al. [12] proposed a method to personalize a user's experience within a folksonomy using clustering. They examined unsupervised clustering methods for extracting commonalities between tags, and use the discovered clusters as intermediaries between a user's profile and resources in order to connect the result of search to the user's interests.

Yuruk et al. [22], in a relevant work, proposed a divisive hierarchical clustering algorithm called DHSCAN, that iteratively removes links based on an ascending order of a structural similarity measure. We did divisive hierarchical clustering in a part of our work as well, but the difference is that we applied betweenness method in order to get more evenly distributed clusters. Newman and Girvan [20], in a similar work to this research, studied betweenness algorithm for divisive hierarchal clustering model, and introduced a measure for evaluating the strength of the communities, called modularity. They applied the algorithm to real networks with known community structure and extracted almost the same structure without much difficulty. Newman [20], however, in a subsequent work proposed a fast algorithm for detecting community structures in networks that was actually an enhancement to the previous algorithm of Newman and Girvan [21] and has a considerable advantageous speed over the earlier algorithm. In our work, we took advantage of modularity formula in both the agglomerative and the divisive clustering algorithms. Our betweenness divisive algorithm is in some manner the implementation of [21] with some distinction for weighted graphs which in practice improved the modularity measured compared to the Newman and Girvan algorithm.

Another similar research is directed by Simpson [16]. He applied two clustering algorithms, 'tag-co-occurrence divisive' and 'betweenness-divisive' algorithm to two different data sets, and examined the effectiveness and robustness of both algorithms to the different types of data. This research and the similar ones [21], disclosed the poor performance of the betweenness-divisive algorithm for large data sets, as we also confirmed in our experiments (Section 4.1). There has always been a tradeoff between performance and accuracy of an algorithm; this is also true for these two mentioned algorithms. The efficacy and utility of this algorithm is shown in [21] and some other authors [8], [9], [10] utilized this method for their networks as well. Nevertheless, we intend to do more experiments on the accuracy of betweenness-divisive algorithm in future works, as it is rather a remarkable theory.

3 TCV Framework

The proposed TCV framework consists of four main modules: *data extraction*, *clustering*, *visualization*, and *statistics*.

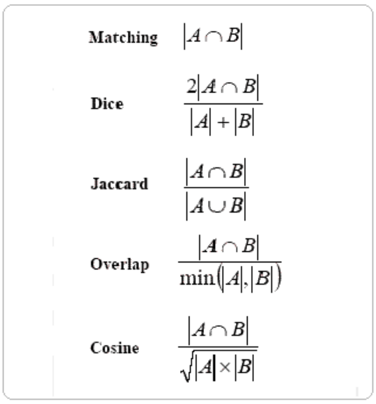
3.1 Data Extraction

One of the early steps in the process of tag analysis is extracting data from web pages. A number of articles [11], [4], [5], [6] studied record extraction from web pages through identifying a set of segments, each of which represented some data, for example a list of objects. They introduced some methods for this aim, but most of them failed to handle complicated or noisy web page structures. In TCV, we utilized the same concept of data extraction from the HTML tags, but in our specific manner and especially for del.icio.us web pages. We made use of a Java library called *htmlparser* [3] to parse html tags either in a linear or nested fashion and preserve data in our data structures for later use.

Correlation Graphs. The extracted data are mainly preserved in three correlation graphs or matrices[1] which will be explained in the following paragraphs. To construct these matrices efficiently, we made use of the idea of bucket sort algorithm [19]. As bucket sort does not make use of comparison sort methods, its computational complexity is linear and therefore applicable for a large number of data entries.

[1] In our design we work on matrices first and then visualize the results in graphs. So the matrices and graphs are applied for the same concept and can be used interchangeably.

User Correlation Matrix. In this matrix the number of matching bookmarks of users-pairs has been counted and reflected.
Bookmark Similarity Matrix. This matrix keeps the compared results of tag sets of bookmark-pairs and calculates the similarity between them. There exists numerous metrics for calculating similarity between two vectors and some of them are listed in Fig. 1.



Matching	$\lvert A \cap B \rvert$
Dice	$\frac{2\lvert A \cap B \rvert}{\lvert A \rvert + \lvert B \rvert}$
Jaccard	$\frac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$
Overlap	$\frac{\lvert A \cap B \rvert}{\min(\lvert A \rvert, \lvert B \rvert)}$
Cosine	$\frac{\lvert A \cap B \rvert}{\sqrt{\lvert A \rvert \times \lvert B \rvert}}$

Fig. 1. A set of common similarity measures

We made use of cosine similarity measurement regarding what had been studied in [1] by Xu et al. on accuracy and performance of the above measures.
User Similarity Matrix. This matrix which needs the prepared data in both previous matrices creates user similarities by analyzing the tagging habits of the users. To do this, we need the knowledge of bookmarks' correlation of user-pairs (user correlation matrix) and the measure of similarity of bookmark-pairs (bookmark similarity matrix).

3.2 Clustering

Greedy Agglomerative Hierarchical Clustering. Agglomerative clustering is a bottom-up clustering approach where each cluster is a subset of a bigger cluster and this trend continues until all the nodes located in at least one cluster. The classic example of agglomerative clustering is species taxonomy.

In the agglomerative clustering approach in TCV, we applied greedy algorithm based on the modularity concepts addressed in [20]. Modularity is a measurement to determine the strength of division of the network to densely connected components with sparse connections between components.

Let e_{ij} be the fraction of edges that connect the vertices in group *i* to vertices in group *j*. e_{ij} which defines the edges end in group *j* are actually equal to e_{ji} because of the symmetric characteristics of networks in undirected graphs in this study. e_{ii} therefore is the fraction of edges in a group connect all the nodes inside the cluster *i*. To get dense components with sparse connections, increasing the sum of edges in a cluster $\sum_{i} e_{ii}$ could be an appropriate step to gain higher modularity. However this sum is not precise enough as it could reach to its maximum value when no clustering is done and all the nodes are in the same cluster. Newman suggests calculating the sum $\sum_{i} e_{ii}$ and subtracting the value that it would take if edged where placed randomly. Here another term is introduced called a_i that is the fraction of edges connected to vertices of group *i*.

$$a_i = \sum_j e_{ij} \tag{1}$$

Therefore the modularity formula is represented in the following format.

$$Q = \sum_i (e_{ii} - a_i^2) \tag{2}$$

In the greedy approach for agglomerative clustering, in each step we select two clusters which merging them maximize the modularity value.

$$\Delta Q = Q_i - Q_j = (e_{ij} - a_i a_j) - (e_{ji} - a_j a_i) = 2(e_{ij} - a_i a_j) \tag{3}$$

And the pseudo code of the greedy agglomerative algorithm is given in the following piece of code.

```
ALGORITHM AGGLOMERATIVE(G=<V,E>)
//there are n clusters each contains one node
s:=0;
while (s<STEPS) do{
      ClaculateModularityofClusters();    //Eq.2
      $q_1$,$q_2$:=GetTwoBestClusterstoMerge();
       // Merging $q_1$ and $q_2$ gives the highest modularity
       based on Eq.3
      MergeClusters($q_1$,$q_2$);
       // $q_1$,$q_2$ makes one new cluster
      s:= s+1; }
```

Betweenness Divisive Hierarchical Clustering. Hierarchical divisive is a top-down method of clustering which generates clusters by sub-dividing the single cluster containing the entire network at first. At each step the least similar vertices or the most distant ones are selected and the connecting edge is removed until no more edge is remained.

Instead of cutting the least similar edges in the correlation graphs, we intend to remove the edges with the highest betweenness, as it has occasionally studied in the past and it brings well spread clusters based on the modularity definition (Eq. 2). Betweenness is a measure that disfavor edges inside communities and get points to those that exist between communities [21]. The edges with the highest betweenness are allocated to connect many nodes together and therefore many clusters, so by removing them the inherent clusters will appear step by step. The betweenness divisive method consists of two main processes: vertex value and betweenness value calculation.

Calculate Vertex Values.

1. The initial vertex *s* is given, distance d_s*=0* and weight w_s*=1.*
- *d* is the distance from the source node to this node in breadth-first search (BFS).
- *w* is the weight of each node shows how many distinct paths there are from the source node to this node.
2. Every vertex *i* adjacent to *s* is given, distance $d_i = d_s + 1 = 1$ and $w_i = w_s = 1$.
3. For each vertex *j* adjacent to vertices *i* we select one of these cases: a) if *j* has not been assigned values for *d* and *w*, *dj=di +1* and *wj=wi*. b) if *j* has already been assigned and *dj=*

di+1 then *wj=wi+wj.* c) if *j* has already been assigned and *dj<di+1* the weight is remained unchanged.

- In the b branch a vertex is being considered more than once, meaning that there are some ways from source to this node, so adding the weight value ensures that we consider all the distinct paths in our calculation.

4. Repeat from step 3 until all the vertices have been assigned a distance and weight value.

Calculate Betweenness Values.

5. Find every leaf vertex *t*. a leaf vertex is a vertex that no vertex should pass through it in its path from *s*.

- To distinguish the leaf vertex we should find vertices for which there is no neighbor with the higher *d* compare to *d* of the leaf vertex.
  $d_t < d_i \quad i \in \{neighbors\}$

6. For each vertex *i* neighboring *t* assign the value to the edge from *t* to *i* of $\frac{w_i}{w_t}$.

7. Now we go through other vertices *j* which all their below neighbors edges are already calculated. By below neighbors we mean neighbors vertices with lower *d*. the edge from vertex *i* to *j*, with *j* being farther from *s* than *i*, assign the value that is 1 plus the sum of the values on the neighboring edges immediately below it all multiplied by $\frac{w_i}{w_j}$.

- Here is the modification point in the main algorithm: instead of adding just 1 to the above sum, we add1 plus the weight of the edge showing that some edges are more likely to be between other nodes according to their weights.

8. Repeat the step 7 until node *s* is reached.

When the calculation for a single source is finished, the process will be start over until all nodes become source once; we then have the betweenness values for each edge which is the sum of scores for the different source nodes.

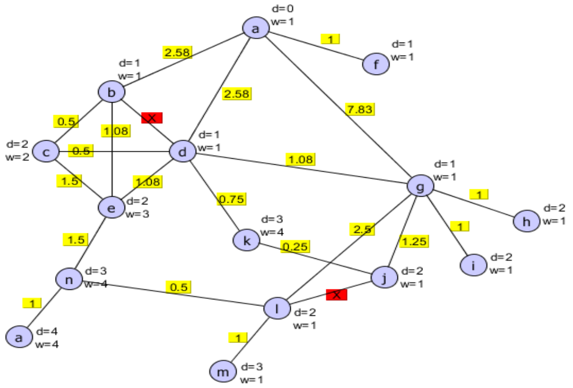


Fig. 2. Calculate edges values when ‘a’ is the source node

3.3 Visualization

For visualizing clusters before and after clustering, we make use of *prefuse* visualizing toolkit [17], which is a software framework for creating dynamic visualizations for both structured and unstructured data [7]. We applied various layouts of *prefuse* to demonstrate our original graphs and the communities formed after the clustering process. In Fig. 3 the process of visualizing in the TCV framework is demonstrated.

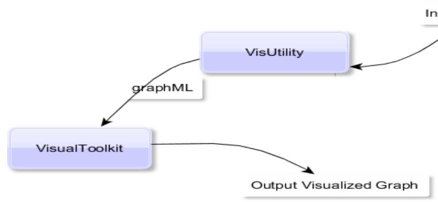
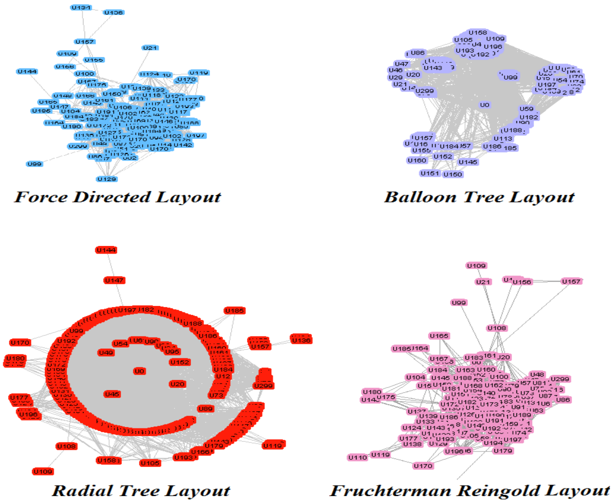


Fig. 3. How visual toolkit works in TCV framework

The input graph which is already saved in a text file is given to VisUtility class that creates graphML [29] out of the input graph. VisualToolkit class is able to create different layouts of the output graph when it is supplied with the proper graphML. Some samples of the Bookmark Similarity Graph represented in different layouts are demonstrated in Fig. 4.



Force Directed Layout

Balloon Tree Layout

Radial Tree Layout

Fruchterman Reingold Layout

Fig. 4. Different layouts of *prefuse* for bookmark similarity matrix

3.4 Statistics

In the statistic part of the TCV framework, performance of the clustering methods can be compared and demonstrated in different excel charts. The clustering algorithm and the matrix are selected and the chosen type of the chart will be drawn in this step. With the help of a Java chart library, *jfreechart* [18], various excel charts can be drawn to have different looks of the clustering process for all three matrices.

4 Experiments and Results

This section presents some experiments conducted to consider different aspects of a tagging system which leads to some series of results and conclusions. For several experiments we use two sets of data from del.icio.us, a sample of 20 users from the *popular* bookmarks and another sample of 50 users from the *recent* page. Table 1 shows the characteristics of two samples of the data set.

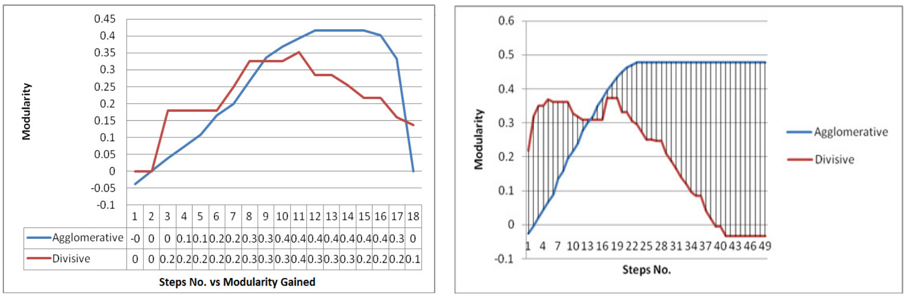
Table 1. Statistics of two samples of data sets

	User Source	***Users No.***	***Bookmarks per User***	***Total Bookmark No.***	***Tag No.***
Sample 1	popular	20	<=300	1714	1603
Sample 2	recent	50	<=300	3992	3832

4.1 Agglomerative vs. Divisive Approach

In this section we do a comparative study on agglomerative and divisive clustering methods regarding the modularity and timing issues.

Modularity.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Agglomerative	-0	0	0	0.1	0.1	0.2	0.2	0.3	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.3	0
Divisive	0	0	0.2	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.4	0.3	0.3	0.3	0.2	0.2	0.2	0.1

(a) Sample 1

(b) Sample 2

Fig. 5. Modularity vs. steps No. for agglomerative and divisive method

In Fig. 5 the horizontal axis is the number of steps and the vertical axis is the modularity measured in each step. As the experiment in sample 1 shows, the maximum value for modularity is achieved in the agglomerative algorithm (about 0.42). Although modularity value closer to 1.0 is an indicator for a better clustering, in real networks, however the highest reported value is 0.75. In practice, Newman [20] found that modularity values around 0.3 indicate a strong community structure for the given network. Here the agglomerative method achieves about 7% higher modularity rather than the divisive method.

In sample 2 diagram, the superiority of agglomerative method to the divisive algorithm is even more observable. The maximum modularity value in the divisive approach is 0.37 while it is 0.47 for the agglomerative approach. (about 10% improvement for agglomerative)

Timing. Although the increasing speed of computers will probably alleviate time limits in the coming years, the time complexity is still an important issue in most of the algorithms. Fig. 6 shows a comparison between running time of agglomerative and divisive algorithms in the user correlation matrix of 50 users (Sample 2).

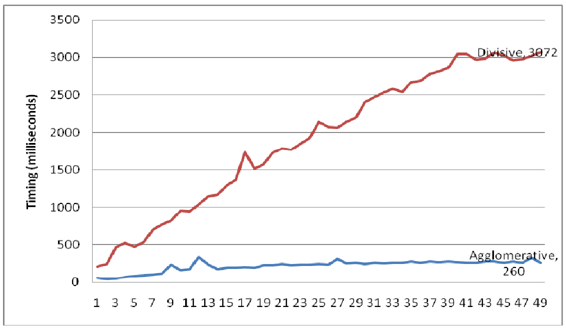
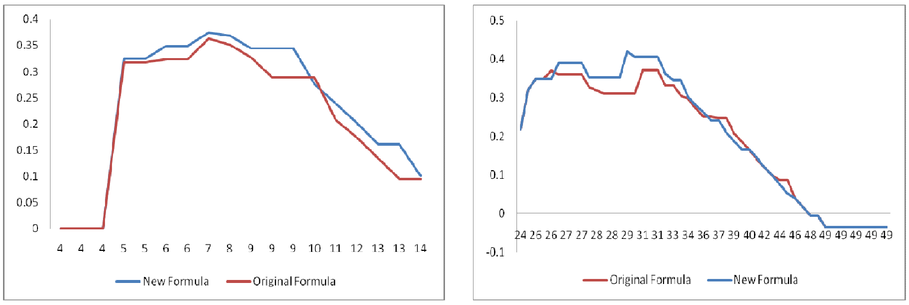


Fig. 6. Measured time in agglomerative and divisive algorithms

The above graph shows that there is a steady upward trend for agglomerative time consumption, while the divisive trend has a sharp increase in the amount time needed to cluster the same sets of objects. As it is also clarified in [21], the divisive clustering method is entirely computational intensive and it operates in $O(n^3)$ time on sparse graphs. Therefore, for larger graphs with more than 10000 nodes this algorithm is rather inapplicable.

4.2 Betweenness Divisive Method



(a) Sample 1

(b) Sample 2

Fig. 7. Modularity vs. number of clusters for the original formula and the new formula

The new formula which has addressed in section 3.2, step 8 of the betweenness algorithm, deals with the weights of the edges in our similarity graphs. The changes made in this formula in a successful manner increased the overall modularity. As Fig. 7 shows, the new formula obtained better results rather than the original one. For data set of sample 1, the modularity value improved by almost 1%, and as the diagram of sample 2 shows, the maximum modularity gained by the original formula, for un-weighted graphs, is almost 0.37 while it reached a peak at 0.42 for our weighted graph (about 5% improvement).

5 Conclusion and Future Works

In this work, we proposed a framework for clustering and visualizing the data of a well-known bookmarking system, del.icio.us. The main goal was to improve the data exploration and data navigation in collaborative tagging systems. Two hierarchical clustering model has been analysed and implemented, greedy agglomerative and betweenness divisive. We also made some modifications to the main betweenness divisive algorithm and proposed a new formula to improve the accuracy of the mentioned algorithm. However, according to our experiments in section 4.1 and 4.2, we observed higher accuracy and superior performance in agglomerative method rather than divisive approach. In fact betweenness divisive results proved its inferiority in computational timing and even modularity gained, and therefore it tends to be less widely used rather than the agglomerative methods based on the acquired results.

One of the open issues directly related to our framework is developing a search engine in bookmarking environments and applying clustering to it in order to show improvement in search result over clustered entities. Another further work regarding the divisive clustering algorithms is to improve its speed. Some solutions such as parallelization were proposed in [21], but the issue is still remained open. Another work could be a comparative study of traditional divisive method and the betweenness divisive algorithm introduced in [21]. Although in [21] the concept of betweenness was addressed as a superior approach, we couldn't find any research specifically dealing with the superiority of this approach over the traditional one.

Acknowledgement. I would like to thank BIHE[2] for giving me the opportunity to pursue my studies and researches, while it is not possible in Iran for Baha´i[3] scholars. And special thanks to Mr. Farid who is now in prison for the sake of educating Baha´í students! Many thanks for all his supports and devotions. I must thank my supervisors Dr. Hercock and Dr. Duman for great guidance and advice and my family for all their supports and sacrifices.

References

1. Xu, K., Chen, Y., Jiang, Y., Tang, R., Liu, Y. and Gong, J.: A Comparative Study of Correlation Measurements for Searching Similar Tags. In: ADMA '08 Proceedings of the 4th international conference on Advanced Data Mining and Applications, Heidelberg, Berlin, Springer-Verlag, pp. 709-716 (2008)
2. Mathes, A.: Folksonomies – Cooperative Classification and Communication Trough Shared Metadata, http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html (2004)
3. http://htmlparser.sourceforge.net/
4. Hiremath, P. S. and Algur, S. P.: Extraction of Flat and Nested Data Records from Web Pages. In: International Journal on Computer Science and Engineering, 2(1), pp. 36-45 (2010)
5. Alvarez, M., Pan, A., Raposo, J., Bellas, F. and Cacheda, F.: Fining and Extracting Data Records from Web Pages. In: Journal of Signal Processing Systems, 59(1), pp. 123-137 (2010)

[2] http://www.bihe.org/

[3] http://www.bahai.org/

6. Hiremath, P. S. and Algur, S. P.: Extraction of Data from Web Pages: A Vision Based Approach. In: International Journal of Computing and Information Science and Engineering, 3(1), pp. 214-324 (2009)
7. Heer, J., Card, S.K. and Landay, J.A.: Prefuse: a Toolkit for Interactive Information Visualization. In: CHI '05 Proceedings of the SIGCHI conference on Human factors in computing systems, New York, USA, ACM Press, pp. 421-430 (2005)
8. Holme P., Huss M**.** and Jeong, H.: Subnetwork hierarchies of biochemical pathways. Bioinformatics, 19, pp. 532-538 (2003)
9. D. Wilkinson and B. A. Huberman, e-print cond-mat/0210147.
10. P. Gleiser and L. Danon, e-print cond-mat/0307434.
11. Miao, G., Tatemura,J., Hsiung, W.,P., Sawires, A. and Moser, L.,E.: Extracting Data Records from the Web Using Tag Path Clustering. In: WWW '09 Proceedings of the 18th international conference on World Wide Web, New York, USA, ACM Press, pp. 981-990 (2009)
12. Gemmell, J., Shepitsen, A., Mobasher, B. and Burke, R.: Personalization in Folksonomies Based on Tag Clustering. In: Proceedings of the 6th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems, Chicago, USA, AAAI Press (2008)
13. Sbodio, M.L. and Simpson, E.: Tag Clustering with Self Organizing Maps. Hewlett-Packard Development Company, L.P (2009)
14. Begelman, G., Keller, P., Smadja, F.: Automated Tag Clustering: Improving Search and Exploration in the Tag Space, http://www.pui.ch/phred/automated_tag_clustering (2006)
15. Schrammel, J., Leitner, M. and Tscheligi, M.: Semantically Structured Tag Clouds: An Empirical Evaluation of Clustered Presentation Approache. In: CHI '09 Proceedings of the 27th international conference on Human factors in computing systems, New York, USA, ACM Press, pp. 2037-2040 (2009)
16. Simpson, E.: Clustering Tags in Enterprise and Web Folksonomies. Second International Conference on Weblogs and Social Media*,* AAAI Press (2007)
17. http://www.prefuse.org/
18. www.jfree.org/jfreechart/
19. http://en.wikipedia.org/wiki/Bucket_sort
20. Newman, M.E.J.: Fast Algorithm for Detecting Community Structure in Networks. PHYSICAL REVIEW E 69 (2004)
21. Newman, M.E.J. and Girvan, M.: Finding and Evaluating Community Structures in Networks. *Physical Review E*, 69(2) (2004)
22. Yuruk, N., Mete, M., Xu, X. and Schweiger, T.A.J.: A Divisive Hierarchical Structural Clustering Algorithm for Networks. In: ICDMW '07 Proceedings of the Seventh IEEE International Conference on Data Mining Workshops, Washington, DC, USA, IEEE Computer Society , pp. 441-448 (2007)
23. http://en.wikipedia.org/wiki/GraphML
24. http://www.pui.ch/phred/automated_tag_clustering/

Session V

Intelligent Systems

Chairman: Mário Antunes, Alina Trifan

Mining Association Rules for Ordinal Data Classification using an Unimodal Model
Cláudio Rebelo de Sá, Carlos Soares, Joaquim Pinto Da Costa, Alípio Jorge and Paulo Azevedo

Network Traffic Classification under Time-Frequency Distribution
André Riboira and Angelos K. Marnerides

Knowledge Extraction From Social Networks
Nuno Silva

Spatio-temporal clustering methods classification
Hadi Fanaee Tork

Relevance Ranking for Predicting Web Search Results
Pedro M. Teixeira

M6: a method for compressing complete genomes using Markov models
Diogo Pratas and Armando J. Pinho

User Clustering in Smartphone Applications
Klaus Schaefers

Mining Association Rules for Ordinal Data Classification using an Unimodal Model

Cláudio Rebelo de Sá[1], Joaquim Costa[4], Carlos Soares[1,2], Paulo Azevedo[5], and Alípio Mário Jorge[1,3]

[1] INESC TEC Porto, Porto, Portugal
[2] Faculdade de Economia, Universidade do Porto
[3] DCC - Faculdade de Ciencias, Universidade do Porto
[4] DM - Faculdade de Ciencias, Universidade do Porto
[5] CCTC, Departamento de Informática, Universidade do Minho
claudio.r.sa@inescporto.pt, jpcosta@fc.up.pt, csoares@fep.up.pt, pja@uminho.pt, amjorge@fc.up.pt

Abstract. Some real life problems require the classification of items into naturally ordered classes. Conventional methods, intended for the classification of nominal classes, are traditionally used to deal with these problems where the classes are ordered. This paper proposes an adaptation of association rules for classification intended for multi-class problems where the order relation is not ignored. The theoretical background assumes that the random variable class associated with a given query should follow a unimodal distribution. The adaptation, which uses class association rules (CAR's), is essentially in terms of the output handling, i.e the voting system for the predicted class. The experiments in real datasets are presented. Despite this very simple variant of association rules for classification, the results indicate that the method is making valid predictions and is competitive with state-of-the-art algorithms.

1 Introduction

Many classification problems require classifying examples into naturally ordered classes. These problems are commonly found in many study fields, such as econometric modeling and collaborative filtering. Other applications include biomedical classification problems, in which is very frequent that the classes are ordered, although that is almost never taken into account and the conventional methods, for nominal classes or regression, are used.

Conventional methods of supervised classification can be used, but first of all it is usually harder and slower to train with these methods and secondly the derived classifier might not be really appropriate. On the other hand, using regression methods introduces an arbitrary selection of numbers to represent the classes, which in turn influence both the prediction function and the usual measures of performance assessment that are used. However, the use of methods specifically designed for ordered classes results in simpler classifiers, making it easier to interpret the factors that are being used to discriminate among classes [9].

Association rules mining is a very important and successful task in data mining. Although its original purpose was only descriptive, several adaptations have been proposed for predictive problems like in [8]. This work proposes an adaptation of association rules for classification intended for multi-class problems where the order relation of the classes is not ignored. The method searches for Class Association Rules (CAR's) and handles them taking into account the order relation of the classes. This is done by forcing an unimodal distribution [5] of the class probabilities, even if their empirical distribution is not unimodal.

The paper is organized as follows: sections 2 and 3 introduce the unimodal paradigm and the task of association rule mining, respectively; section 4 gives a simple description of the method proposed here; section 6 presents the experimental setup and discusses the results; finally, section 7 concludes this paper.

2 The unimodal paradigm

Let us define a supervised classification problem with K ordered classes $\mathcal{C}_1 < \ldots < \mathcal{C}_K$ and denote the feature space as $\mathbb{X}$. In common supervised classification problems, the goal is to find a mapping:

$$f_T : \mathbb{X} \rightarrow \{\mathcal{C}_i\}_{i=1}^{K}$$

that minimizes certain cost functional relative to the ℓ examples in a given training set $T = \{(\mathbf{x}_i, \mathcal{C}_{\mathbf{X}_i})\}_{i=1}^{\ell} \subset \mathbb{X} \times \{\mathcal{C}_i\}_{i=1}^{K}$. Bayes decision theory aims to maximize the *a posteriori* probability $P(\mathcal{C}_k|\mathbf{x})$ in the classification of new examples $\mathbf{x}$ with the class $\mathcal{C}_k$. To that end, we must find a function which estimates the *a posteriori* probabilities:

$$f_T(\mathbf{x}) = \arg\max_{\mathcal{C}_k}\{P(\mathcal{C}_k|\mathbf{x})\}$$

However, should we consider this very same foreground in every classification problem? Let us assume we wave a temperature classification problem with $K = 5$ classes: {Very cold, Cold, Mild, Hot, Very hot}. It is intuitive to consider a natural ordering between these classes: Very cold < Cold < Mild < Hot < Very hot. Thus if the model obtains $P(C_4|\mathbf{x})$ as the highest *a posteriori* probability for a given query point $\mathbf{x}$, the second most probable class should be $P(C_3|\mathbf{x})$ or $P(C_5|\mathbf{x})$. This means that if the most likely is a Hot day, then the second most likely should either be a Mild day or a Very hot day. In other words, we can assume that the probabilities should decrease monotonically to the left and to the right of the class with maximum probability. By using classifiers which do not take into account the order relation presented, the second highest *a posteriori* probability can be, for instance, $P(\mathcal{C}_1|\mathbf{x})$, which makes no sense.

More formally, the unimodal paradigm introduced in [5] assumes that in a supervised classification problem with ordered classes, the random variable class $\mathcal{C}_{\mathbf{X}}$ associated with a given query point $\mathbf{x}$ should follow a unimodal distribution. We assume a particular unimodal discrete distribution for $\mathcal{C}_{\mathbf{X}}$, and a classifier f_T estimates the *a posteriori* probabilities by estimating the parameters of the assumed distribution.

3 Association Rules Mining

An association rule (AR) is an implication: $A \to C$ where $A \bigcap C = \emptyset$, $A, C \subseteq desc(\mathbb{X})$ where $desc(\mathbb{X})$ is the set of descriptors of instances in $\mathbb{X}$, typically pairs $\langle attribute, value \rangle$. We also denote $desc(x_i)$ as the set of descriptors of instance x_i.

Association rules are typically characterized by two measures, support and confidence. The support of rule $A \to C$ in T is *sup* if *sup*% of the cases in it contain A and C. Additionally, it has a confidence *conf* in T if *conf*% of cases in T that contain A also contain C.

The original method for induction of AR is the APRIORI algorithm that was proposed in 1994 [1]. APRIORI identifies all AR that have a support and confidence higher than a given minimal support threshold (*minsup*) and a minimal confidence threshold (*minconf*), respectively. Thus, the model generated is a set of AR of the form $A \to C$, where $A, C \subseteq desc(\mathbb{X})$, and $sup(A \to C) \geq minsup$ and $conf(A \to C) \geq minconf$. For a more detailed description see [1].

Despite the usefulness and simplicity of APRIORI, it runs a time consuming candidate generation process and needs space and memory that is proportional to the number of possible combinations in the database. Additionally it needs multiple scans of the database and typically generates a very large number of rules. Because of this, many new pruning methods were proposed in order to avoid that. Such as the hashing [10], dynamic itemset counting [4], parallel and distributed mining [11], relational database systems integrated with mining [12].

Association rules were originally proposed for descriptive purposes. However, they have been adapted for predictive tasks such as classification (e.g., [8]) which is described in Section 3.2.

3.1 Pruning

AR algorithms typically generate a large number of rules (possibly tens of thousands), some of which represent only small variations from others. This is known as the rule explosion problem [3]. It is due to the fact that the algorithm might find rules for which the confidence can be marginally improved by adding further conditions to the antecedent.

Pruning methods are usually employed to reduce the amount of rules, without reducing the quality of the model. A common pruning method is based on the improvement that a refined rule yields in comparison to the original one [3]. The *improvement* of a rule is defined as the smallest difference between the confidence of a rule and the confidence of all sub-rules sharing the same consequent. More formally, for a rule $A \to C$

$$imp(A \to C) = min\left(\forall A' \subset A, conf\left(A \to C\right) - conf\left(A' \to C\right)\right)$$

As an example, if one defines $minImp = 0.1\%$, the rule $A_1 \to C$ will be kept, if, and only if $conf\left(A_1 \to C\right) - conf\left(A \to C\right) \geq 0.001$, where $A \subset A_1$.

3.2 Class Association Rules

Classification Association Rules (CAR), were proposed as part of the Classification Based on AR (CBA) algorithm [8]. A class association rule (CAR) is an implication of the form: $A \rightarrow \mathcal{C}$ where $A \subseteq desc(\mathbb{X})$, and $\mathcal{C} \in \mathcal{L}$, which is the class label. A rule $A \rightarrow \mathcal{C}$ holds in T with confidence *conf* if *conf*% of cases in T that contain A are labeled with class $\mathcal{C}$, and with support *sup* in T if *sup*% of the cases in it contain A and are labeled with class $\mathcal{C}$.

CBA takes a tabular data set $T = \{\langle x_i, \mathcal{C}_i\rangle\}$, where x_i is a set of items and $\mathcal{C}_i$ the corresponding class, and look for all frequent *ruleitems* of the form $\langle A, \mathcal{C}\rangle$, where A is a set of items and $\mathcal{C} \in \mathcal{L}$. The algorithm aims to choose a set of high accuracy rules $\mathcal{R}_\mathcal{C}$ to match T. $R_\mathcal{C}$ matches an instance $< x_i, \mathcal{C}_i > \in T$ if there is at least one rule $A \rightarrow \mathcal{C} \in \mathcal{R}_\mathcal{C}$, with $A \subseteq desc(x_i), x_i \in \mathbb{X}$, and $\mathcal{C} \in \mathcal{L}$. If the available rules cannot classify a new example, a default class is given to it (e.g., the majority class in the training data).

4 Association Rules for unimodal class distribution

Although the original purpose of Association rules was only descriptive, several adaptations have been proposed for predictive problems like in [8]. The objective of this work is to propose a supervised classification method, built in association rules, more suitable for predicting ordinal classes. We believe that our contribution will improve the performance of the supervised classification in the presence of ordinal classes in the association rules domain.

After generating the CARs, given some user defined interest measures, these are grouped and counted per class for each unclassified example x. This is, considering the classes {Very cold, Cold, Mild, Hot, Very hot} the method can obtain a bunch of rules for an unclassified example, like, for instance:

(Very cold-5 rules, Cold-7 rules, Mild-15 rules, Hot-2 rules, Very hot-3 rules)

Which is an empirical distribution of the classes where each CAR contributes in equal terms for the counting. These numbers are then used to estimate the probabilities of each of the five classes $P(C_i|x)$. Now, the unimodal paradigm enters: instead of using the usual estimator for these probabilities, for instance $\hat{P}(\text{Verycold}|x) = 5/38$, we will use a different estimator that smooths these numbers and at the same time verify the unimodal paradigm described above; this paradigm is what makes sense, in our opinion, with ordered classes. To do so, we will use a specific unimodal parametric statistical model to be fitted to the numbers (5,13,15,2,3): the binomial distribution. Our purpose will then be to find a binomial distribution $B(4, p)$ whose 5 probabilities best approximate the empirical probabilities (5/38, 13/38, 15/38, 2/38, 3/38). In order to do so, the parameter p will be estimated by maximum likelihood [7].

This p is then used to calculate the probability for the set of classes. The class with higher probability will be the one chose for the recommendation. The method, as we propose, is independent of the CARs generator, once only the recommendation process is affected.

We believe that this method contributes for better performance because it will only force the unimodality in the presence of non unimodal distributions. In those cases where the empirical distribution is already unimodal the probabilities will be slightly smoothed only.

5 Error measures

In this supervised classification problem with ordered classes, we measure the performance of the classifier using two different measures. The Misclassification Error Rate (MER) and the Mean Absolute Error (MAE). The first one considers every misclassification equally costly. In the latter the performance of a classifier f_T is assessed in a dataset $\mathcal{O} \subset \mathcal{X}$ through

$$\mathrm{MAE} \ = \ \frac{1}{\mathrm{card}(\mathcal{O})} \sum_{\mathbf{x} \in \mathcal{O}} |g(\mathcal{C}_{\mathbf{x}}) - g(f_T(\mathbf{x}))|,$$

where $g(\cdot)$ corresponds to the number assigned to a class. Assuming this assignment is arbitrary the performance measurement given by MAE can be inadequate. However, as we are dealing with ordinal classes in this work, the assignment will take into account the classes natural order. This measure can be more informative than MER, because its values increase with the absolute differences between the “true” and “predicted” class numbers and so the misclassifications are measured as a distance factor.

6 Experimental Results

The datasets in this work (Table 1) were taken from:

- (A) www.gatsby.ucl.ac.uk/ chuwei/ordinalregression.html
- (B) www.cs.waikato.ac.nz/ ml/index.html

All the variables were discretized by the following order: (1) *recursive minimum entropy partitioning* criterion ([6]) with the *minimum description length* (MDL) as stopping rule (2) *equal width* bins. The division into *equal width* bins will only be used in the attributes left undiscretized by the previous method.

The evaluation measures used are the Misclassification Error Rate (MER) and Mean Absolute Error (MAE) and the performance of the method was estimated using ten-fold cross-validation. The performance of the Unimodal method was compared with a very simple CAR method, which classifies the examples with the most frequent class from the set of all the matching CARs. For the generation of frequent items we used CAREN [2] and *minsup* was fixed in 0.1% and *mimp* was set to 0.001%.

The number of each class per dataset is graphically presented in Fig. 1. Some datasets do not seem to have an unimodal distribution in the classes. This can be because their distribution is not binomial at all or due to the fact that the data has unbalanced information.

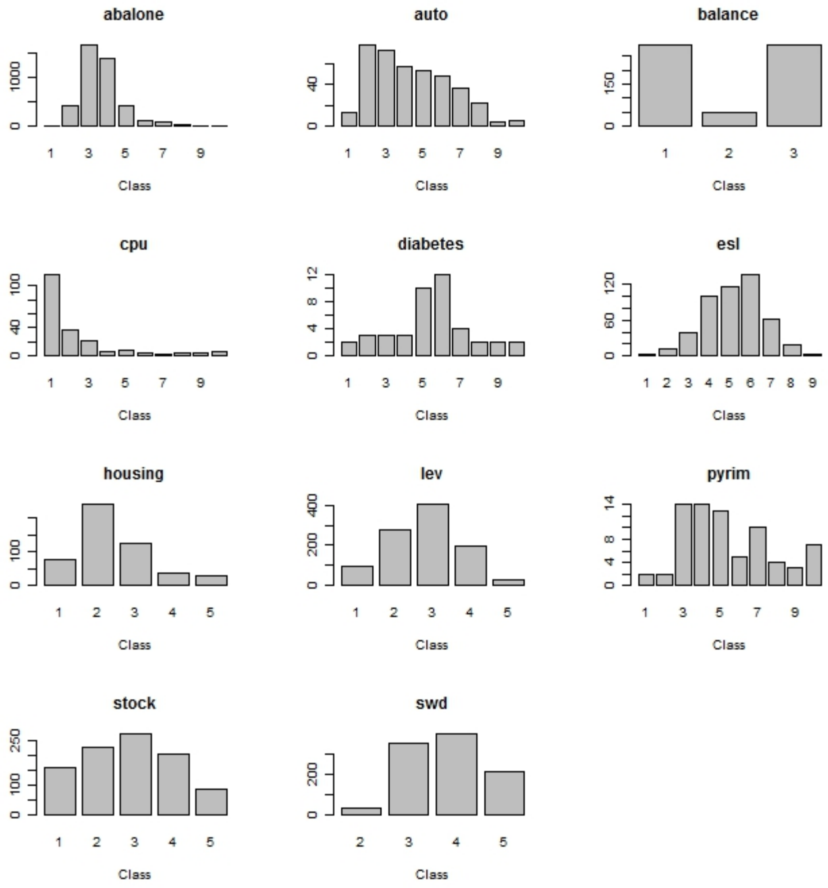


Fig. 1. Distribution of the classes by dataset

Table 1. Summary of the datasets

Datasets	source	#examples	#attributes	#classes
abalone	A	4177	10	10
auto	A	392	7	10
balance	A	625	4	3
cpu	A	209	6	10
diabetes	A	43	2	10
esl	B	488	4	9
housing	A	506	13	5
lev	B	1000	4	5
pyrim	A	74	27	10
stock	A	950	9	5
swd	B	1000	10	4

Table 2. Results obtained in terms of the Misclassification Error Rate

minconf	75%		50%		25%	
	CARs	*Unim*	*CARs*	*Unim*	*CARs*	*Unim*
abalone	.520	**.518**	.450	**.445**	**.466**	.485
auto	**.719**	.729	.566	**.559**	.528	**.492**
balance	**.248**	.333	**.248**	.333	**.248**	.333
cpu	.455	**.440**	.450	**.436**	.435	**.426**
diabetes	.780	.780	.905	**.880**	**.895**	.935
esl	.416	**.399**	.352	**.334**	.348	**.332**
housing	**.296**	.306	**.334**	.352	.391	**.389**
lev	.569	**.569**	.503	**.483**	.484	**.478**
pyrim	.755	**.689**	**.737**	.850	.834	**.771**
stock	**.149**	.183	**.161**	.202	**.143**	.225
swd	.521	**.519**	.446	**.441**	**.449**	.462

6.1 Results

The results presented in table 2 indicate that the classifier using the unimodal method obtains slightly better results than the ones without it. This error measure is only an indicator of the performance of a classifier because, as said before, when dealing with ordered classes there are more fair error measures that should be considered to measure the distance of the predicted class from the real class. For this reason Table 3 presents the results obtained with MAE, which penalizes the bigger deviations of the recommended class in comparison with the real ones. In this case, the Unimodal method clearly outperforms the other method in every confidence values considered.

Despite the absence of a statistical test, the improvement observed is a good motivation for a more throughout experimental study in the future. This indicates that this association rules method is in fact improving its performance by

Table 3. Results obtained in terms of the Mean Absolute Error

minconf	75%		50%		25%	
	CARs	*Unim*	*CARs*	*Unim*	*CARs*	*Unim*
abalone	.758	**.757**	**.599**	.601	**.605**	.626
auto	**1.941**	1.951	1.168	**1.160**	.694	**.645**
balance	.418	**.345**	.418	**.345**	.418	**.345**
cpu	1.231	**1.212**	1.177	**1.163**	.862	**.780**
diabetes	1.495	1.495	2.030	**1.875**	2.090	**1.940**
esl	.500	**.477**	.381	**.355**	.379	**.355**
housing	.332	**.330**	.405	**.397**	.462	**.435**
lev	.669	.669	.554	**.536**	.535	**.523**
pyrim	1.663	**1.264**	1.623	**1.479**	1.702	**1.454**
stock	**.151**	.184	**.164**	.203	**.145**	.225
swd	.638	**.634**	.481	**.479**	**.491**	.505

using the Unimodal method when the class distribution is ordered. This also means that, despite the simplicity of the adaptation, this can be considered a competitive method. We expect that the results can be significantly improved, for instance, by implementing more suitable pruning methods.

7 Conclusions

In this paper we present a simple adaptation of association rules for ordinal classification. This adaptation essentially consists of adapting a well known method taking into consideration the properties of the classes of the datasets. Which, in this particular case, is the natural orders of the classes.

These results are positive and clearly show that this method, in most cases, decreases the error rate when compared with normal association rules methods in this scenario. We think that it executes well its function to better classify classes with a binomial distribution.

This work uncovered several problems that could be better studied in order to study and improve the method's performance. They include: improving the prediction generation method; implementing better pruning methods; choice of parameters.

We also want to undertake statistical tests, in future work, to give a stronger analysis of the performance of the method.

Acknowledgments

This work was partially supported by the European Regional Development Fund through the programme COMPETE (PEst-C/MAT/UI0144/2011) from FCT and Palco AdI project Palco3.0 financed by QREN and Fundo Europeu de Desenvolvimento Regional (FEDER). We thank the anonymous referees for useful comments.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proc. 20th Int. Conf. Very Large Data Bases, VLDB. vol. 1215, p. 487499. Citeseer (1994)
2. Azevedo, P.J., Jorge, A.M.: Ensembles of jittered association rule classifiers. Data Min. Knowl. Discov. 21(1), 91–129 (2010)
3. Bayardo, R., Agrawal, R., Gunopulos, D.: Constraint-based rule mining in large, dense databases. Data Mining and Knowledge Discovery 4(2), 217–240 (2000)
4. Brin, S., Motwani, R., Ullman, J.D., Tsur, S.: Dynamic itemset counting and implication rules for market basket data. Proceedings of the 1997 ACM SIGMOD international conference on Management of data - SIGMOD '97 pp. 255–264 (1997), http://portal.acm.org/citation.cfm?doid=253260.253325
5. da Costa, J.F.P., Alonso, H., Cardoso, J.S.: The unimodal model for the classification of ordinal data. Neural Networks 21(1), 78–91 (2008)
6. Fayyad, Irani: Multi-interval discretization of continuous-valued attributes for classification learning. In: International Conference on Machine Learning. pp. 1022–1027 (1993)
7. Jae, I., Myung: Tutorial on maximum likelihood estimation. Journal of Mathematical Psychology 47(1), 90–100 (2003), http://www.sciencedirect.com/science/article/pii/S0022249602000287
8. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. Knowledge Discovery and Data Mining pp. 80–86 (1998)
9. Mathieson, M.: Ordinal models for neural networks (1996)
10. Park, J.S., Chen, M.S., Yu, P.S.: An effective hash-based algorithm for mining association rules. ACM SIGMOD Record 24(2), 175–186 (May 1995), http://portal.acm.org/citation.cfm?doid=568271.223813
11. Park, J., Chen, M., Yu, P.: Efficient parallel data mining for association rules. of the fourth international conference on (1995), http://portal.acm.org/citation.cfm?id=221270.221320
12. Thomas, S., Sarawagi, S.: Mining generalized association rules and sequential patterns using SQL queries. . Conf. on Knowledge Discovery and Data Mining (1998), http://www.aaai.org/Papers/KDD/1998/KDD98-062.pdf

Network Traffic Classification under Time-Frequency Distribution

André Riboira [1] and Angelos K. Marnerides[2]

[1]Dep. of Informatics Engineering, Faculty of Engineering
[2]Dep. of Computer Science, Faculty of Sciences
University of Porto, Portugal
[1]andre.riboira@fe.up.pt, [2]amarnerides@dcc.fc.up.pt

Abstract. The Internet is continuously being expanded. This expansion is represented by an incremental utilization of network traffic. Consequently, such immense amounts of traffic volume require efficient traffic engineering techniques embodied within Quality of Service strategies. Traffic classification constitutes a basis for such techniques and enables the selective prioritization of network flows. Most offline network traffic classification techniques are based on classification models formulated by supervised machine learning (ML) algorithms. The adequate selection of suitable features for further usage in ML-algorithms is fundamental to achieve high classification accuracy rates. This paper expands the Marnerides' previous work with respect to the exploitation of the Renyi information within the traffic classification process. We provide a novel, and detailed comparison with other features and we further illustrate that the Renyi information allows a far more accurate ML-based classification of the network flows. The Renyi information is derived from the smoothed Pseudo Wigner-Ville (SPWV) energy time-frequency (TF) distribution of the traffic signal as composed by the counts of bytes of the network packets that constitutes a network flow. The impact of this new feature was evaluated using packet traces captured at the Keio university campus network in Japan.

Keywords: Network traffic classification, machine learning, decision tree, neural network, time-frequency distribution, Renyi information

1 Introduction

The Internet is a resource that is continuously being expanded in terms of users, devices and applications. The expansion of the Internet is represented by an incremental network traffic volume persona in backbone networks. This high traffic volume utilization requires proper traffic engineering techniques, to guarantee the best usage of the network infrastructure[18]. The diversity of application-layer protocols (e.g. HTTP, FTP) exposes different performance characteristics for a given network flow (e.g. TCP, UDP flow) such as latency, throughput and jitter. Quality of Service (QoS) is a set of traffic engineering techniques that tries to assess and optimize the aforementioned characteristics by defining priority levels based on the criticality of each flow [14]. Accurate traffic classification is a challenging task and is considered as fundamental for supporting selective prioritization of network flows. Two different approaches exist to perform network traffic classification: (1) online traffic classification and (2) offline traffic classification [8]. The first approach aims to perform network traffic classification close to real-time, mostly by using directly the data captured from a network interface. The second approach aims to perform network traffic classification using a set of historical data, captured over a limited period of time in the past. These two different network classification approaches are used under the scope of different purposes. Offline traffic classification is mostly performed using ML techniques [19]. Supervised ML-algorithms tend to create a classification model based on the information obtained through the analysis of a training set. The training set has records with different data fields named features, and is considered as the a priori knowledge for the whole supervised ML technique. A test set has the same features of the training set and the class of all the records is also known in advance, and is used to verify the accuracy of the formulated model. This information is used to compare the class predicted by the model with the actual record class [2]. It is important to explore new features that may increase the accuracy of the ML-algorithms. This paper expands the work done in [17] with respect to the exploitation of the Renyi information within the traffic classification process. We provide a novel, and detailed comparison with other features and we further illustrate that the Renyi information allows a more accurate ML-based classification of the network flows, on certain conditions.

Traditionally traffic classification has been mainly assessed with the usage of the src/dst port numbers that allow the identification of an application layer protocol (eg. HTTP). Although this feature allows a fast classification, some applications do not use well known ports, and this consequently results to lower classification accuracy rates. On top of that, some applications, mainly peer-to-peer (P2P), use ports that are usually assigned to other well-known applications and services, in an attempt to avoid their proper classification. The goal of this behavior is mainly to bypass the proper traffic prioritization, because usually the P2P applications and services have low priority and have their use limited with traffic engineering techniques such as traffic shaping [16]. Another feature used in current and past literature is packet payload information. The analysis of the message cargo allows an accurate classification of each network flow. The main disadvantages of this feature are related with confidentiality issues, and encrypted payloads. Moreover, the payload analysis is based on semantic parsing, which is usually a resource intensive methodology [16]. New features are methodically tested to assess the importance of their use in supervised ML-algorithms. The purpose of this work is to experiment and evaluate the accuracy derived by the usage of two new features, obtained by calculating the Renyi Information [1] of a signal representing the energy-TF distribution [5] of the Smoothed Pseudo-Wigner-Ville (PWV) [3], based on the network features of the byte-size and the inter-arrival interval of the packets for a given flow. In particular, this work depends on three supervised ML-algorithms namely decision trees [20], neural networks [9] and random forests [4]. The algorithmic objective of a decision tree classification scheme is to construct a tree structure where the interior nodes corresponds to features of the data set, and the edges to children nodes corresponds to the possible values for each feature. Leaf nodes represents the target classes. The algorithmic objective of a neural network classification scheme is to create an interconnected group of artificial neurons, and process information using an approach oriented to connections. The random forest algorithm consists of many decision trees and outputs the class that is the mode of the classes output by individual trees.

The remainder of this paper is structured as follows. Section 2 presents the procedure, methods and techniques. The overall methodology was based on data pre-processing, including calculating the new features, and the application of different supervised ML-algorithms to analyze the impact of these new features. The Section 3 presents the results and discussion. The overall results were based in comparing the accuracy of the models formulated by supervised ML-algorithms, and the discussion presents reasonable explanations for the obtained results. Section 4 concludes the paper.

2 Methodology

This section presents the data sources used during this experiment, as well as the data processing and analysis techniques that were applied. The data sources subsection details the origin of the data sets used during this experiment. Follows a subsection with data processing tasks, concerning feature extraction processes and data preparation. Finally the subsection with data analysis details the supervised ML-algorithms employed during this experiment.

2.1 Data sources

This paper illustrates an experimentation based on two packet-level traces (i.e. libpcap format [11]) captured on a Gigabig Ethernet (GbE) link from Keio Campus network (Japan). The first capture (i.e. Keio-I) took place on the 8th of August, 2006 from 10:43:04 to 11:13:04. Keio-II was the second trace used and was captured on the 9th of August, 2006 between 16:18:41 to 16:48:31. During this experiment, Keio-I was used as training set and Keio-II was used as test set.

2.2 Data processing

The data processing task was held in the following five main steps:

1. Data extraction, from the network capture dump (with binary information).
2. Data transformation, which included the following parts:
   (a) Conversion from the CAIDA's Coral Reef tools [15] output to CSV files.
   (b) Anonymization of the sensitive information of the data set.

   (c) Map all the extracted packets to the correspondent flows.
3. Loading the data to a data mart created for this experiment.
4. Processing the new features using Octave [7].
5. Data validation process at which some features were re-calculated by using the mapped packets. This was done in order to ensure correct mapping of packets with their corresponding flows.

The two data sets used during this experiment contained raw network traffic data. Due to the raw nature of the data sets it was essential to perform some data processing tasks in order to extract meaningful packet and flow-level information. This task was performed using CAIDA's Coral Reef Suite [15]. Keio-I had 9.103.698 network packets composing 877.921 network flows. Keio-II had 7.398.404 network packets composing 497.042 network flows. These tools were able to determine the class of each network flow (from both data sets), based mainly on the payload information. This classification was used as this experiment ground truth. Both network traffic traces used during this experiment contained sensitive information, such as the source and destination IP addresses of each packet. To ensure the network users confidentiality, all IP addresses present in both data sets were replaced by an encrypted identifier. The anonymized data was used to populate a MySQL database. This database was created by following a dimensional model, to support the new features extraction process. In order to achieve this, a Java application was developed and used. Particularly, this Java application used the JavaOctave[1] to perform Octave calculations, complemented with the TFTB Toolbox [10] to generate the TF distribution signals based on the bytesize and the inter-arrival intervals of the network packets of each network flow, as well as their correspondent Renyi information. The commands used to calculate the Renyi information values from the generated signals were the following:

```
real(renyi(tfrpwv(hilbert([ bytesizeA bytesizeB ... bytesizeN ]'))));
real(renyi(tfrpwv(hilbert([ intervalA intervalB ... intervalN ]'))));
```

The first action was to generate the signal applying the Hilbert transform [13] employed by the Octave `hilbert()` function, used for deriving the analytical signal formed by both packets' byte sizes and packets' inter-arrival times. Subsequently, we computed the energy TF distribution of the analytical signal by applying the `tfrpwv()` function which in practice computes the smoothed PWV distribution. The resulted distribution enabled an investigation on the TF plane and the estimation of the Renyi information. This value represents the entropy of the energy-TF distribution. The goal of this work was to use this value as a new feature when using ML-algorithms, and evaluate its relevancy considering the accuracy of the formulated classification models. This process was accomplished by using the `renyi` function in Octave. The `renyi` function may return complex numbers that are not employed by the supervised ML-algorithms. To overcome this drawback, only the real part of the Renyi information value was considered. This process was accomplished by using the `real` function in Octave.

[1] `http://kenai.com/projects/javaoctave` Jan 2012

During this experiment, Octave stood unable to process flows with more than 10.000 packets. Three different environments were used on two different machines, and every attempt to process flows with more than 10.000 packets failed. These environments were namely: Apple MacOS X Lion, Microsoft Windows 7 and Canonical Ubuntu Linux 11.10, with their respective versions of Octave (using both 32 and 64 bit architecture versions). The first machine had an Intel Core 2 Duo 2.0GHz processor and 2GB DDR3 1066MHz of RAM, where the second one had an Intel Core 2 Duo 2.8GHz processor and 8GB DDR2 800MHz of RAM. Due to Octave's aforementioned limitation it was feasible to compute Renyi values for flows that contained less than 10.000 packets. In addition, the minimum packet size for a given flow whilst computing the Renyi information was fixed to 5 since a lower threshold would not allow the generation of a meaningful signal. The subset of flows with 5 or more packets and less than 10.000 packets corresponds to 35.24% of Keio-I and 35.40% of Keio-II.

2.3 Data analysis

In order to conduct a robust analysis scheme for our new features, our experimentation strictly considered the subset of flows with a Renyi value. In more detail, it was feasible to compute the Renyi values for 309.420 records of Keio-I and 176.002 records of Keio-II. This filtering process resulted in a Keio-I subset with 29 distinct classes and a Keio-II subset with 25 distinct classes. Table 1 presents statistical information related with the classes distribution on both training and test data sets[2]. The number of records per class were rather irregular.

Table 1. Statistical information related with classes distribution

Number of Records per Class	Training Set	Test Set
Minimum	1	1
Average	10 670	7 040
Median	82	51
Standard Deviation	39 755	20 947
Maximum	211 567 (68.375% of data set)	99 593 (56.5863% of data set)

The ML-algorithms were executed using two different approaches: (1) with and (2) without the Renyi information features. We compared the results of the formulated classification models to analyze the impact of the new features. In this experiment we worked with features derived from two different sources: (1) from the packets' byte size and (2) from the packets' inter-arrival interval.

[2] Due to space restrictions, the classes distribution analysis is not presented in great detail in this paper.

Therefore, we used the following statistical information as features:

- **renyi_time** : Renyi information value for inter-arrival interval;
- **min_time**, **avg_time**, **std_time**, **max_time** and **total_time** : Minimum, average, standard deviation, maximum and sum of the packet inter-arrival interval values;
- **renyi_size** : Renyi information value for byte-size;
- **min_size**, **avg_size**, **std_size**, **max_size** and **total_size** : Minimum, average, standard deviation, maximum and sum of the packet byte-size values.

According to the basic statistical descriptors of minimum, average, median, standard deviation and maximum values, the training and test sets exposed similar statistical characteristics[3]. The ML-algorithms were employed via the RapidMiner [12] tool. Keio-I filtered data subset was used as the training set and Keio-II filtered data subset was used as the test set. By virtue of its algorithmic nature, the decision tree algorithm was used in order to first explore and analyze the importance of the newly added feature. Decision tree algorithm selects the most relevant features to build the decision paths of its tree. This behavior allows us to analyze if a given feature is relevant or not. We executed the supervised ML-algorithms several times varying its parameters to detect the values that produced better results. Nevertheless, we have used the following parameters for the decision tree algorithm: criterion was `gain_ratio`, the minimal size for split was 4, the minimal leaf size was 2, the minimal gain was 0.1, the maximal depth varied from 3 to 25 and the confidence was 0.25. The "pruning" technique is intended to minimize over-fitting issues but was not used during this experiment to allow a deeper analysis of the features influence. We have achieved over-fitting redundancy by using a test set obtained from traces of a different network capture. In parallel, the neural network algorithm was used with a varying training cycle (i.e. 20-320 cycles), the learning rate was 0.3 and the momentum was 0.2, whereas data was normalized and shuffling was not employed. Neural network was also used with multiple features combinations to analyze the impact of each feature on the resulting classification models. These executions were carried out with 50 training cycles, having a learning rate of 0.3 and a momentum of 0.2 (also with data normalized and not shuffled). The random forest algorithm used the following parameters: The number of trees in the forest varied from 5 to 80, the number of features to consider was $logM + 1$, where M was the number of inputs, and the random number seed was 1.0.

3 Results and Discussion

This section presents the results obtained during this experiment and the discussion about their meaning. The results subsection presents an analysis of the different accuracies of the models formulated by the ML-algorithms. We present the results concerning the decision tree, the neural network and the random forest algorithms. In discussion subsection we present the possible reasons for the observed differences.

[3] The values are not be presented on this paper due to the space restrictions.

3.1 Results

As already mentioned (Section 2.3), the decision tree algorithm was accommodated multiple times with varying features and the `max_depth` parameter in order to examine the efficiency of the newly added features. Ultimately we aimed at identifying the effect of different features on the overall accuracy rate. The `max_depth` parameter was also varied to analyze the behavior of the algorithm when different tree depth restrictions were applied. Defining a small value for the max tree depth forces the algorithm to select the most influential features to create its decision tree. In opposition, high max tree depth values allows the algorithm to use a larger number of features to produce its decision tree. The accuracy of the model was recorded for every execution of the decision tree algorithm, and was calculated by comparing the predicted class of a record from the test set with its actual class. Features related with the packets' inter-arrival interval were used in an initial approach. Having the intention to analyze the impact of the newly added features we considered to execute the decision tree with and without the new features. Figure 1 denotes that the features derived from the packets' inter-arrival intervals were not able to produce accurate classification models, and the addition of the Renyi information feature had no positive impact. The best results obtained during this experiment had an accuracy rate lower than 60%. A hypothetical classification model that defaults to the most populous class [4] would achieve an accuracy rate higher than the formulated decision tree models that used the packets' inter-arrival interval derived features.

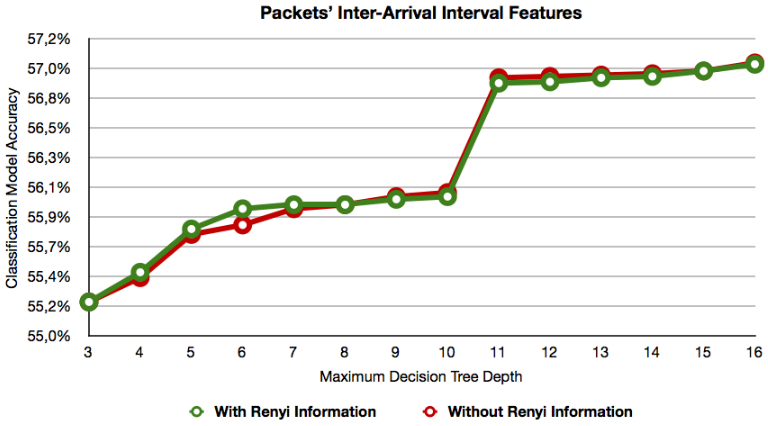


Fig. 1. Accuracy of the decision tree algorithm (packets' inter-arrival interval features).

On the other hand, the models formulated by the decision tree algorithm that used the packets' byte-size derived features presented an higher accuracy rate.

[4] The most populous class represented 68.375% of the training set.

Observing the Figure 2 it is possible to note that when the Renyi information is not used, the accuracy rate achieves a maximum of 74.28% (in decision trees with 10 levels) and remains with the same accuracy rate thenceforth. This accuracy rate is possibly the maximum for these features. Adding the Renyi information feature decreases the model accuracy when the tree depth is limited to a level lower than 16. A possible reason for this behavior is that the Renyi information feature may perform better when deeply associated with other features, and only on trees with more than 15 levels is possible to observe its advantage. Above 16 levels the Renyi information feature increases considerably the accuracy rate of the formulated models. Due to computational limitations[5], it was not possible to calculate the accuracy of decision tree models with more than 25 levels. Decision trees having a maximum depth of 25 levels had an accuracy rate of 75.62%.

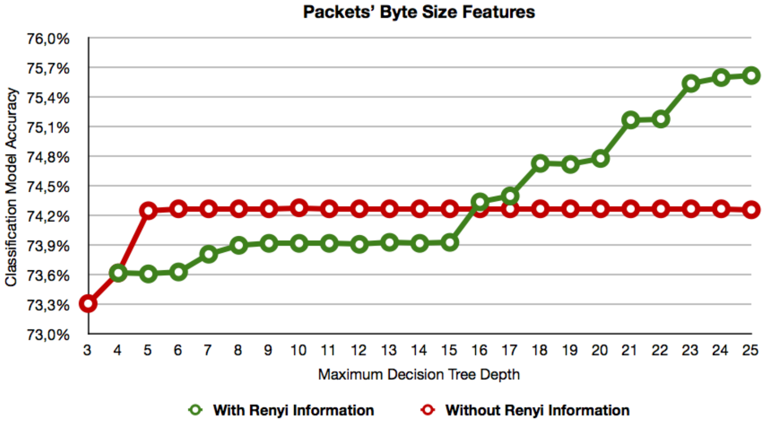


Fig. 2. Accuracy of the decision tree algorithm (packets' byte-size features).

The Neural Network algorithm was also used to analyze the importance of the `renyi_size` feature. We did not considered the `renyi_time` feature due to the poor results obtained with the decision tree algorithm. In an attempt to analyze the impact of the different features used by the neural network algorithm, we used the algorithm with multiple feature combinations. On the first execution stage, all the features were used, then several executions were performed without a feature at a time. This was done since our aim was to analyze the impact of the unused feature on the resulting model accuracy (see Figure 3). It could be observed that the feature with the highest classification impact was the `std_size` that represented an accuracy drop of 44.67%, followed by the feature `avg_size` that represented an accuracy drop of 37.55%. In addition, the Renyi information feature (`renyi_size`) was the third most influential feature, representing an accuracy drop of 33.62%. The remaining three features were the less influential

[5] Processing power and RAM resources.

in the resulting model accuracy, representing an accuracy drop ranging from 28.31% to 2.31%. The neural network algorithm performance varied much with the different features selection.

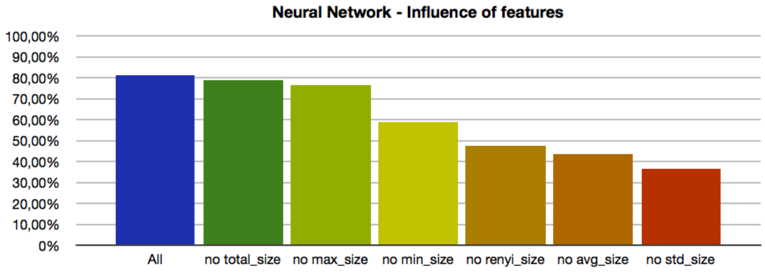


Fig. 3. Comparison of different features' impact using the neural network algorithm.

For a complementary analysis, the Decision Tree, the Neural Network and the Random Forest algorithms were executed in order to compare their performance when using the packets' byte-size features including the Renyi information. During the experiment the number of training cycles of the neural network algorithm was varied from 20 to 320, and the number of trees in the forest of the random forest algorithm was varied from 5 to 80[6]. The model accuracy rates and the algorithm execution times were recorded, to allow a comparison between the performance of the three machine learning algorithms that were used during this experiment. Figure 4) denotes that the accuracy results showed little variation respecting each supervised ML-algorithm. Comparing with the other algorithms, neural networks were the ones who varied most with the increase of execution times[7]. However, neural networks never surpassed the accuracy rates of random forests, that outperformed the other two algorithms during this experiment with a top accuracy rate of 85.65%[8].

3.2 Discussion

The results obtained during this experiment showed that features derived from the packets' inter-arrival intervals did not contribute to produce accurate classification models. These particular features has been indiscriminately used in the past and were considered as de-facto features in traffic classification. This work contributes to emphasize the irrelevancy of these kind of features on traffic classification. In fact, during this experiment they contributed to the generation of classification models with accuracy rates lower than a hypothetical-model that defaults to the most populous class.

[6] The parameters details are not presented in this paper due to space restrictions.
[7] Execution time varied with the number of training cycles.
[8] When using 80 trees in the forest.

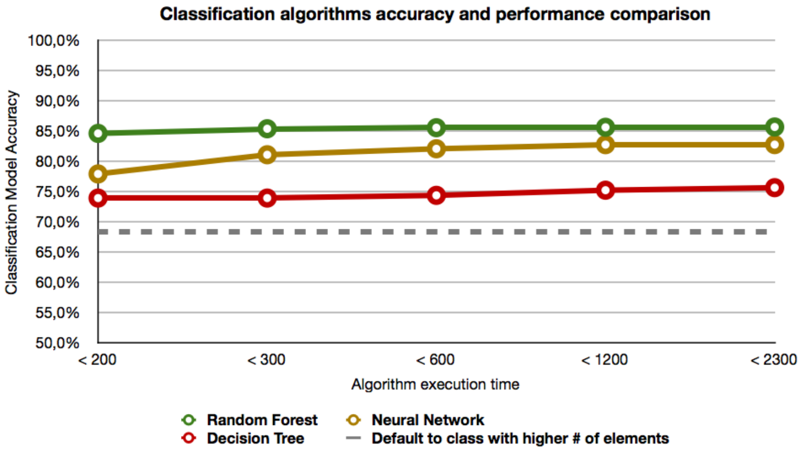


Fig. 4. Comparison of classification algorithms (accuracy rates vs. execution times).

In the contrary, the use of features that derived from the packets' byte-sizes did contribute to produce rather accurate classification models in certain conditions. A reasonable explanation for the higher accuracy of the features derived from the packets' byte-sizes, compared with the features derived from the packets' inter-arrival intervals, is related with the operation of the IP networks. A flow's packet size is defined at an application level, and remains unchanged during the network transfer [21], allowing the exploration of their characteristics. However, the inter-arrival interval of the packets may be changed during the network transfer, due to the topological distance and the inter and intra-transfers of packets in local or foreign ISPs. Therefore, the inter-arrival frequency of the packets looses the source application footprint.

The accuracy of the classification models obtained from these subsets may not be directly compared with other experiments where unfiltered data sets were used. Using the entire data set as well as extra features such as protocol flags may lead to higher accuracy rates, as presented by H. Kim et al [16]. The available computational resources were a major difference from this experiment and others of the state of the art [16]. Under powerful computational resources we consider as highly possible to explore resource demanding techniques and achieve higher accuracy classification models. The Support Vector Machine algorithm (SVM) [6] was also executed during this experiment, but was only able to process small samples of the data set, due to computational resources limitations. Despite all these restrictions, this study showed that even with limited computational resources it was feasible to present the promising rates that can be achieved with the usage of the newly proposed features.

4 Conclusion

This work presents a new feature (Renyi information of a signal representing the energy-TF distribution of the Smoothed-PWV, based on the network features of the packets' byte-size) to use with supervised ML-algorithms, when generating network traffic classification models. The conducted experiment showed that the proposed feature increases the accuracy of the formulated classification models in certain conditions. Therefore, this new feature should be properly explored in future researches.

Using features derived from the packets' inter-arrival intervals shown to be ineffective when formulating classification models. One of the possible reasons for that observation is that the inter-arrival intervals of the network packets may suffer changes during the network transfers, due to traffic prioritization or routing operations. These events may prevent the use of features based on this type of information to classify the network flows. On the other hand, the byte-size information of the network packets are defined at an application level. This information remains unchanged during the network transfer, and may turn into an unique identifier of the sender application. The pattern created by the variation of the byte-size information of all the packets from a given flow may be explored as a source of identity of the sender application. During this work, this pattern (treated as a signal) was exploited with the use of the entropy value (Renyi information) of its energy-TF distribution as a feature. Several experiments were conducted using supervised ML-algorithms to evaluate the influence of the presented feature. The results obtained shown that in certain conditions this feature increases the accuracy of the formulated classification models, therefore it should be properly explored in the future.

An important limitation of this approach is the confinement of using only flows with more than 4 packets. The large amount of resources needed to process the feature when dealing with network flows with a very large number of network packets are also a major drawback. During this experiment only flows with less than 100.000 packets were able to be processed.

Nevertheless, the results denotes an increase of the formulated classification models accuracy that suggests the usefulness of a deeper analysis of this feature. The presented feature may be used with other offline traffic classification techniques, based on supervised ML-algorithms, such as the SVM algorithm. With higher computational resources it may be possible to analyze in depth the influence of the presented feature using more resource demanding classification algorithms.

The combination of this new feature with other existent features should also be tested to define the best feature combination when using a supervised ML-algorithm to classify network traffic.

Acknowledgment

The authors wish to acknowledge the help of H. Kim and CAIDA in data collection and network capture processing tools.

References

1. R G Baraniuk, P Flandrin, A J E M Janssen, and O J J Michel. Measuring time-frequency information content using the Renyi entropies. *IEEE Transactions on Information Theory*, 47(4):1391–1409, 2001.
2. Christopher M Bishop. *Pattern Recognition and Machine Learning*, volume 4 of *Information science and statistics*. Springer, 2006.
3. B Boashash and P O'Shea. Use of the cross Wigner-Ville distribution for estimation of instantaneous frequency. *IEEE Transactions on Signal Processing*, 41(3):1439–1445, 1993.
4. Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.
5. L. Cohen. Time-frequency distributions-a review. *Proceedings of the IEEE*, 77(7):941–981, July 1989.
6. Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, volume 3. Cambridge University Press, 2000.
7. John W Eaton. GNU Octave. *History*, (February):1–356, 1997.
8. J Erman, A Mahanti, M Arlitt, I Cohen, and C Williamson. Offline/realtime traffic classification using semi-supervised learning. *Performance Evaluation*, 64(9-12):1194–1213, 2007.
9. E Fiesler. Neural network classification and formalization. *Computer Standards Interfaces*, 16(3):231–239, 1994.
10. Patrick Flandrin and Olivier Lemoine. Time-Frequency Toolbox. *Recherche*, pages 1995–1996, 1996.
11. Luis Martin Garcia. Programming with Libpcap - Sniffing the Network From Our Own Application. *Hakin9 Magazine*, 3(2):38–46, 2008.
12. K Goslin and M Hofmann. Integrated Tutorial Tool for RapidMiner 5. In *RCOMM 2010*. University of Dortmund, 2010.
13. Mathias Johansson. The Hilbert transform. *October*, 19(1):1–12, 2006.
14. Shivkumar Kalyanaraman. Quality of Service (QoS). *Quality*, pages 29–55, 2006.
15. K. Keys, D. Moore, R. Koga, E. Lagache, M. Tesch, and K. Claffy. The Architecture of CoralReef: An Internet Traffic Monitoring Software Suite. In *Proceedings of the Passive and Active Network Measurement Workshop (PAM 2001)*, Amsterdam, Netherlands, 2001. RIPE NCC.
16. H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee. Internet traffic classification demystified. In *Proceedings of the 2008 ACM CoNEXT Conference (CONEXT 2008)*, pages 1–12, New York, New York, USA, December 2008. ACM Press.
17. K. Marnerides, A. *On Characterization & Decomposition of Internet Traffic Dynamics*. Phd thesis, Lancaster University, 2011.
18. H Michiel and K Laevens. Teletraffic engineering in a broad-band era. *Proceedings of the IEEE*, 7(8):721–2033, 1997.
19. T. Nguyen and G. Armitage. A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys & Tutorials*, 10(4):56–76, 2008.
20. J R Quinlan. Learning decision tree classifiers. *ACM Computing Surveys*, 28(1):71–72, 1996.
21. Rishi Sinha, Christos Papadopoulos, and John Heidemann. Internet Packet Size Distributions : Some Observations. *Network*, pages 1–7, 2007.

Knowledge Extraction from social networks

Online Survey and data mining

Nuno Filipe Cruz Barbosa da Silva

University of Porto
Faculty of Engineering of U.Porto,
Paranhos, 4200 Porto, Portugal

nunobarbosasilva@gmail.com

Abstract
In this research were collected data on the characteristics and preferences of users of online social networks, for the collection of information was used an online survey which had 196 responses, this information after treatment resulted in a dataset with (N = 166).

The goal of this research is study the influence of social networks in interpersonal relationships, identify criteria to characterize users who buy with influence of social networks, gaining knowledge about groups susceptible to advertising on social networks in order to allow for channelling investment and increase the return.

The collected data was prepared, classified with decision trees and we made a cluster analysis, obtaining a subset of 4 features (age, children, profession and Hours a day in the social networks), that can classify and predict with 96% accuracy if a user of social networking online is a potential buyer of products advertised in online social networks.

Keywords. Social networks, Marketing, Knowledge extraction, decision tree, clustering, statistical classification.

1 Introduction

Online social networks are shared every day by millions of people who communicate, interact, share information between people of different nationalities, cultures and areas of knowledge and interests, visitors can enjoy social activities such as chat, share multimedia, photos, videos, online games, post ideas, opinions, publicize activities, products and events such as workshops, demonstrations, sports activities, and others, have created new ways to communicate and socialize.

This phenomenon is a globalized world and created new habits, the high turnout and its impact on the habits of the people has sparked interest among researchers from various fields, computer science, mathematics, humanities, economics, marketing and others.

In a growing number of millions of users becomes important to know their characteristics and interests in order to use social networks more efficiently and directed.

This study analyses the characteristics and interests of users of social networks in order to classify them according to already have purchased a product advertised on social networking, is performed a collection of features and interests with an online survey. The data set was prepared, classified with decision trees and we made a cluster analysis, with the tools SPSS and RapidMiner.

The document consists of 7 points, the point 2 corresponds to the introductory description of the algorithms used, decision tree and clustering, Section 3 contains the experimental context, the characterization and description of the dataset features, characteristics of participants and description of the dataset preprocessing, the point 4 explains the experimental evaluation, section 5 presents the results, the section 6 is the conclusion and finally section 7 contains the references.

2 Algorithms used

2.1 Decision tree

Decision tree learning is a method commonly used in data mining, is a classifier that is depicted in a flowchart like tree structure, which has been widely used to represent classification models, due to its comprehensible nature that resembles the human reasoning. Due to their intuitive representation, they are easy to assimilate by humans. They can be constructed relatively fast compared to other methods. The accuracy of decision tree classifiers is comparable or superior to other models. Decision-tree algorithms present several advantages over other learning algorithms present several advantages over other learning algorithms, such as robustness to noise, low computational cost for the generation of the model, and ability to deal with redundant attributes.

The goal is to create a model that predicts the value of a target variable based on several input variables.

Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions.

Most decision-tree algorithms are based on a greedy top-down recursive partitioning strategy for tree growth. They use different variants of impurity measures, such as information gain[1], gain ratio[2], Gini index[3] and distance based measures to select an input attribute to be associated with an internal node.

[1] $Gain(S,A) = Info(N,P) - Info(A)$

In data mining, decision trees can be described also as the combination of mathematical and computational techniques to aid the description, categorisation and generalisation of a given set of data.

2.2 Clustering

Cluster analysis or clustering can be considered the most important unsupervised learning problem; so, as every other problem of this kind, it deals with finding a structure in a collection of unlabelled data, is a method by which large sets of data are grouped into clusters of smaller sets of similar data. Cluster is a collection of data objects similar to one another within the same cluster and dissimilar to the objects in other clusters.

Clustering is a main task of explorative data mining, and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

Cluster analysis itself is not one specific algorithm, but the general task to be solved. Various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them can achieve it. Popular notions of clusters include groups with low distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery that involves try and failure. It will often be necessary to modify pre-processing and parameters until the result achieves the desired properties.

3 Experimental context

3.1 Online Survey

For data collection was carried an online survey, with 16 questions about the characteristics and interests of users of online social networks, this survey aims to extract knowledge for the study of social networks, 196 people attended this survey, this information after treatment resulted in a dataset with (N = 166).

KNOWLEDGE EXTRACTION FROM SOCIAL NETWORKS: DATA MINING

Questionnaire easy to understand and fast filling. Target Audience: Users of online social networks. Work performed under the Course Unit "Knowledge Extraction and Machine Learning." This work aims to extract knowledge for the study of social networks using methods of classification and prediction.

Fig. 1. SURVEY `http://dl.dropbox.com/u/2761856/quiz.html`

[2] $Gain\ Ratio(S, Day) = \frac{Gain(S,Day)}{Split(S,Day)}$

[3] $Gini(T) = 1 - \sum_{j=1}^{n} p_j^2$

3.2 Data set characterization

3.2.1 Participants

In the online questionnaire we obtained 196 responses, which after treatment resulted a dataset with (N = 166), comprising 58 females and 108 males between 18 and 62 years old (Mean = 32,65; Std. Deviation=8,294).

With regard to marital state the dataset consists of 94 single subjects, 64 married and 8 divorced a total of 166 subjects.

Regarding the use of different social networks, it was found that the most used social network is the Facebook with 152 users, the LinkedIn was the second with 93 users, the rest of the social networks had no significant results.

Of the 166 responses, 40 participants said that they have bought products by influence of advertising on social networks and all the people who have bought products have aged between 25 and 41 years.

3.2.2 Dataset features

In this section we describe the features used in this investigation. The data set consists of 26 features that resulted from the online survey described in section online survey and from the task of pre-processing described in section Description of the data set Conducted pre-processing task.

In the table 1 we present all the features and their significance.

Feature Nº	Name	Description
1	Age	Person's age
2	Gender	Person's gender
3	Nationality	Person's nationality
4	Marital_Status	Person's marital status
5	Children	This feature contains the value 0 if the person has no children or 1 If the person has children.
6	Education	Person's level of education, primary, secondary, higher, master's, doctoral, other
7	Profession	Person's profession
8	Hours_Network_Day	Hours a day in social networks
9	Goal_join	The goal of the person for to join in a social network
10	Bouht_some_product	if the person bought some product advertised on social networks this feature have the value 1, if do not bought it has the value 0 other cases has the value 3.
11	Participate_by_invitations	If the person has engaged in any event posted on social networks
12	Agree_with_use_socialnetworks_busine	If person agrees with the use of social net-

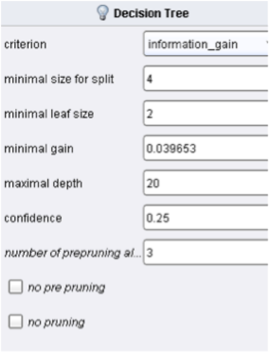
	ss_purpos	works for business purposes
13	I_met_most_of_the_my_frineds	how you met most of your friends
14	When_have_something_important_say_prefer	When I have something important to say, I prefer
15	Social_network	person's social networks
16	facebook	1- if using facebook 0 - if not use
17	Hi5	1- if using HI5 0 - if not use
18	Google_plus	1- if using Google Plus 0 - if not use
19	twitter	1- if using Twitter 0 - if not use
20	linkedin	1- if using LinkedIn 0 - if not use
21	Academia.edu	1- if using Academia.edu 0 - if not use
22	orkut	1- if using orkut 0 - if not use
23	Flickr	1- if using Flickr 0 - if not use
24	MySpace	1- if using MySpace 0 - if not use
25	Netlog	1- if using NetLog 0 - if not use

Table 1. Data set Features

3.2.3 Characterizing dataset for meta-learning

The task of characterizing dataset for meta-learning is to capture the information about learning complexity on the given dataset. This information should enable the prediction of performance of learning algorithms. It should also be computable within a relative short time comparing to the whole learning process. In this section we introduce new measures to characterize the dataset by measuring a variety of properties of a decision tree induced from that dataset. The major idea here is to measure the model complexity by measuring the structure and size of decision tree, and use these measures to predict the complexity of other learning algorithms. We employed the standard decision tree learner. There are several reasons for selecting decision trees. The major reason is that decision tree has been one of the most popularly used machine learning algorithms, and the induction of decision tree is deterministic, i.e. the same training set could produce the similar structure of decision tree. (Y. Peng et al 2002)

To characterize the dataset was analysed the frequency distribution of all variables and built a decision tree using the SPSS and RapidMiner and in both cases was used as target variable bought_some_product - have already purchased a product by advertising influence on social networks.



Decision Tree

criterion: information_gain

minimal size for split: 4

minimal leaf size: 2

minimal gain: 0.039653

maximal depth: 20

confidence: 0.25

number of prepruning al...: 3

□ no pre pruning

□ no pruning

Fig. 2. Decision Tree Parameters

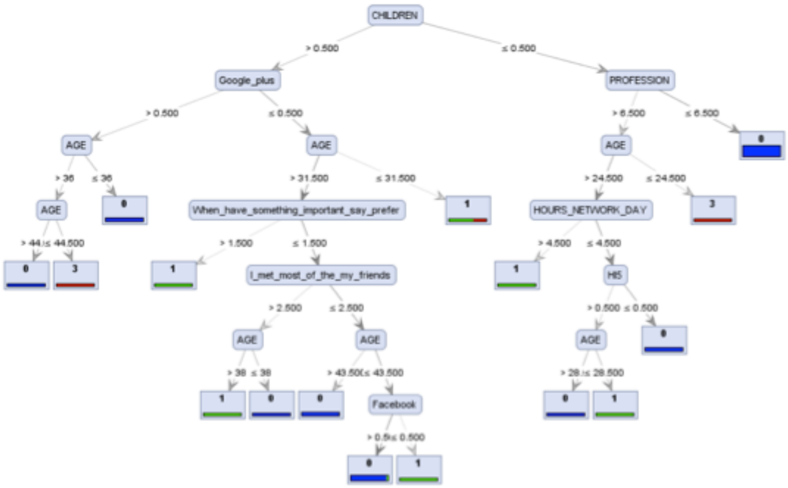


Fig. 3. Decision tree

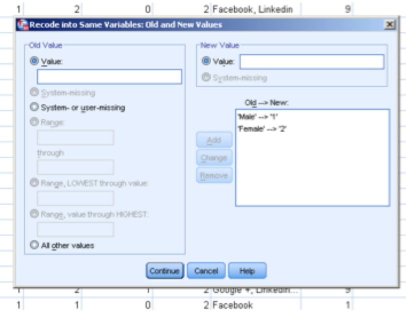
To build a decision tree with RapidMiner was necessary to lower the minimal gain as a result we obtained the tree in fig.2, with a bit of entropy but allowed us to assess the complexity of the problem and make an initial sub-dataset for the problem, constituted by the features 1, 5, 7, 8, 14, 15.

3.3 Description of the data set pre-processing task conducted

In the collected data set were eliminated the missing values, the data in string format were coded in numbers and the information with multiple possibilities was coded into single variables.

The elimination of missing values was done by hand, eliminating the records with incomplete values.

The coding of variables of type STRING in numbers was carried out using SPSS functionality Transform -> Recode into Same Variables, fig 3.

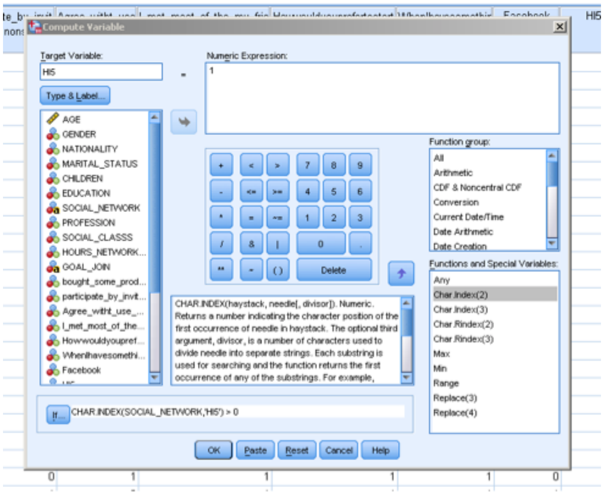


Recode into Same Variables: Old and New Values

Old --> New:
'Male' --> '1'
'Female' --> '2'

Fig. 4. Transform -> Recode into Same Variables

The transformation of variables with multiple values in single variables was carried out using SPSS functionality of the Transform-> Compute Variable, fig 4.



Compute Variable

Target Variable: HI5

Numeric Expression: 1

If... CHAR.INDEX(SOCIAL_NETWORK,'HI5') > 0

CHAR.INDEX(haystack, needle[, divisor]). Numeric. Returns a number indicating the character position of the first occurrence of needle in haystack. The optional third argument, divisor, is a number of characters used to divide needle into separate strings. Each substring is used for searching and the function returns the first occurrence of any of the substrings. For example,

Fig. 5. Transform-> Compute Variable

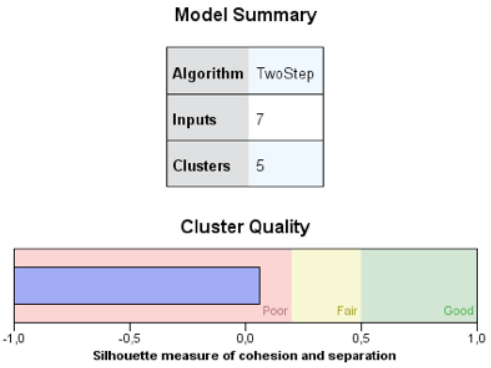
4 Experimental evaluation

4.1 TwoStep Cluster Analysis

After the collection, preparation and characterization of the data-set,

as explained in section "Description of the data set pre-processing task conducted", we initiated the experimental evaluation

We start with TwoStep Cluster Analysis, for make the cluster we chose for categorical variables, the variables chosen by the decision-tree they are (1, 5, 7, 8, 14, 15 and 10.

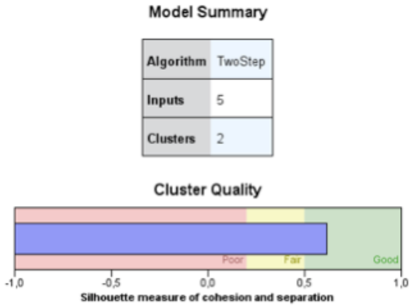


Model Summary

Algorithm	TwoStep
Inputs	7
Clusters	5

Fig. 6. TwoStep Cluster Analysis

With the initial data set was not obtained good results, so we continued to evaluate with other sub-dataset, we have got a good, result fig 6, with the sub-dataset 10, 9, 5, 1 and 7.



Model Summary

Algorithm	TwoStep
Inputs	5
Clusters	2

Fig. 7. TwoStep Cluster Analysis

So we've reached 5 variables that make the TwoStep Cluster Analysis with good results, they are the age, if have children, profession, have already purchased a product by advertising influence on social networks and goal to join in a social network.

Final Cluster Centers

	Cluster					
	1	2	3	4	5	6
AGE	51	30	24	62	46	25
CHILDREN	1	0	0	1	1	0
PROFESSION	9	3	2	3	9	9
bought_some_product	0	0	0	0	0	1

Final Cluster Centers

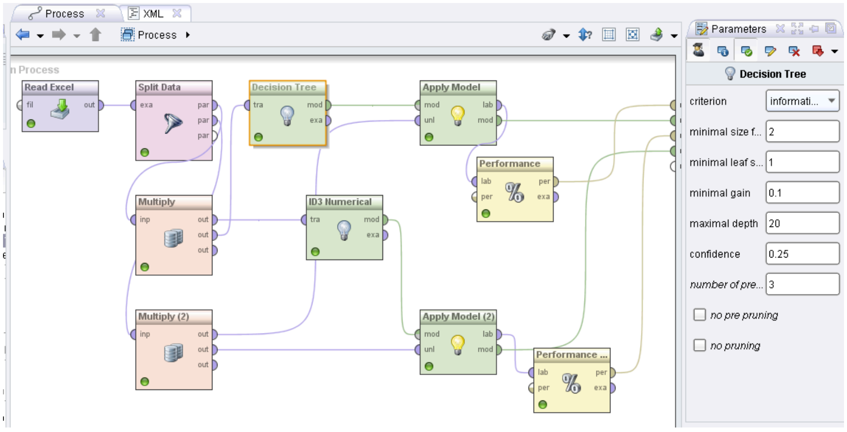
	Cluster				
	7	8	9	10	11
AGE	19	36	29	35	53
CHILDREN	0	1	0	1	1
PROFESSION	1	3	9	7	3
bought_some_product	0	1	0	0	0

Table 2. K-Means Cluster Analysis

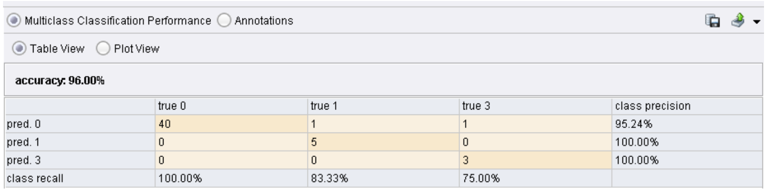
4.2 Decision tree

For classify the data set and predict was used the software rapidminer and built the following Main Process:

It was imported the data-set of a excel file and divided into training set and test set, the training set was 70% of the data-set and the test set was 30%. In order to compare different algorithms with the same training set and the same set of tests was used an operator that replicates the set of input data into multiple outputs, we also use two operators allowed us to evaluate the performance of each model and came to the conclusion that the best algorithm for this data set was the decision tree with Criterion = information_gain, minimal size for split = 2, minimal leaf size = 1 and all other values are the standard values. The variable bought_some_product was used as target variable.



criterion: informati...
minimal size f...: 2
minimal leaf s...: 1
minimal gain: 0.1
maximal depth: 20
confidence: 0.25
number of pre...: 3
□ no pre pruning
□ no pruning



accuracy: 96.00%

	true 0	true 1	true 3	class precision
pred. 0	40	1	1	95.24%
pred. 1	0	5	0	100.00%
pred. 3	0	0	3	100.00%
class recall	100.00%	83.33%	75.00%	

Fig. 8. Main Process

5 Results

Regarding the use of different social networks, it was found that the most used social network is the Facebook with 152 users, the LinkedIn was the second with 93 users, and the rest of the social networks had no significant results.
What concern about, if already have bought products advertised in social networks, of the 166 responses, 40 participants said that they have bought products by influence of advertising on social networks and all the people who have bought products have aged between 25 and 41 years.

In this research we can identify characteristics of users of online social networks that can classify with 96% accuracy users of social networks as buyers of products advertised in online social networks. The characteristics are in the table as follows:

age	The age of the person
children	If the person has children
bought_some_product	Have already purchased a product by advertising influence on social networks
goal_join	Goal to join in a social network
profession	The profession of the person

Table 3. Characteristics

With these attributes and with the help of decision trees, we train a model that achieved a 96% accuracy that can classify a user of social networks and predict if he is a buyer or not buyer of products advertised in social networks.

6 Conclusion

In this research were collected data on the characteristics and preferences of users of online social networks, for the collection of information was used an online survey to which 196 responded responses, this information after treatment resulted in a dataset with (N = 166).

The goal of this research is study the influence of social networks in interpersonal relationships, identify criteria to characterize clusters of users who buy with influence of social networks and classify user in function of if a users are buyers of products advertised on social networks, gaining knowledge about groups susceptible to advertising on social networks in order to allow channelling investment and increase the return.

The collected data was classified with decision trees and are made a cluster analysis, obtaining a subset of 4 features, the age, if have children, profession and the goal for to join on a social network, that variables can classify and predict with 96% accuracy if a user of social networking online is a buyer of products advertised in online social networks.

7 References

1. A Survey of Evolutionary Algorithms for Decision-Tree Induction, R C Barros, M P Basgalupp, A De Carvalho, A A Freitas.
2. Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management, Michael Berry and Gordon Linoff - John Wiley & Sons Inc – Third Edition, 2011.
3. Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations, Ian H. Witten, Eibe Frank - Morgan Kaufmann, Third Edition, 2011.
4. Handbook of Statistical Analysis and Data Mining Applications, R. Nisbet, J. Elder and G. Miner – Academic Press, 2009.
5. Improved Dataset Characterisation for Meta-learning, Yonghong Peng1 , Peter A. Flach1 , Carlos Soares , and Pavel Brazdil.
6. Mining: Concepts and Techniques, Jiawei Han, Micheline Kamber, Morgan Kaufmann – Third Edition, 2011.

Spatio-temporal clustering methods classification

Hadi Fanaee Tork
info@fanaee.com

Abstract. Nowadays, a vast amount of spatio-temporal data are being generated by devices like cell phones, GPS and remote sensing devices and therefore discovering interesting patterns in such data became an interesting topics for researchers. One of these topics has been spatio-temporal clustering which is a novel sub field of data mining and Recent researches in this area has focused on new methods and ways which are adapting previous methods and solutions to the new problem. In this paper we first define what the spatio-temporal data is and what different it has with other types of data. Then try to classify the clustering methods and done works in this area based on the proposed solutions. classification has been made based on this fact that how these works import and adapt temporal concept in their solutions.

Keywords: Spatial Clustering, Spatio-temporal Clustering, Data Mining, GIS

1 Spatio-Temporal Data

Before bringing a clear definition about spatio-temporal clustering, we should explain more about the nature of spatial and spatio-temporal data and its difference with classical data. As it can be seen on figure 1 which shows a sample of classical data, each points is represented by its x and y values in a 2-D space and it doesn't show anything about the spatial or temporal situation of the points.

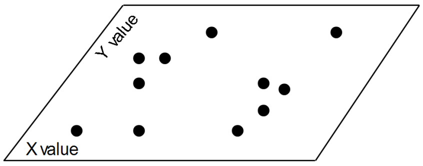


Figure 1 – a Simplified classical data in 2-D space

In Spatial data, the data item is representing by its spatial location (usually on earth) and it doesn't provide any information about other features of that item. Figure 2 shows a sample of spatial data. In this case we don't also have any temporal information for each data item which cause spatial data to be different from spatio-temporal data.

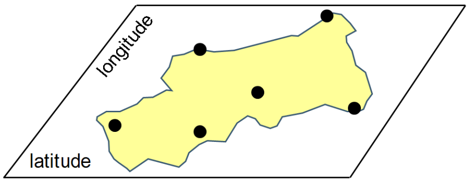


Figure 2 – A sample of spatial data

Spatio-Temporal data is more complicated, because time factor can be involved in different ways. By the way if we had a temporal information for each data item we are facing with spatio-temporal data and not spatial data. We have three types of spatio-temporal data:

Events : if there is no correlation between data items and data set doesn't include any identification for each data item or at least its not important for us. A sample of such data set is presented in table 1. As it can be seen there are 18 objects that for each object we have both spatial and temporal data. So for example <X6,Y7,2> implies on object 6 and it shows that object 6 has occurred in time = 2 and its Longitude and Latitude are X6 and Y7 respectively.

Table 1. A sample data set of Events data

Longitude	*Latitude*	*Time*
X1	Y1	1
.	.	1
.	.	
.	.	
X6	Y6	1
X7	Y7	2
.	.	2
.	.	
X12	Y12	3
.	.	3
.	.	
X18	Y18	3

Geo-Referenced data items : such data items are objects that in addition to their spatial and temporal position, a non-spatial value related to them is added to data item. A sample of such data type is presented on Table 2. for example each data items in this case can be a weather station location and corresponding temperature value at the different time sequences. For instance <X6,Y6,2,V62> implies on object number 6 with Longitude X6 and Latitude Y6 and 2 shows that this object has occurred In time=2 and V62 is for example temperature value related to this object. Likewise V61 is the temperature value related to object 6 but at time=1.

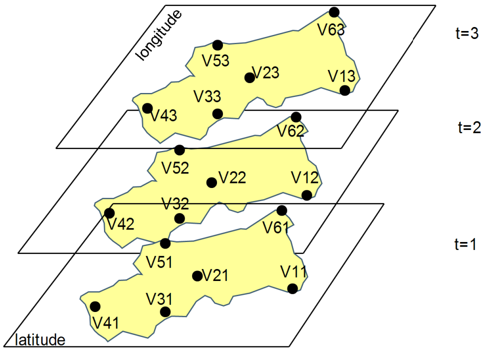


Figure 3 – Geo-Referenced data items

Table 2. A sample data set of Geo-Referenced data items

Longitude	*Latitude*	*Time*	*Value*
X1	Y1	1	V11
X2	Y2	1	V12
.	.	1	.
.	.		.
.	.		.
X6	Y6	1	V16
X1	Y1	2	V12
X2	Y2	2	V22
.	.	2	.
.	.		.
.	.		.
X6	Y6	2	V62
X1	Y1	3	V13
X2	Y2	3	V23
.	.	3	.
.	.		.
.	.		.
X6	Y6	3	V63

Moving data items: In such data sets, data items are moving and are not static. For identification of data items entity they should have an ID to be able to trace their movement during the time. For example figure 4 shows some moving points data set during the time between t=1 and t=3. As it can be seen from figure, green object is moving to left, orange object has not moved at least from t=1 to t=3, blue and yellow objects are being closer together and white object is moving to right. Also we have red object in t=1 and we don't have this object in the next time sequences. It means we have not had any information about the red object location at that periods. It can be due to lack of GPS data at that time (e.g. user has entered to the inside of a house). As we observed, the important thing that enabled us to identify the moving behavior of objects was their color or better say their identification parameter. So in such data sets we have all ID, spatial location and temporal information as records. Table 3 shows a sample of such data sets.

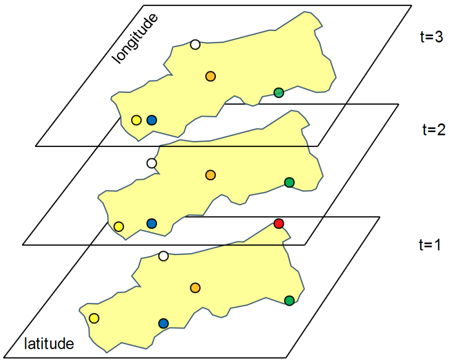


Figure 4 – Moving data items

Table 3. A sample data set of Moving data items

ID	*Longitude*	*Latitude*	*Time*
GREEN	X1	Y1	1
GREEN	X2	Y2	2
GREEN	X3	Y3	3
RED	X4	Y4	1
RED	---	---	2
RED	---	---	3
ORANGE	X5	Y5	1
ORANGE	X6	Y6	2
ORANGE	X7	Y7	3
BLUE	X8	X8	1
BLUE	X9	Y9	2
BLUE	X10	Y10	3
WHITE	X11	Y11	1
WHITE	X12	Y12	2
WHITE	X13	Y13	3
YELLOW	X14	Y14	1
YELLOW	X15	Y15	2
YELLOW	X16	Y16	3

2 Spatial data Clustering

Spatial data clustering is not a new task and we had already the same concept in classical data clustering. The only difference is the difference between the nature of input values. In classical data as it has been shown in figure 1, we have values which can be presented in multi-dimensional vectors and therefore <x,y> represents two distinctive values in a 2-d space. With this definition spatial clustering can be simplified as a vector with two values like x,y but this time instead of values x and y, the longitude and latitude a object can be replaced. With this assumption the spatial clustering problem is exactly like clustering of 2-d vectors.

This clustering also can be done via density-based methods or distance-based methods. distance-based methods have two weakness which leads to be not suitable for spatial data clustering, first they need a number of clusters as an input and second they allocate all objects to the clusters and never identify noises. By the way there are some related works like[13] which firstly transform spatial or spatio-temporal data to same length multi-dimensional vectors and then apply a generic clustering algorithm like k-mean on the data.

Density-based clustering of spatial data mostly is based on two well-known density-based algorithms DBSCAN (Density-Based Spatial Clustering of Applications with Noise) and OPTICS(Ordering Points To Identify the Clustering Structure). In both of these algorithms there is a density function which compute the distance of objects in order to allocate them into the clusters. Two input parameters of these algorithms are distance threshold and minimum number of neighbors which can make objects a member of cluster or a noise. The most important property of density function that distinguish them from other algorithms is flexibility of that distance function so that we can configure this distance function according to our specific purpose. Despite of similarity in login between OPTICS and DBSCAN the most important difference between them is related to the order of visiting of the objects in data set. Contrary to DBSCAN, OPTICS visit the objects based on their ordered distances from the visited objects, thus its no such sensitive to the input parameters and also have good tolerance to the noises. Also DBSCAN can not recognize hierarchical clusters and does not perform well in case the clusters have different densities, while OPTICS can do both.

3 Spatio-temporal data Clustering

As a matter of fact, spatio-temporal data clustering is not much difference with spatial data clustering. All the related works are trying to import time concept in the data or algorithms as a threshold or by the distance function or even they transform the spatio-temporal clustering problem to a multi-sequence spatial data clustering. So with this explanation, we can say there are the following possible strategies:

- Different distance functions

- Importing time to the spatial data
- Transform spatio-temporal data to the new objects
- Configure the algorithm
- Progressive clustering
- Performing clustering task on each time sequences
- Thresholds-based clustering
- Spatio-temporal pattern discovery

In the remained part of the paper we explain more about the above strategies.

3.1 Different distance functions

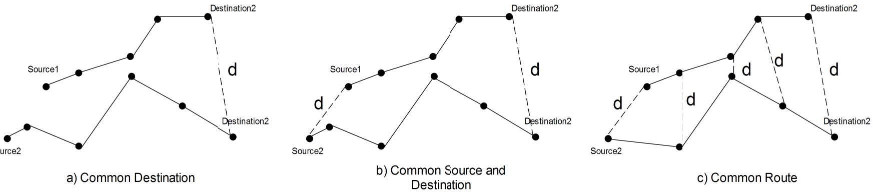
In this strategy, different distance functions are employed according to the specific goal of analysis. In a part of analysis we might use one specific distance function and in other part we might use another one. This strategy can be applied to two different ways with respect to the type of the given data :

Events : When we are dealing with events data (like data set of table 1), we have some events like crime points which are occurred in a specific times. In this case, we like to apply a density-based clustering algorithm like DBSCAN on data set.s However, in the stage of search for neighbors of a point in DBSCAN we need a function that could calculate the distance between the give point and other points and then retrieve the neighbors of the given point. This distance function can be selected according to the goal of analysis. We have three possible scenario as follows:

> **Spatial distance function**: in this case, we ignore the temporal part of data items and apply for example DBSCAN on all data items. In this case our problem is changed to spatial data clustering. The goal could be finding the regions of the city that have the most crime activates. But yet we don’t know in which periods of time these crimes are occurring more.
> **Temporal distance function** : in this case, we ignore the spatial part of the data items and apply clustering on all data items. Our problem will be changed to temporal clustering. As the final result we can understand in which periods of the year, month, week or time , depending on the type of temporal part(year, month, week, stime,…) the crime amounts are the high or low.
>
> **Spatio-Temporal distance function** : Some works like [13],[14] benefit from this method and created modified algorithm namely ST-DBSCAN while other works like[4] a spatio-temporal distance function with spatial and time threshold is using in order to discover the interdiction process over the three years. In both ways the goal is to discover the spatio-temporal behavior of events. For example we might like to discover the spatio-temporal regions of crime spots. More precisely as the results, we may understand that downtown of the city between hours 19 to 23 and a specific

zone of city between 12 to 13 are crime spots. So these spatio-temporal regions can be reported to the police to have more patrol there during that time periods. In order to cluster such data we need a little modification in one stage of algorithm and that is when algorithm is going to calculate the distance between points. As well as spatial threshold like 5km We need to add a time threshold like 1.5h to filter the points that has not enough distance to the given point. In fact firstly when for example DBSCAN searches for objects that have lower distance than 5km to the given object. Then before starting the counts of the retrieved objects we add another filter step. In this new filter step, we check that whether these objects have lower time distance of 1.5h to the given object or not. If they didn’t satisfy the threshold they will be removed from the retrieved list. Then we count the final retrieved list after filtering. If the number of objects was greater than MinPts (minimum number of points required to form a cluster) a cluster is created and that point is a core point. Another section of the algorithm is exactly like normal one.

Trajectories or Moving points: In this case, we are dealing with moving points data usually called trajectories. As we already discussed, the most important different between moving points data and events data is that in events data, objects are not moving and they are constant points without any direct correlation to other points. But In terms of moving points always each point is identified with its ID and it has direct relation to other points, so that some points generate a trajectory. So all the related points of this trajectory have same ID and they are connected together. We can not see such relations in events data. So the nature of these two data is different from perspective of analysis. However as like as event data we also are able to use different distance function according to the goal of analysis or in progressive clustering according to the stage of analysis. in[3] some distance functions like similar routes , similar destinations, similar source, similar route and destinations, similar directions are mentioned and are employed in time of the analysis. For example in similar destination or similar source as depicted in figure 5-a problem is transferred to a single points clustering. Because we ignore the whole trajectories points and we just take their start or end points. In case of similar routes as shown in figure 5-c we need to compare whole trajectories. So the problem of comparison is comparing of two polylines. The determination of selected point per se is a complicated problem as will be discussed more in section 3-3. However a direct method for such comparison is computing the Euclidian distance between the corresponding points.



a) Common Destination

b) Common Source and Destination

c) Common Route

Figure 5 – Similarity calculation of two trajectories based on different goals a) common destinations b) common sources and destinations c) common routes

Concerning the distance functions like similar source and destination (figure 5-b) we need to discover some groups of trajectories which they are coming from the same root and go to the same destination. These trajectories might be a short direct way from the source to the destination or ones which pass zigzag path to reach to the destination. In all such distance functions we need to modify our density based algorithm to adapt the clustering task with our desired goal. For example in [7] a new algorithm namely TRJ-DBSCAN is proposed to find trajectories within *e* of given object at a fixed time *t* which employs route similarity distance function.

3.2 Importing time to data and define a new threshold

In this strategy, a new time dimension is added to the 2-d vector of spatial data [15] and then problem will be transformed to clustering of a 3-d vectors of <x,y,t> which can be performed by both distance-based (e.g. k-mean) and density based algorithms(e.g. DBSCAN or OPTICS). In both algorithms a standard distance measure such as the Euclidean distance (equation 1) will be used as distance measure criteria between two objects <x1,y1,t1> and <x2,y2,t2>.

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2 + (t2 - t1)^2} \qquad (1)$$

3.3 Transform spatio-temporal data to the new objects

In this strategy which in general is using for trajectories, we make a new object from some spatio-temporal data items so that in nature has the both spatial and temporal concept inside of itself. Then we apply clustering algorithms on the new data objects[1,2,3,4,5,6,7,8,10,11]. Also In this case for comparing the new objects we need a new distance function that could be as above a Euclidean distance which is able to compute the distance between two new spatio-temporal objects. For instance, regarding the green and blue points on figure 4 and data items in table 3, according to the spatial location visited by green point during the time between t=1 to t=3 we can transform all data items related to this point to a single object like Trj_Green: (X1,Y1)->(X2,Y2)->(X3,Y3) and likewise for the blue point we have : Trj_Blue: (X8,Y8)->(X9,Y9)->(X10,Y10). So with a Euclidean distance function we can compute the similarity of these two new spatio-temporal objects. sometimes comparing two trajectories by just comparing their spatial locations is not reasonable because it neglects the temporal aspect. Some works like [16] consider the properties of moving objects in road network space and define temporal similarity as well as spatio-temporal similarity between trajectories based on POI (Points of Interest) and TOI (Times of Interest) on road networks. Comparing trajectories is not always easy as mentioned above, in some circumstances it's a difficult task especially when in a real-world problems two trajectories length may completely be different. Also if we consider each point of trajectories, it would be an expensive task for long trajectories. Some trajectory simplification techniques like Douglas-Peucker(DP) algorithm[7,11,17] are used to reduce the trajectory comparison computation costs. The goal of trajectory simplification is transforming a polylines to another polylines with less points. In [7] another two algorithm DP+ and DP* is also presented for increasing of efficiency. Goal of all of these methods is having less points while we are comparing trajectories and therefore less computation.

3.4 Progressive Clustering

Motivation of all works done here [1,2,3,4,5] is doing filtering on the data set to first reduce the computing costs and second get better results according to the specific goals. In some works like [2,3,4] , multiple distance functions and different input parameters are employed in a progressive way so that according to the goal of analytics, suitable inputs and distance function are being used and in some other work like [1] authors are trying to filter the input data with focusing on time periods which cause the clustering result to be same as using whole data set. For instance if the goal is understanding the traffic patterns in a city, instead of applying clustering to whole data set we can first identify which time periods give us the result we are looking for and then filter the data set according to that time periods and then perform clustering. In this case first we are facing with a smaller data set and second the result will be more meaningful because for example the traffic patters in city in weekends is completely different with weekdays and by mixing the both data items the result may not what we like to know. Also in [5] a heuristic method is used for clustering of very large spatio-temporal data set. The idea behind that is this fact that dense regions in data set remain dense also in a sample subset. So if we were able to discover the fundamental frame we can classify other objects to the discovered clusters more easier and cheaper. In order to do this, firstly a proper subset is selected and a density based clustering like OPTICS is applied to that subset, then an analytics do modification and revision on the results, then build a classifier and employ that classifier to allocate each new objects to the obtained clusters.

3.5 Performing clustering task on each time sequences

Sometimes we perform clustering on each time sequences separated, for example in [4] landing events are clustered irrespective of time distance threshold for three years of 2005,2006 and 2007 and then clustering results of these years is useful to find out how pattern of landings has been changed over these years. Also this strategy is employed in the moving clustering problem[6]. Moving clusters are group of objects which enter or leave the cluster during some time intervals but having the portion of common objects higher than predefined threshold. For discovering of moving clusters it needs to perform clustering on each time sequences.

3.6 Thresholds-based clustering

In this strategy, generally spatial and temporal threshold is used for grouping objects and in general no clustering algorithm is involved. Such strategy is used in [9] for discovering of important places from trajectories, every new location is compared to the previous location. If the distance is less than a threshold, the new location is added to the previously created cluster. Otherwise, the new candidate cluster is created with the new location. The candidate cluster becomes a cluster of important places when the time difference between first point in a cluster and the last point is greater than the threshold. In another work like [8], they benefit from this strategy for discovering of flocks. Flocks are group of trajectories that stay together within a specific disk size for the duration of a given time. For example if a disk includes two trajectories in three consecutive sequences it can be a flock with thresholds M=2 and K=3.

3.7 Spatio-temporal pattern discovery

Spatio-temporal patterns which in general is not a clustering problem its more a pattern discovery problem, describe a general spatio-temporal behavior of a group of objects[10,11]. For example as a result of trajectory pattern mining on travelers trajectories we might have patterns like $Sation \xrightarrow{7\min} Downtown \xrightarrow{15\min} beach$ and $Sation \xrightarrow{2\min} Bridge \xrightarrow{8\min} University$ so that the first one implies the travel pattern of tourists and second one could be a travel pattern of students. In order to discover such patterns, four steps is needed[10], first a set of input trajectories over given grid and using spatial neighborhood or radius to allocate the members. Second computing the point of interests from trajectories points by a density based clustering algorithm and then mining the frequent patterns.

4 Conclusion

Several methods and techniques introduced during the paper in terms of spatio-temporal data clustering, however summary of all methods can be summarized in three ways, first those methods who try to transform spatio-temporal data to a condition which can be used in classical clustering algorithms mostly density-based, second trying to change a bit in classic and spatial-suited algorithms to enable algorithm handle the new data type. The third method are related to event type data which both change in algorithm and data can is not sufficient to solve the problem and should be seen separately and is a collection of tasks in progressive ways.

References

1. Nanni M, Pedreschi D (2006) Time-focused clustering of trajectories of moving objects. Journal of Intelligent Information Systems 27(3):267–289
2. Rinzivillo S, Pedreschi D, Nanni M, Giannotti F, Andrienko N, Andrienko G (2008) Visually driven analysis of movement data by progressive clustering. Information Visualization 7(3):225–239
3. Andrienko G, Andrienko N (2008) Spatio-temporal aggregation for visual analysis of movements. In: Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST 2008), IEEE Computer Society Press, pp 51–58
4. Andrienko G, Andrienko N (2009) Interactive cluster analysis of diverse types of spatiotemporal data. ACM SIGKDD Explorations
5. Andrienko G, Andrienko N, Rinzivillo S, NanniM, Pedreschi D, Giannotti F (2009) Interactive Visual Clustering of Large Collections of Trajectories. VAST 2009
6. Kalnis P, Mamoulis N, Bakiras S (2005) On discovering moving clusters in spatio- temporal data. Advances in Spatial and Temporal Databases pp 364–381
7. Jeung H, Yiu ML, Zhou X, Jensen CS, Shen HT (2008) Discovery of convoys in
trajectory databases. Proc VLDB Endow 1(1):1068–1080
8. Vieira MR, Bakalov P, Tsotras VJ (2009) On-line discovery of ?ock patterns in spatio-temporal data. In: GIS ’09: Proceedings of the 17th ACMSIGSPATIAL In- ternational Conference on Advances in Geographic Information Systems, ACM, New York, NY, USA, pp 286–295
9. Kang JH,WelbourneW, Stewart B, Borriello G (2004) Extracting places from traces of locations. In: WMASH ’04: Proceedings of the 2nd ACM international work- shop on Wireless mobile applications and services on WLAN hotspots, ACM,
New York, NY, USA, pp 110–118
10. Giannotti F, Nanni M, Pinelli F, Pedreschi D (2007) Trajectory pattern mining. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge
discovery and data mining, ACM, p 339
11. Kang J, Yong HS (2009) Mining Trajectory Patterns by Incorporating Tempo- ral Properties. Proceedings of the 1st International Conference on Emerging Databases
12. Reades J, Calabrese F, Sevtsuk A, Ratti C (2007) Cellular census: Explorations in urban data collection. IEEE Pervasive Computing 6(3):30–38
13. Birant D, Kut A (2007) St-dbscan: An algorithm for clustering spatial-temporal data. Data Knowl Eng 60(1):208–221
14. Wang M, Wang A, Li A (2006) Mining Spatial-temporal Clusters from Geo- databases. Lecture Notes in Computer Science 4093:263
15. Roberto Trasarti , Mastering the Spatio-Temporal Knowledge Discovery Process, PhD Thesis , University of Pisa Department of Computer Science.
16. J. Hwang, H. Kang, and K. Li. Searching for similar trajectories on road networks using spatio-temporal similarity. In Proc. of the East-European Conference on Advances in Databases and Information Systems, pp. 282--295, 2006.
17. D. Douglas and T. Peucker. Algorithms for the reduction of the number of points required to represent a line or its character. The American Cartographer, 10(42):112–123, 1973

Relevance Ranking for Predicting Web Search Results

Pedro M. Teixeira

Doctoral Program in Informatics Engineering,
Faculdade de Engenharia da Universidade do Porto
pro11007@fe.up.pt

Abstract. In this paper, we focus on how incorporating user behaviour data on a web search engine can significantly improve the ranking accuracy of top results in real web search engine. A large click log dataset and a set of query-url relevance flags previously labelled by juries was used to train and evaluate a classifier, as well as to build a re-ranking alternative based on a cascade click model. Each one of these implicit feedback based ranking methods can improve the precision of a web search ranking algorithms by as much as 17% compared to the its original rankings.

Keywords: Data Mining, Search Engine, Classification, Click Model

1 Introduction

A web search engine is designed to retrieve information to its user through an organized structure, typically an URL list sorted by their relevance. Since the goal of the search engine is to address the user's information needs, it is important to access the effectiveness of the retrieval method. The quality of the results can be assessed through *explicit relevance feedback*, using human assessors to judge the relevance of query-url pairs. However, explicit human ratings are expensive and difficult to obtain and the reason a document meets the information needs of a user group might not be so to another one. Disagreements in relevance judgements are often a reflection of diverse intents being realized with the same query.

Recently, *implicit relevance feedback* has developed into an active area of research, at least in part due to an increase of available resources. Millions of people interact daily with web search engines; large amounts of interaction data are produced every day, providing valuable implicit feedback through the user's search session attributes such as query text, timestamps, localities, click-or-not flags, etc. Given a query, whether user clicks a url is strongly correlated with the user's opinions on the url. However, individual users may behave irrationally or maliciously, or may not even be real users; all of this affects the data that can be gathered [1].

The focus of this research was to explore how implicit measures of user interest (e.g. time data, click through data, query region) could be used to develop a predictive model of binary relevance (relevant or irrelevant) between a query and an url. The click log and relevance labels were provided from a single source, in this case from a russian search engine [9].

As a baseline model, the search engine's url rankings were evaluated using the jury explicit information; then, the url list was resorted using a predictive click model and evaluated with the same jury information as described in Section 4.1. Finally, the implicit data was used as features to train a classifier in order to sort the url list by its relevance, as described in Section 4.2. The results are presented and discussed in Section 5.

2 Background and Related Work

Ranking search results is a fundamental problem in information retrieval. The most common approaches in the context of the web use both the similarity of the query to the page content, and the overall quality of a page. A state-of- the-art search engine may use hundreds of features to describe a candidate page, employing sophisticated algorithms to rank pages based on these features. Current search engines are commonly tuned on human relevance judgments. Human annotators rate a set of pages for a query according to perceived relevance, creating the gold standard against which different ranking algorithms can be evaluated. Reducing the dependence on explicit human judgments by using implicit relevance feedback has been an active topic of research.

Several research groups have evaluated the relationship between implicit measures and user interest. In these studies, both reading time and explicit ratings of interest were collected. Claypool et al. [2] studied how several implicit measures related to the interests of the user. They developed a custom browser called the Curious Browser to gather data, in a computer lab, about implicit interest indicators and to probe for explicit judgments of Web pages visited. Claypool et al. found that the time spent on a page, the amount of scrolling on a page, and the combination of time and scrolling have a strong positive relationship with explicit interest, while individual scrolling methods and mouse-clicks were not correlated with explicit interest.

Implicit feedback such as click data has been also used in various ways: towards the optimization of search engine ranking functions (e.g. [3] [4] [5]), towards the evaluation of different ranking functions (e.g. [6] [7] [8]). Most of the works above rely on a core method: to learn a click model. Basically, the search engine logs a large number of real-time query sessions, along with the user's click-or-not flags; this data is regarded as the training data for the click model, which is used for predicting the click through rate (CTR) of future query sessions. However, clicks are biased with respect to presenting order, reputation of sites, user-side configuration (e.g. display resolution, web browser) [4]. The most substantial evidence is given by the eye-tracking experiment carried out by T. Joachims [8], in which it was observed that users tend to click web documents at the top even if the search results are shown in reverse order.

3 Dataset

The dataset used in this research was gathered from the Yandex [9] search engine. Yandex is the leading internet company in Russia, operating the most popular search engine and the most viewed website. The dataset includes anonymous user sessions extracted from the logs, with queries, URL rankings and clicks [10]. For the purpose of training relevance prediction models, it also includes relevance judgements for the ranked URLs. The logs are about two years old, do not contain queries with detected commercial intent and the user data is fully anonymized.

Some characteristics of the dataset are:

- Unique queries: 30.717.251
- Unique urls: 117.093.258
- Sessions: 43.977.859
- Total records in the log: 340.796.067
- Assessed query-region-url triples for the total query set (training + test): 71.930
- Query-region pairs with assesed urls (training + test): 8.410

The dataset is composed of two distinct entries: the user log, and the relevance labels.

User log The user log represents a stream of user actions, each line representing a query or a click.

A query action is represented in a single line, within the format:

```
SessionID TimePassed TypeOfAction QueryID RegionID ListOfURLs
```

A click action is represented in a single line, within the format:

```
SessionID TimePassed TypeOfAction URLID
```

Each of the previously stated placeholders is defined as follows:

- `SessionID` is the unique identifier of a query session.
- `TimePassed` is the time passed since the start of the session with the `SessionID` in units of time.
- `TypeOfAction` is the type of the action. Its either a query (Q) or a click (C).
- `QueryID` is the unique identifier of a query.
- `RegionID` is the unique identifier of the country the user is querying from. There are 4 possible identifiers (integers from 0 to 3).
- `URLID` is the unique identifier of an URL.
- `ListOfURLs` is the list of `URLID`s ordered from left to right as they were shown to the user from the top to the bottom. Example:

```
10989856    0   Q   10364965 2  671723   21839763  3840421  180513
10989856    103 C   21839763
10989856    955 Q   1009161  2  197515   197539   11   179526   5859272
10989856    960 C   197515
```

Relevance labels The relevance labels consist of labels assigned by Yandex judges to a subset of URLs appearing in the logs. Labels are binary: Relevant (1) and Irrelevant (0). The judgement of relevance was based not only on the text of the query, but also on the region of the user, but not in every case.

A relevance label is represented in a single line, within the format:

```
QueryID RegionID URLID RelevanceLabel
```

Where each placeholder is defined as follows:

- `QueryID` is the unique identifier of a query.
- `RegionID` is the unique identifier of the country the supposed user is querying from.
- `URLID` is the unique identifier of an URL.
- `RelevanceLabel` is the relevance label (0 or 1).
  Example:

```
1209161 2 5839294  1
1209161 2 1912415  1
1209161 2 1621201  1
1209161 2 1111     0
```

4 Methodology

The task we are addressing is the automatic ranking of web search results based on relevance, e.g, given a set of web page URLs $U = \{u_a, u_b, ..., u_n\}$ we want to order it from the most relevant to the less for a given query q_i and a region (user's location) r_i. In order to do so, we used a predictive click model do estimate the relevance between an url and a query and trained a classifier using implicit data as features.

4.1 Using a Click Model for Relevance prediction

Our goal was to model the relationship between clicks and relevance in a way that would allow us to estimate a distribution of relevance from the clicks on an url. This can be done using a click model. Two different types of the click models are position models [11] and the cascade model [12]. A position model assumes that a click depends on both relevance and examination. Each rank has a certain probability of being examined, which decays by rank and depends only on rank. A click on the first url indicates that the url is examined and considered relevant by the user, treating the individual urls in a search result page independently. The cascade model assumes that the user views search results from top to bottom and decides whether to click each url. Once a click is issued, documents below the clicked result are not examined regardless of the position. With the cascade model, each url u, is either clicked with probability r_i (i.e. probability that the url i is relevant) or skipped with probability (1-r_i). The cascade model assumes that a user who clicks never comes back, and a user who skips always continues. Let E_i,C_i be the probabilistic events indicating whether the url i ($1 \leq i \leq M$) is examined and clicked respectively. The cascade model makes the following assumptions:

- $P(E_1) = 1$
- $P(E_{i+1}|E_i = 0) = 0$
- $P(E_{i+1}|E_i = 1, C_i) = 1 - C_i$
- $P(C_{i+1} = 1|E_i = 1) = r_{u_i,q}$, where u_i is the ith url

The cascade model was chosen to our relevance prediction due to its simplicity of implementation and calculation speed given the large size of the dataset. With the number of estimation views for a given url, the perceived relevance can be calculated by

$$\text{Perceived Relevance of url } i = \frac{\text{Number of clicks on url } i}{\text{Number of estimated views on url } i} \tag{1}$$

The term *Number of estimated views on url i* means the number of times an url has been placed above the last clicked url, respecting the cascade click model. It is emphasized in [5] that a click does not necessarily imply the users satisfaction on the content, instead, the user may have been attracted by some misleading abstracts. Therefore, the introduction of $satisfaction$ value is given by

$$\text{Satisfaction of url } i = \frac{\text{Number of last clicks on url } i}{\text{Number of clicks on url } i} \tag{2}$$

The term *last clicks on url i* means the last url that was clicked on a given query session. To depict the actual relevance, rather than using perceived relevance alone, one can use the following formula

$$\text{Actual Relevance of url } i = \text{Perceived Relevance} \times \text{Satisfaction} \tag{3}$$

Once no training phase is required to predict the url relevance using a click model, the full training set and test set were used to compare the click model results against the correspondent relevance labels from the jury.

4.2 Using Supervised Learning for Relevance Classification

The accuracy of click model's *actual relevance* may not directly translate to relevance. We also used a supervised learning approach to tackle this problem. Therefore the major challenge lies in inferring the proper characteristics (features) of a tri-tuple $\{q_i, r_j, u_k\}$ in terms of expressing web search result relevance. Table 1 depicts the features to be used in a classifier based on [4] and [5]. Three classifiers were used in this experiment: Naive Bayes, C4.5 and Nearest-neighbour Classifier. The Weka toolkit [13] was used for the experiments reported in this paper. The labelled jury dataset was splitted into training and an unseen test set. The unseen test set is used to report accuracy measures.

Table 1: Features for Relevance Classification Training

Position	Mean position of the URL in current query, region pair
ClickFrequency	Number of clicks for this query, URL pair
IsNextClicked	$\sum$ clicks on next position
isPreviousClicked	$\sum$ clicks on previous position
TimeOnPage	URL page dwell time
CumulativeTimeOnPage	Cumulative time for all subsequent urls after search
DwellTimeDeviation	Deviation from overall average dwell time on url
CumulativeDeviation	Deviation from average cumulative time on url
PerceivedRelevance	NumberOfClicks / NumberOfEstimatedViews
Satisfaction	NumberOfLastClicks / NumberOfClicks
ActualRelevance	PerceivedRelevance * Satisfaction

4.3 Evaluation metrics

We evaluated the ranking algorithms over a range of accepted information retrieval metrics used in [3], namely Precision at K ($P(K)$) and Normalized Discounted Cumulative Gain (NDCG). These metrics are described below.

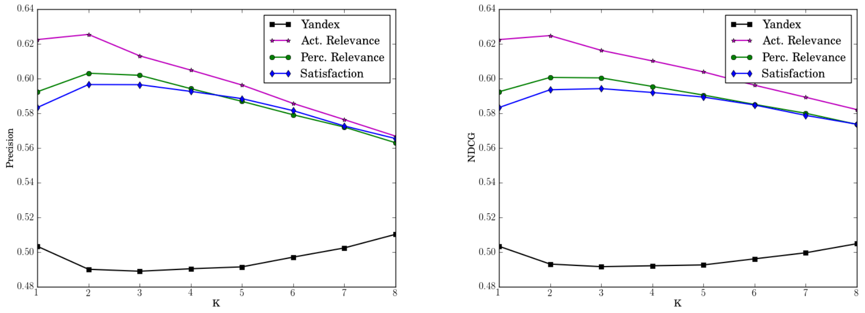
- `Precision at K`: As the most intuitive metric, $P(K)$ reports the fraction of documents ranked in the top K results that are labeled as relevant. In our setting, we require a relevant document to be labelled Good or higher. The position of relevant documents within the top K is irrelevant, and hence this metric measure overall user satisfaction with the top K results.
- `NDCG at K`: NDCG is a retrieval measure devised specifically for web search evaluation [10]. For a given query q, the ranked results are examined from the top ranked down, and the NDCG computed as:

$$N_q = M_q \sum_{j=1}^{K} (2^{r(j)} - 1)/\log(1+j) \tag{4}$$

  Where M_q is a normalization constant calculated so that a perfect ordering would obtain NDCG of 1; and each $r(j)$ is an integer relevance label (0="irrelevant" and 1="relevant") of result returned at position j. Note that unlabelled documents do not contribute to the sum, but will reduce NDCG for the query pushing down the relevant labeled documents, reducing their contributions. NDCG is well suited to web search evaluation, as it rewards relevant documents in the top ranked results more heavily than those ranked lower.

5 Results and Discussion

We compared our methods over the search engine's baseline rankings using the performance metrics NDCG and Precision at K. The performance of the click model was directly evaluated using the full dataset, once no training dataset was required. In order to evaluate the performance of the machine learning classifiers, a stratified sample of 44% of the full dataset was made to build the testing dataset. The test and training datasets are disjoint sets. We then drill down to examine the effects on re-ranking for the attempted queries in more detail, analysing where implicit feedback proved most beneficial. We first experimented with different methods (Perceived Relevance, Satisfaction and Actual Relevance) of re-ranking the search engine baseline outputs. Figures 1(a) and 1(b) report Precision and NDCG for the search engine baseline rankings, as well as for the click model re-ranking strategies results with user feedback. The improvement is consistent across the top 8 results and largest for the top results: NDCG at $K = 2$ for *Actual Relevance* is 0.624 compared to 0.493 of the original results, and precision at $K = 2$ similarly increases from 0.490 to 0.625.



(a) Precision at K

(b) NDCG at K

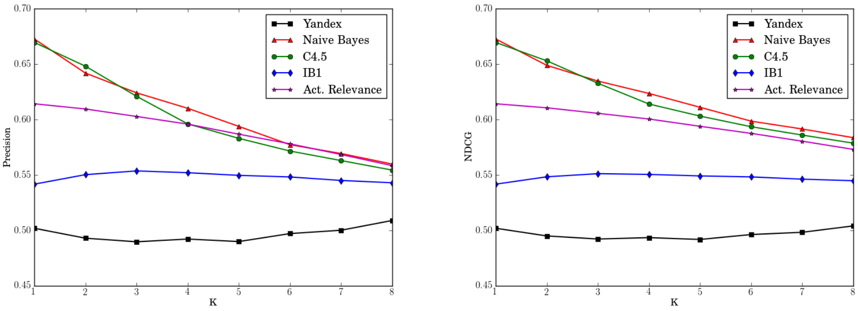
Fig. 1: Precision and NDCG at K for Yandex original rankings, rankings sorted by Perceived Relevance, Satisfaction and Actual Relevance.

Table 2: Evaluation results of the machine learning classifiers

Classifier	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Naive Bayes	0.47	0.435	0.64	0.47	0.336	0.61
C4.5	0.596	0.417	0.594	0.596	0.595	0.626
Nearest-neighbour classifier	0.55	0.462	0.549	0.55	0.549	0.544

On Table 2 is presented the detailed accuracy of the classifiers. The Naive Bayes classifier produces a better precision of 0.64 compared with 0.594 and

0.549 of the C4.5 and Nearest-neighbour classifier respectively. Interestingly, using clickthrough alone, while giving significant benefit over the original search engine ranking, is not as effective as considering the full set of features in Table 1 used to train the classifiers. Figure 2(a) and 2(b) reports Precision and NDCG at K for the different classifiers and also for the Actual Relevance from the click model.



(a) Precision at K

(b) NDCG at K

Fig. 2: Precision and NDCG at K for Yandex original rankings, rankings sorted by Actual Relevance and machine learning classifiers: Naive Bayes, C4.5 and Nearest-neighbour.

Our experimental results, incorporating implicit feedback, resulted in significant improvements over the original rankings, using both click model and machine learning classifiers. The set of implicit features, such time region indicator, provides advantages over using click data alone as an indicator of interest. Furthermore, incorporating implicit feedback data as features into the learned ranking function is more effective than using implicit feedback for re-ranking. These results are in accordance with the results reported in [3].

6 Conclusion

In this paper we explored the utility of incorporating implicit feedback obtained from Yandex search engine to improve its web search ranking. We performed a large-scale evaluation over the click log, using 71.930 manually labelled queries, establishing the utility of incorporating implicit feedback to improve web search relevance. We compared two alternatives of incorporating implicit feedback into the search process, namely re-ranking with implicit feedback and incorporating implicit feedback features directly into the trained ranking function on a machine learning classifier. Our experiments showed significant improvement over methods that do not consider implicit feedback, gaining improvements as high as 17%. Our experiments showed that implicit user feedback can further improve web search performance, when incorporated directly with popular content- and link-based features.

References

1. Zhu, Z., Chen, W., Minka, T., & Zhu, C. (2010). A novel click model and its applications to online advertising. Proceedings of the third, 321-330.
2. M. Claypool, D. Brown, P. Lee and M. Waseda. Inferring user interest, in IEEE Internet Computing. 2001
3. Agichtein, E., Brill, E., and Dumais, S. Improving web search ranking by incorporating user behavior information. In SIGIR 2006.
4. Agichtein, E., Brill, E., Dumais, S., and Ragno, R. Learning User Interaction Models for Predicting Web Search Result Preferences. In SIGIR 2006.
5. Chapelle, O. and Zhang, Y. A Dynamic Bayesian Network Click Model for Web Search Ranking. In WWW 2009.
6. Carterette, B. and Jones, R. Evaluating search engines by modeling the relationship between relevance and clicks. In NIPS 2008.
7. Joachims, T. Evaluating retrieval performance using clickthrough data. In SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval 2002.
8. Joachims, T., Granka, L., Pan, B., Hembrooke, H., and Gay, G. Accurately Interpreting Clickthrough Data as Implicit Feedback. In SIGIR 2005.
9. Yandex Company `http://company.yandex.com/`
10. Yandex: Internet Mathematics 2011 `http://imat-relpred.yandex.ru/en/datasets`
11. G. Dupret and B. Piwowarski. User browsing model to predict search engine click data from past observations. In SIGIR 08: Proceedings of the 31st Annual International Conference on Research and Development in Information Retrieval, 2008.
12. N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In WSDM 08: Proceedings of the international conference on Web search and web data mining, pages 8794. ACM, 2008
13. Weka Toolkit(`http://www.cs.waikato.ac.nz/~ml/weka/`)

M6: a method for compressing complete genomes using Markov models

Diogo Pratas and Armando J. Pinho

Signal Processing Lab, DETI / IEETA,
University of Aveiro, 3810–193 Aveiro, Portugal
pratas,ap@ua.pt

Abstract. Recent advances in DNA sequencing technology have caused an exponential growth of publicly available genomic sequence data. For this reason, being able to efficiently store sequenced genomes is a problem of paramount importance.

In this paper, it is proposed M6, a method for compressing complete genomes using a mixture of six Markov models with different depths, that explores inter-chromosomal properties.

Preliminary results show that the compression rate, in several of the cases, is improved compared to the currently sate-of-the-art Expert-Model. Moreover, the presented method is substantially faster, motivating further research efforts.

1 Introduction

DNA sequences are made up by a 4-symbol (4-bases) alphabet: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). A complete sequence of DNA from a species can be seen has a genome, which in some cases can have millions of symbols. For example, the human genome is determined by approximately 3 000 million bases [1]. This means that it takes approximately 750 MBytes to represent the human genome. Moreover, publicly available genomic sequence data have grown exponential, motivating the study of DNA data compression algorithms for efficient storage or transmission of the information.

Almost twenty years ago, Grumbach and Tahi [2] proposed the first algorithm dedicated to DNA coding, *Biocompress*. Since this work, several other contributions have been made in the area of DNA data compression, such as [3,4,5,6,7,8,9,10]. Most of these works explore the non-stationary nature of the DNA sequence data, typically using at least two encoding methods, one based on a Lempel-Ziv-like substitutional procedure [11] and another based on a low-order context-based arithmetic coding.

According to the substitutional paradigm, repeated regions of the DNA sequence are represented by a reference to a past occurrence of the repetition and by the length of the repeating sequence. Both exact and approximate repetitions have been explored, as well as their inverted complements. In the case of approximate repetitions, it is also required to indicate where the sequences differ.

The substitutional approach is usually the main encoding method, with the low-order Markov model assuming the role of a fall-back, secondary choice. When the substitutional method is unable to provide satisfactory performance, the corresponding region of the DNA sequence is represented by a low-order Markov model. This scheme for representing DNA data has been significantly improved by Tabus and by Korodi et al., based on the normalized maximum likelihood (NML) algorithm [12,7,9], and by Cao *et al.* [10], using the state-of-the-art expert-model (XM).

The most recent version of the NML-based approach [9] is an evolution of the normalized maximum likelihood model introduced in [12] and improved in [7]. This new version, NML-1, aims at finding the best regress block, in other words, an approximate repetition, using first-order dependencies.

The other method, XM [10], relies on a mixture of experts for providing symbol by symbol probability estimates, which are then used for driving an arithmetic encoder. The algorithm comprises three types of experts:

- (1) order-2 Markov models;
- (2) order-1 context Markov models, i.e., Markov models that use statistical information only of a recent past (typically, the 512 previous symbols);
- (3) the copy expert, that considers the next symbol as part of a copied region from a particular offset. The probability estimates provided by the set of experts are then combined using Bayesian averaging and sent to the arithmetic encoder.

Recent work has shown that the ability of Markov modelling to represent DNA data sequences seems to go beyond a simple secondary role [13,14]. In [14] we have addressed the problem of DNA sequence compression using two Markov models competing for encoding the data on a block basis. The idea was to use a low-order model for modelling high information content regions and a high-order model for those regions having low entropy. The method described in [14] is forward-adaptive, therefore requiring side information, in this case for indicating which of the two models encodes the block.

In this paper, it is addressed the problem of representation of complete genomes using exclusively a combination of six Markov models. To investigate this matter, it is used a method based on a mixture of Markov models. Each of the Markov models has a different depth, exploring different contexts associated with the nature of the data. Moreover, it is used concatenation between chromosomes in order to explore inter-chromosomal relations [15], consequently providing better compression gains. The method is tested in a set of genomes, in order to compare its performance with the state-of-the-art XM encoder. The results show that the proposed approach not only provides better compression, but it is also considerably faster.

This paper is organized as follows. In Section 2, it is described the materials and methods, namely the DNA sequences used, the mixture of the Markov models and the chosen Markov parameters. In Section 3, it is provided experimental results using eight genomes. Finally, in Section 4, some conclusions are drawn.

2 Materials and methods

2.1 DNA sequences

In this study, it has been used eight genomes obtained from the National Center of Biotechnology Information (`ftp://ftp.ncbi.nlm.nih.gov/genomes/`), as follows:

- *Home sapiens*, Build 33;
- *Mus musculus*, MGSCv37 Build 37;
- *Escherichia coli*, 536 uid58531;
- *Pseudomonas aeruginosa*, LESB58 uid59275;
- *Schizosaccharomyces pombe*, uid127;
- *Candida albicans*, uid14005;
- *Haloquadratum walsbyi, DSM 16790 uid58673*;
- *Arabidopsis thaliana*, 408:796-815;

The genomes have been compressed in a Linux server, running 16 Intel(R) Xeon(R) CPU E7320 @ 2.13GHz, with 256 GB of RAM.

2.2 Mixture of Markov models

A Markov model of an information source assigns probability estimates to the symbols of the alphabet, according to a conditioning context computed over a finite and fixed number, $k > 0$, of past outcomes $x_{n-k+1..n} = x_{n-k+1} \dots x_n$ (order-k Markov model). Table 1 shows an example of how statistical data are usually collected in Markov modelling. In this example, an order-5 Markov model.

Table 1. Simple example illustrating how statistical data are typically collected in Markov models. Each row of the table represents a probability model at a given instant n. In this example, the particular model that is chosen for encoding a symbol depends on the last five processed symbols (order-5 context).

Context, c	n^c_{A}	n^c_{C}	n^c_{G}	n^c_{T}	$n^c = \sum_{a \in \mathcal{A}} n^c_a$
AAAAA	23	41	3	12	79
⋮	⋮	⋮	⋮	⋮	⋮
ATAGA	16	6	21	15	58
⋮	⋮	⋮	⋮	⋮	⋮
GTCTA	19	30	0	4	53
⋮	⋮	⋮	⋮	⋮	⋮
TTTTT	8	2	18	11	39

In practice, the probability that the next outcome x_{n+1} is $s \in \mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$, is obtained using the estimator

$$P(s|x_{n-k+1..n}) = \frac{C(s|x_{n-k+1..n}) + \alpha}{C(x_{n-k+1..n}) + 4\alpha}, \tag{1}$$

where $C(s|x_{n-k+1..n})$ represents the number of times that, in the past, symbol s was found having $x_{n-k+1..n}$ as the conditioning context, and where

$$C(x_{n-k+1..n}) = \sum_{a \in \mathcal{A}} C(a|x_{n-k+1..n}) \tag{2}$$

is the total number of events that has occurred so far in association with context $x_{n-k+1..n}$. The per symbol information content average provided by the Markov model of order-k, after having processed n symbols, is given by

$$H_{k,n} = -\frac{1}{n} \sum_{i=0}^{n-1} \log_2 P(x_{i+1}|x_{i-k+1..i}) \quad \text{bpb}, \tag{3}$$

where "bpb" stands for bits per base. When using several models simultaneously, the $H_{k,n}$ can be viewed as measures of the performance of those models until that position. An example of two models used simultaneously can be seen if Figure 1.

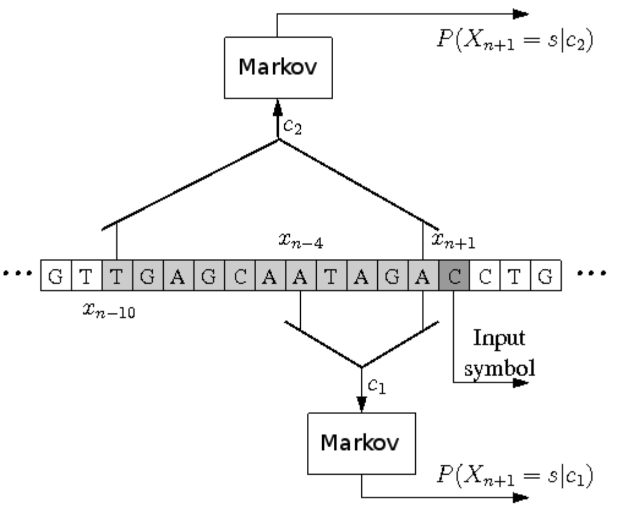


Fig. 1. Example of an overall model for estimating probabilities using multiple Markov models, in this case two. The probability of the next outcome, X_{n+1}, is conditioned by the k_1 or k_2 last outcomes, depending on the eights of the Markov model chosen for handling that particular DNA base. In this example, $k_1 = 5$ and $k_2 = 11$.

Therefore, the probability estimate can be given by a weighted average of the probabilities provided by each model, according to

$$P(x_{n+1}) = \sum_k P(x_{n+1}|x_{n-k+1..n})\, w_{k,n}, \tag{4}$$

where $w_{k,n}$ denotes the weight assigned to model k and

$$\sum_k w_{k,n} = 1. \tag{5}$$

For stationary sources, we could compute weights such that

$$w_{k,n} = P(k|x_{1..n}), \tag{6}$$

i.e., according to the probability that model k has generated the sequence until that point. In that case, we would get

$$w_{k,n} = P(k|x_{1..n}) \propto P(x_{1..n}|k)P(k), \tag{7}$$

where $P(x_{1..n}|k)$ denotes the likelihood of sequence $x_{1..n}$ being generated by model k and $P(k)$ denotes the prior probability of model k. Since the DNA sequences are not stationary, a good performance of a model in a certain region of the sequence might not be attained in other regions. Hence, it is used a mechanism of progressive forgetting of past measures.

$$p_{k,n} = p^{\gamma}_{k,n-1}P(x_n|k, x_{1..n-1}) \tag{8}$$

and, finally,

$$w_{k,n} = \frac{p_{k,n}}{\sum_k p_{k,n}}. \tag{9}$$

2.3 Sequence concatenation

Markov modelling provides probability estimates that depend on the recent past of the sequences. Generally, bigger sequences provide better statistics, consequently providing more accurate models. Normally, the human genome (like other eukaryote organisms) is compressed chromosome by chromosome, which prevents the model from exploring inter-chromosome correlations[15]. In order to explore the advantage of using these models in more than one chromosome at the same time, we compressed chromosome 1 concatenated with chromosome 2 and compared the compression ratio with the average resulting from compressing the chromosomes individually. Table 2 shows the results.

In this case, the average rate without concatenation would stand for 1.7300 bpb. Although, with concatenation, the ratio value is 1.7263 bpb, indicating that there is an advantage of using concatenation in Markov models. Moreover, since this method is used in more sequences (in the case of the human genome there are 24 chromosomes) and it is included more mixture models (six Markov models), the concatenation technique consistently provide better results (less time spent and better compression ratio).

Table 2. Compressing results regarding chromosome 1, chromosome 2 and a concatenation of chromosomes 1 and 2. It has been used two mixture models (order-4 and order-16) taking advantage of the complement inverted repeats.

Sequence	Rate (bpb)	Time (sec.)	Length (Mb)
Chromosome 1	1.7201	648.21	218.71
Chromosome 2	1.7388	695.48	237.04
Concatenated	1.7263	1332.29	455.75

2.4 Markov parameters

We recall that, due to the non-stationary characteristic of the DNA data, the multiple Markov models should combine, at least, a low order model and a high order one. Accordingly, the low order that was choosen was 1 (minimum order of the implementation). For the high order category, an order-16 model seems to be the natural candidate (maximum order of the implementation).

On the other hand, it has been used an order-4 model, because to compress, for example, the human genome with competing models, model order-4 seems to have the best compression ratio in the low order category [16].

Moreover, using the knowledge from the competing Markov models [16], it has been selected order-14. Therefore, for reasons of uniform model order distribution the other assumed orders were 6 and 10. In short, the six order models used are: 1, 4, 6, 10, 14 and 16.

The probabilities associated to the Markov models were estimated using (1), with $\alpha = 1$ (corresponding to Laplace's estimator) for model orders $k = 1, 4, 6, 10$ and with $\alpha = 0.05$ for model orders $k = 14, 16$. The value of α is not too much important for low-order models, but it is crucial in high-order ones. In the latter case, the number of times that a given context occurs is generally small, rendering the estimation of the probability strongly dependent of α. Using $\alpha = 0.05$ would provide, globally, good results. However, this motivates further research efforts to determine the best α in different orders and sequences.

3 Results and discussion

In the previous section, we have presented a method for compressing complete genomes using a mixture of Markov models. On the other hand, we have realized that concatenated sequences achieved better compression results. Therefore, all chromosomes from the genomes, were concatenated, and addressed as a single sequence (*H. sapiens*, *M. musculus*, *S. pombe*, *A. thaliana*).

In this section, we present results for the compressed chromosomes. Table 3 shows that the compression rate in *S. pombe*, *E. coli* and *P. aeruginosa* is improved compared to the currently sate-of-the-art Expert-Model. Moreover, the presented method is substantially faster, motivating further research efforts.

Another observation shows that the proposed method is better than the XM-50 in the *H. sapiens* genome and 2.7 times faster than the state-of-the-art.

Table 3. Compression results of eight genomes using different approaches. The M6 contain the results provided by the mixture of the six Markov models. The XM-50 and XM-200 columns show the results obtained with the XM algorithm, using 50 and 200 experts. The "s" and "m" symbols represent respectively seconds and minutes. The results in bold show the best result for compression ratio and time for each genome.

Genome	Size	XM–50		XM–200		M6	
	(Mb)	Ratio	Time	Ratio	Time	Ratio	Time
H. sapiens	2928	1.644	1035 m	**1.618**	1780 m	1.632	**388 m**
M. musculus	2560	1.560	989 m	**1.541**	1617 m	1.570	**361 m**
A. thaliana	119	1.736	23 m	**1.730**	53 m	1.769	**14 m**
S. pombe	13	1.876	102.8 s	1.876	145.1 s	**1.865**	**71 s**
P. aeruginosa	6.3	1.841	71 s	1.839	174 s	**1.785**	**36 s**
E. coli	4.9	1.912	44 s	1.911	55 s	**1.898**	**27 s**
H. walsbyi	3	1.903	21.8 s	**1.903**	23.4 s	1.914	**17.6 s**
C. albicans	1	1.843	5.5 s	**1.843**	6.4 s	1.850	**5.4 s**

Every data compression method assumes a certain model of the information source that produces the data. When we improve a data compression method, we are also improving the model of the source. This happens because, when the probability distribution of the assumed source model is closer to the true probability distribution of the source, a smaller relative entropy results and, therefore, fewer redundancy bits are required. This is why the importance of data compression goes beyond the usual goal of reducing the storage space or the transmission time of the information. In fact, in some situations, seeking better models is the main aim. Since this method seems to be much faster than the state-of-the-art, it is a natural candidate to explore biological properties on genomes, using techniques such as complexity profiles [17].

4 Conclusions

We have presented M6, a method for compressing complete genomes using a mixture of six Markov models with different depths. Using a concatenation technique the model was able to explore inter-chromosomal properties and by consequence to reduce the ratio of compression.

In general, we were able to compress complete genomes with values that are competitive with the stat-of-the-art XM method and that require much less computation time.

Taking into account the results that we report in this paper, we can say, perhaps somewhat surprisingly, that complete genomes can be quite well described using M6, i.e., by a mixture of six Markov models that rely on short-term knowledge of the past.

In future work we intent to improve M6 using a Markov model with threshold (also used in XM) and perform more tests in more genomes. Finally, we intent to use M6 in biological analysis such as using complexity profiles.

References

1. Rowen, L., Mahairas, G., Hood, L.: Sequencing the human genome. Science **278** (October 1997) 605–607
2. Grumbach, S., Tahi, F.: Compression of DNA sequences. In: Proc. of the Data Compression Conf., DCC-93, Snowbird, Utah (1993) 340–350
3. Rivals, E., Delahaye, J.P., Dauchet, M., Delgrange, O.: A guaranteed compression scheme for repetitive DNA sequences. In: Proc. of the Data Compression Conf., DCC-96, Snowbird, Utah (1996) 453
4. Loewenstern, D., Yianilos, P.N.: Significantly lower entropy estimates for natural DNA sequences. In: Proc. of the Data Compression Conf., DCC-97, Snowbird, Utah (March 1997) 151–160
5. Chen, X., Kwong, S., Li, M.: A compression algorithm for DNA sequences. IEEE Engineering in Medicine and Biology Magazine **20** (2001) 61–66
6. Manzini, G., Rastero, M.: A simple and fast DNA compressor. Software—Practice and Experience **34** (2004) 1397–1411
7. Korodi, G., Tabus, I.: An efficient normalized maximum likelihood algorithm for DNA sequence compression. ACM Trans. on Information Systems **23**(1) (January 2005) 3–34
8. Behzadi, B., Le Fessant, F.: DNA compression challenge revisited. In: Combinatorial Pattern Matching: Proc. of CPM-2005. Volume 3537 of LNCS., Jeju Island, Korea, Springer-Verlag (June 2005) 190–200
9. Korodi, G., Tabus, I.: Normalized maximum likelihood model of order-1 for the compression of DNA sequences. In: Proc. of the Data Compression Conf., DCC-2007, Snowbird, Utah (March 2007) 33–42
10. Cao, M.D., Dix, T.I., Allison, L., Mears, C.: A simple statistical algorithm for biological sequence compression. In: Proc. of the Data Compression Conf., DCC-2007, Snowbird, Utah (March 2007) 43–52
11. Ziv, J., Lempel, A.: A universal algorithm for sequential data compression. IEEE Trans. on Information Theory **23** (1977) 337–343
12. Tabus, I., Korodi, G., Rissanen, J.: DNA sequence compression using the normalized maximum likelihood model for discrete regression. In: Proc. of the Data Compression Conf., DCC-2003, Snowbird, Utah (2003) 253–262
13. Pinho, A.J., Neves, A.J.R., Ferreira, P.J.S.G.: Inverted-repeats-aware finite-context models for DNA coding. In: Proc. of the 16th European Signal Processing Conf., EUSIPCO-2008, Lausanne, Switzerland (August 2008)
14. Pinho, A.J., Neves, A.J.R., Bastos, C.A.C., Ferreira, P.J.S.G.: DNA coding using finite-context models and arithmetic coding. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP-2009, Taipei, Taiwan (April 2009)
15. Botta, M., Haider, S., Leung, I., Lio, P., Mozziconacci, J.: Intra- and interchromosomal interactions correlate with CTCF binding genome wide. Molecular Systems Biology **6** (November 2010)
16. Pratas, D., Pinho, A.J.: Compressing the human genome using exclusively Markov models. In: Advances in Intelligent and Soft Computing, Proc. of the 5th Int. Conf. on Practical Applications of Computational Biology & Bioinformatics, PACBB 2011. Volume 93. (April 2011) 213–220
17. Dix, T.I., Powell, D.R., Allison, L., Bernal, J., Jaeger, S., Stern, L.: Comparative analysis of long DNA sequences by per element information content using different contexts. BMC Bioinformatics **8**(Suppl. 2) (2007) S10

User Clustering in Smartphone Applications

Klaus Schaefers

Doctoral Program in Informatics Engineering,
Faculdade de Engenharia da Universidade do Porto,
pro11010@fe.up.pt
http://fe.up.pt

Abstract. Successful user interface design requires a deep understanding of the targeted audience. This paper explores the application of the K-Means algorithm to cluster smartphone users, in order to offer Human Computer Interaction (HCI) specialists a better insight into their user group. Different hierarchical ordered feature space representations are explored and their performance is evaluated against a synthetic data set. By applying the best performing features against a real world data set, obtained from a public available smartphone application, three distinct user groups could be identified.

Keywords: Data Mining, Usage Mining, Cluster Analysis, HCI, Smartphone

1 Introduction

Designing user interfaces is a challenging task. The success and acceptance of the user interface depends crucially on the HCI specialists understanding of the target group and their needs. Wrong expectation of the target group often lead to severe shortcomings in terms of usability and a low market success. The main objective of this work is it to explore different clustering techniques and their applicability in the context of smartphone applications. Common design processes like user centered design pay special attention to the detailed description of the different user types and their needs. The users stereotypes are in these processes presented as so called personas, each having a background story and a detailed description of their requirements and skills. The precise construction of these personas is a very cumbersome and expensive task, as it requires for instance a large number of interviews [7, 15]. However it is not always possible to get access to members of the target group. This is in particular a challenge for web or smartphone applications, that might be targeted at a very brought and heterogeneous, global audience. In the context of web applications it is therefore common to monitor the user interaction and to perform data mining in the gathered data, to better understand the targeted user group.

Among techniques like pattern mining, particularly clustering techniques have received much attention, as they provide the necessary information to partition the audience into different persona like groups[14, 18]. The knowledge about these user clusters is also a key factor for successful user interface design in the context of smartphone applications, especially if no in depth use case analysis has been carried out before [19]. For example the target user group could be separated into distinct sub sets, each utilizing only parts of the application features. In such a case it might make sense to offer each group a different user interface, focusing on the required features.

In section one the paper introduces the related work, before the experimental setup is presented and the different features spaces are discusses in detail. Afterwards the results will be presented and discussed. The paper ends with a short presentation of future work.

2 Related Work

The problem of user clustering is a field that has been intensively researched in the context of web usage mining, for instance to provide a customized shopping experience or to help search engines in scoring the retrieved documents based in the users interests [14]. Most web mining systems are based on web server log files, from which the user sessions are extracted. Each session consists out of a set of URLs which have been visited and the timestamp of the request [21, 6, 5].

In the current literature the most common approaches try to capture the session data in a vector space model, researching different features set and algorithms. For instance the session data might be transformed into a binary vector model, where each component indicates the presence of the absence of a certain URL in the session [11]. Others have used the page frequency instead as the main feature. Yan et al. for instance use the frequency to construct the feature space and apply the Leader algorithm to perform the clustering [20]. Antonellis et al. use also the page frequency, but deploy the K-mean algorithm [8] instead to determine the clusters [2]. However these two algorithms have in common the need to specify the number of cluster k in advance. Different methods have been suggested to overcome this issue. One approach is the so called "elbow method", which calculates the sum of squared errors *SSE* for each k, and tries to detect a sudden decrease between k and $k + 1$. Another approach is to calculate the Silhouette coefficient, s, or the Davies Bouldin coefficient, d, for each k, and choose the k with the best value of s or d [16]. We may refer the work of Kuo et al. who explored a new method for determining k. They applied an artificial neural network (ANN) to decide the optimal number of clusters k in advance, before using K-Means [11]. Later Kuo et al. also used an hierarchical algorithm instead of K-Means to solve this issue [9]. Due to the potentially large number of web pages, the feature spaces might suffer from a high dimensionality, which causes problems during clustering. This phenomenon is also know as the "curse of dimensionality", and leads to difficulties in the visualization of the clustering results and a poor performance of certain similarity measures [10, 4]. As a solution Fu et al. introduce the concept of a generalized session, which uses attribute - oriented - induction to reduce the number of dimensions. Afterwards they used the BIRCH algorithm [22] to perform a hierarchical clustering [21]. Beside the

URL of the requested page, the temporal dimension of the user interaction is seen as a valuable sign for the personal interest of a user regarding a certain page. Generally it is assumed that a long duration of a page view indicates a strong interest [18, 1], anyhow it is to point out that the duration might differ significantly between users and might be prone to noise. [3, 12].

Although web based applications share certain similarities with smartphone applications, there are some significant differences. Web applications consist normally out of a large number of pages, where the user interaction can be described by the transition from page to page, whereas smartphone applications are only composed out of a few pages. Most of the interaction takes place within the pages, e.g. a click on a button. Therefore the obtained usage data must be more fine grained then the web server log files, and must capture each interaction event.

3 Data Sets

The user interaction can be divided into distinct atomic interaction events. Each interaction event was triggered by the invocation of a widget w_i which is element of the set of all widgets W in the application. Each widget w_i belongs one page p_i, which is an element of the set of all pages P. Each event has a certain type y_i, which is an element of the set of all event types Y that can occur in the application. Typical examples are an *onclick* event which occurs in case a button was pressed, or an *onchange* event if the value of a text field was changed. Moreover each event occurred at one point of time $t_i \in \mathbb{N}$. An event is assigned to a session s_i, which starts and ends with the application and is element of S, the set of all session. Each session is assigned to an element u_i of the set of all users U. Let E be the entire set of events

$$E = (e_1, e_2, .., e_n) \text{ with } e_i \in W \times Y \times P \times \mathbb{N} \times S \times U \tag{1}$$

Each interaction event e_i be can described by the tuple

$$e_i = (e_{iw}, e_{iy}, e_{ip}, e_{in}, e_{is}, e_{iu}) \tag{2}$$

with

$$e_{iw} \in W, e_{iy} \in Y, e_{ip} \in Y, e_{in} \in \mathbb{N}, e_{is} \in S, e_{iu} \in U \tag{3}$$

Due to the lack of a publicily available data set, a custom Android application was created in the scope of this work. The application offers the users the possibility to play a simple memory game, in which they can test there knowledge in certain domains. In addition the users can select between the two levels of difficulty easy and hard. The application implements two main use cases. The first use cases targets new and inexperienced users. These users will face the start page after launching the application and can start an easy memory game directly pressing a large “Start” button. After finishing the game, users have the choice between playing the same game again or moving on to another domain.

Furthermore the users can view their high scores or navigate back to the "Welcome" page. The second use case extends the first user case and targets at more experiences users, that are already familiar with the application. These users can navigate from the "Welcome" page to several configuration pages, to customize the domain of the game or to change level of difficulty from easy to hard. In case the difficulty is set to hard, the users will face a more complex game field which offers more options, making it harder to solve the game in short time. The basic navigation structure of the application is can be seen in Figure 1

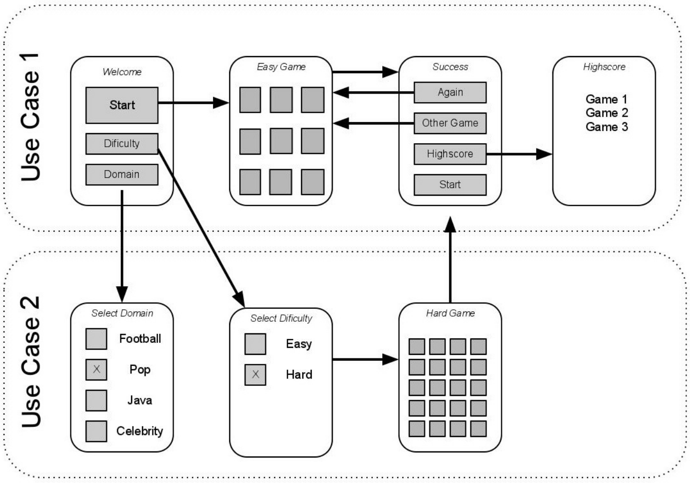


Fig. 1. Conceptual model of the real world applicaton.

The application was published in the Android Market and all interaction events were monitored and transmitted in regular intervals to a central server. After a period of three weeks 41 users had installed the application, and more then 6500 events were submitted, belonging to 68 sessions.

4 Feature Spaces

The clustering was performed using nine different normalized feature spaces, in which each user is presented as an instance. Before the different presentations where calculated, the event stream was filtered to only include *onclick* events. This step is mandatory as a magnitude of events was logged in the real android application. For instance changing the value of a text field, would result in one *onclick*, one *onfocus*, one *onchange* and one *onblur* event. As result certain type of widgets, for instance text fields, would have a significant higher event frequency then other widget, e.g. buttons, and would hence distort the result.

The features spaces utilize on one hand the event frequency to describe the user behavior, and on the other hand use the time stamps to model the temporal dimension of the user interaction. The temporal dimension can reveal the users interest in a certain page [18], but can also be seen as a hint regarding the level of experience the user has archived so far with the application. New users will have a longer latency between the interaction events, as they have to orient themself in the new application, whereas experienced users will know the application, and hence have a lower latency [19].

Following the natural hierarchy of modern smart phone user interfaces, the features space form a hierarchy as well. A smartphone application consists out of n pages, each containing m widgets. Thus it is possible to calculate aggregated values for different dimensions, starting at the widget level, over to the page level and ending on the application level.

The first feature space V_1 can been seen as a application level summary of the entire interaction. It contains the number of session $s(u_i) \rightarrow \mathbb{N}$, the number of visited pages $p(u_i) \rightarrow \mathbb{N}$, the number of invoked widgets $w(u_i) \rightarrow \mathbb{N}$, the mean session duration $m(u_i) \rightarrow \mathbb{R}$ and the mean latency $l(u_i) \rightarrow \mathbb{R}$ between interaction events. Hence $v_{1i} \in V_1$ for one user $u_i \in U$ is

$$v_{1i} = \{s(u_i), p(u_i), w(u_i), m(u_i), l(u_i)\} \tag{4}$$

Feature space V_2 targets the page level, and can be considered a kind of "drill down", offering more details about the user interaction. It is composed out of the page frequencies $pf(u_i, p_j) \rightarrow \mathbb{R}$, this means $v_{2i} \in V_2$ contains the relative amount of events the were observed in each page $p_j \in P$ for a given user $u_i \in U$.

$$v_{2i} = \{pf(u_i, p_1), pf(u_i, p_2), ..., pf(u_i, p_n)\} \tag{5}$$

The temporal dimension of the user interaction is considered in the third feature space V_3. Instead of the event frequency, $v_{3i} \in V_3$ for a user $u_i \in U$ will contain the mean time $pt(u_i, p_j) \rightarrow \mathbb{R}$ u_i spend on each page $p : j \in P$.

$$v_{3i} = \{pt(u_i, p_1), pt(u_i, p_2), ..., pt(u_i, p_n)\} \tag{6}$$

V_4 is the union of V_2 and V_3. It is composed out of the page frequency and the mean time users spend on a page, and tries to capture the temporal characteristics and as well the preferred pages for a user $u_i \in U$. v_{4i} is defined as:

$$v_{4i} = \{pf(u_i, p_1), pt(u_i, p_1), pf(u_i, p_2), pt(u_i, p_2), ..., pf(u_i, p_n), pt(u_i, p_n)\} \tag{7}$$

The following three feature spaces will explore similar feature space representations, but on the widget level. $v_{5i} \in V_5$ is composed out of the frequencies $wf(u_i, w_i) \rightarrow \mathbb{R}$ of a widget $w_j \in W$ for each user $u_i \in U$.

$$v_{5i} = \{wf(u_i, p_1), wf(u_i, p_2), ..., wf(u_i, p_n)\} \tag{8}$$

V_6 presents the mean widget latency $wl(u_i, w_j) \rightarrow \mathbb{R}$ for a given user $u_i \in U$. $v_{6i} \in V_6$ is therefore defined as

$$v_{6i} = \{wl(u_i, w_1), wl(u_i, w_2), ..., wl(u_i, w_n)\} \tag{9}$$

The union of V_5 and V_6 is V_7. Like V_4 it tries to model the temporal characteristics for a user $u_i \in U$ and his preferred widgets.

$$v_{7i} = \{wf(u_i, w_1), wl(u_i, w_1), wf(u_i, w_2), wl(u_i, w_2), ..., wf(u_i, w_n), wl(u_i, w_n)\} \tag{10}$$

Moreover two additional feature spaces will be investigated on the widget level. v_8 is inspired by the popular *tf-idf* measure form the field of natural language processing [13]. The *tf-idf* measure gives a higher importance to the term occurrence in a document, if only a small set of other documents contain the term as well. As a result it is possible to recognize characteristic terms for a each document. Each widget w_j is considered the equivalent to a term and the set of all events $E_i \subset E$ of an user u_i is the considered the equivalent of a document. By doing so, $v_8 \in V_8$ will have a higher value for the characteristic widgets per user. $v_8 \in V_8$ is defined as

$$v_{8i} = \{tfidf(u_i, w_1), tfidf(u_i, w_2), ..., tfidf(u_i, w_n)\} \tag{11}$$

where

$$tfidf(u_i, w_j) = \frac{|(e_k \in E_i : e_{kw} = w_j)|}{|E_i|} \times log(\frac{|U|}{|x : w_j \in E_j, j \in 1, ..|U||}) \tag{12}$$

V_9 is a binary feature space which is composed out of the k most frequent widgets. Let T_{ki} be the set of the k most frequent used widgets of a given user u_i. Then $tw(u_i, w_j) \in \{0, 1\}$ is a boolean function which returns 1 iff widget $w_j \in T_{ki}$ and 0 otherwise. For this paper $k = 10$ was found to give good results. $v_{9i} \in V_9$ can be defined as

$$v_{9i} = \{tw(u_i, w_1), tw(u_i, w_2), ..., tw(u_i, w_n)\} \tag{13}$$

5 User clustering

The K-Means algorithm, implemented in the R toolkit [17], was used to find the clusters within the data sets. This paper will use the “elbow” method, the Silhouette coefficient and the Davies Bouldin coefficient for $k \in \{2..15\}$ to determine the best k. The maximum number of 15 was chosen, since a higher number of clusters seems to be unreasonable for a population of only 41 users.

The different feature representations might result in different values for an optimal k. To ensure the general feasibility of the taken approach, four synthetic data sets were created. These data sets were generated for a prototypical application which consists out of two pages, each containing a number of widgets. The widgets can be grouped into four distinct sets A,B,C and D, each presenting one application feature. To model different kind of users, for instance beginners and professionals, several simulated users have been created, each interacting differently with the application. The main variables of these simulated users are the average number and length (in events) of sessions and the average latency between interaction events. Each of the variables is modeled through a Gaussian distribution, with a default standard deviation per parameter. The number of sessions and the session length can be used to simulate the personal interest of a user. The latency related parameters allow the simulation of the different levels of experience. Additionally the users are parametrized with a weighted set of application features, they use most of the times. An application feature is modeled through the set of widgets A, B, C, and D. By utilizing different combinations of all these parameters the following four synthetic data set were created:

Data Set 1: This data set simulates two very heterogeneous groups of users and consists out of 200 users. The first group only utilizes feature A and B , has a short session length and in general a low number of sessions per user. The second group utilizes feature C and D, has a medium session length and a larger number of sessions.

Data Set 2: The data set consists out of 300 users, which can are divided into three heterogeneous groups. Each group has a medium session length, a small number of sessions and invokes the features A, B and C, but with a significant different weight. The first group focuses on feature A and has a low latency between the events. The second group on prefers feature B and shows a medium latency and the thirst group invokes feature C mostly with a large latency.

Data Set 3: This data set simulates experienced and inexperienced users and consists out of 200 users. The inexperienced group only utilizes feature A, has a medium latency but only short number of sessions and session events. The experienced group uses features A and B, has a short latency and a large number of session and a medium session length.

Data Set 4: This data set model four set of users, which are partly overlapping. The data set consists out of 400 users. Two groups have a small number and length of sessions and a medium latency, whereas the two remaining groups show a higher value for the session length and number of sessions and in addition have a lower latency. Each group utilizes three of the four features, with a different weight, and each group has a different latency.

The expected results for each data set and each feature space are summarized in Table 1.

6 Results and Conclusions

To ensure that the random seed of the K-Means algorithm does not influence the results, the clustering was performed 100 times for all synthetic data sets,

Table 1. Expected best k for synthetic data sets per feature space

Feature Space	Data Set 1	Data Set 2	Data Set 3	Data Set 4
V_1	2	3	2	2
V_2	2	3	2	3
V_3	2	3	2	2
V_4	2	3	2	2
V_5	2	3	2	4
V_6	2	3	2	2
V_7	2	3	2	4
V_8	2	3	2	4
V_9	2	3	2	4

and the mean values for the "elbow" method, the Silhouette coefficient and the Davies Bouldin coefficient are shown in Table 2.

Before the result will be discussed in detail, some general issues have to be explained. It can be seen, that the Davies Bouldin coefficient returned in several cases, which are marked with * in Table 2, a value of 15. For the further analysis these values will not be taken into consideration, and will be treated as outliers. In some cases the values for the best and the second best k were quite close for the Davies Bouldin coefficient. In such a case the second best k is noted behind the best k in parenthesis. Moreover the "elbow" method was not conclusive in four cases, which are marked with $**$, because the "elbow" could not be clearly localized. This happened for data set 2 and 4 and the feature set V_5, V_8 and V_9, anyhow it is to mention that the value for k was close to the expected value, therefore a value range is given for the k. For the real world application the "elbow" method turned out to be not useful in most of the cases, as the curve rendered by the SSE / k pairs did not show any "elbow". These result are denoted with a "-".

When comparing the outcome of the analysis of the synthetic data sets with the expected values for the data sets it can be seen, that V_1 performed generally well. All methods return the expected k value, the only exception the "elbow" value for data set 2. This is remarkable, because Data Set 3 was constructed to have three clearly separated user groups, but might be explained with the fact that the number of sessions and number of events per session was equal for all three groups. V_9 performed also well, if the outlier value of 15 for the Davies Bouldin in data set 4 is not considered. The "elbow" method anyhow does not deliver a non-ambiguous result, and should therefore be interpreted with care in the context of V_9. V_5 seemed to perform well in the more simply data sets 1,2 and 3, where the user groups are clearly separated and there is only a small overlap in the invoked features. The temporal dimension of the user interaction seems to be well captured by V_3 on the page level, whereas the result in the widget level V_6 show only a good result for the "elbow" method. The combined feature spaces V_4 and V_7 do not show satisfying results, and should therefore not be considered in the analysis of the real world application.

Table 2. Best k for all data sets per feature space and evaluation method

Feature Space	Method	Data Set 1	Data Set 2	Data Set 3	Data Set 4	Real Data
V_1	Elbow	2	5	2	2	4
	Silhouette	2	3	2	2	2
	Davies Bouldin	2	3	2	2	4
V_2	Elbow	2	2	2	2	6
	Silhouette	2	2	2	3	2
	Davies Bouldin	2	2	2	3	6
V_3	Elbow	2	3	2	2	-
	Silhouette	4	3	2	2	3
	Davies Bouldin	15 *	3	2	2	3
V_4	Elbow	2	3	2	2	-
	Silhouette	4	3	2	2	3
	Davies Bouldin	15 *	3	2	2	3
V_5	Elbow	2	3-4 **	2	5	-
	Silhouette	2	3	2	5	2
	Davies Bouldin	2	3	2	3	2
V_6	Elbow	2	3	2	2	-
	Silhouette	2	2	2	3	2
	Davies Bouldin	2	2	2	2	6 (2)
V_7	Elbow	2	4	2	3	-
	Silhouette	2	2	2	2	2
	Davies Bouldin	2	2	2	2	6 (2)
V_8	Elbow	2	3-4 **	2	5	6
	Silhouette	2	2	2	3	2
	Davies Bouldin	2	2	2	3	2
V_9	Elbow	2	3-4 **	2	4-5 **	-
	Silhouette	2	3	2	4	2
	Davies Bouldin	2	3	15 *	4	15

The real word usage data shows problems regarding the application of the "elbow" method. The reason for this might be the low number of users in the data set, which makes it hard to find large and consistent clusters. Only in V_1 this method returns a clear result of 4. The Davis Bouldin method indicates 4 as well, only the Silhouette coefficient returns 2. The widget frequency represented by V_5 has a non ambiguous result of 2 .V_9, which performed well on the synthetic data set, confirms the value of $k = 2$, but like in the data set 4 the Davies Bouldin index returns $k = 15$. The screen latency which is model by V_3 returns the value 3 for k for the Silhouette and the Davies Bouldin method. Table 3 shows the cluster sizes, the Silhouette and the Davies Bouldin coefficient for the most promising features spaces and the corresponding best k.

The resulting clusters of V_1 and V_3 have both one cluster which only contains a single user. As V_1 and V_3 both contain the the number of invoked pages, we might conclude that this user distinguishes himself only through the num-

Table 3. Cluster sizes and cluster quality

Feature Space	k	Cluster Sizes	Silhouette	Davies Bouldin
V_1	4	25, 1, 8,7	0.71	0.39
V_3	3	33, 7, 1	0.65	0.69
v_5	2	35, 6	0.57	1.00

ber of pages he has used. Furthermore all clusters contain one cluster which is significantly larger then the remaining ones, which can be seen as an indicator, that the entire set of users can be split into one main group and several smaller groups.

Figure 2 shows the plot of the mean latency $l(u_i)$ against the number of invoked widgets $w(u_i)$ per user. The size of the circles presents the number of sessions per users. The figure reveals, like the cluster analysis before, the existence of a larger group which utilizes only a subset of the application widgets and which has a low latency. In contrast to this group, two smaller subsets can be identified. One group with a larger latency, which only uses parts of the widgets and another group with a larger number of widgets and a a low latency. It can be also seen, that this group has in general a larger number of sessions, which is in indication for an particular interest in the application.

This observations are reasonable in the context of the real world application, since the application offers one main use case for new users, and one extended use case for more experienced users. The correlation between the larger number of sessions and the low latency seems to be sound as well, as users gather more experience in the application, and can therefore full fill there goals faster.

7 Conclusion and Future Work

In this work nine different feature spaces were subject to investigation, regarding their feasibility to cluster smartphone usage data. The feature spaces were evaluated against four synthetic data sets. A features space summarising the user interaction in only 5 dimensions showed the best results. However, to capture the temporal dimension of the user interaction, the mean latency per screen was found to form a good feature space. The widget frequency performed also well, but focuses on the most frequent used functions of the application.

The major problem during the elaboration of the results was the lack of a sufficient data set. To overcome this issue, a custom smartphone application was developed to harvest real world usage data. Within three weeks, the data of 41 users could be collected. After applying the best performing feature spaces in conjunction with the K-Means algorithm, three user groups could be identified. The discovered clusters show good values for the Silhouette and Davies Bouldin coefficient, and fit the use case structure of the developed application.

Nonetheless the collected data set was quite small, which made it challenging to draw valid conclusions. Therefore one of the next goals should be the

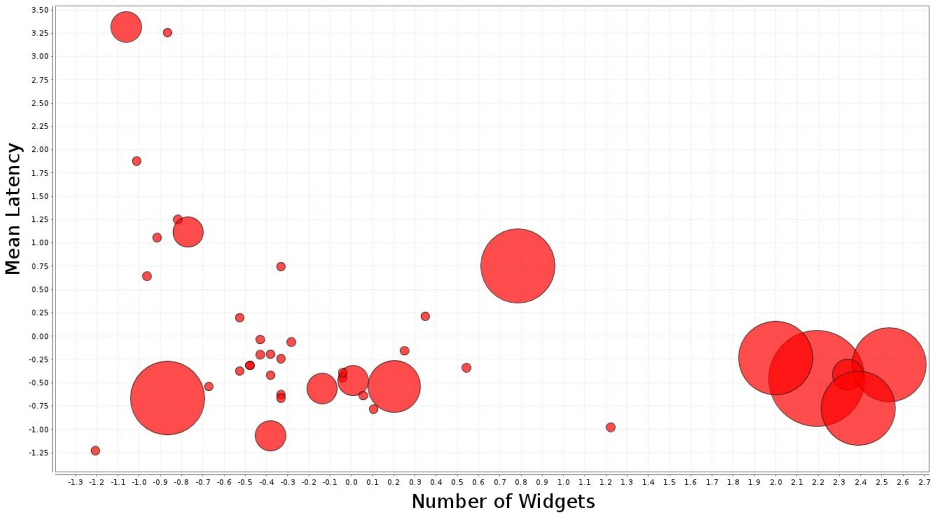


Fig. 2. Mean latency $l(u_i)$ against the number of invoked widgets $w(u_i)$. The size of the circle indicates the number of sessions $s(u_i)$.

acquisition of a larger data set. Once this data set is available other algorithms should be subject to further research, for instance hierarchical or density based algorithms. Furthermore new ways of cluster visualization should be explored, as it might be hard to interpret the results, especially in the context of high dimensional feature spaces, like those working on the widget level.

References

1. Shafiq Alam, Gillian Dobbie, and Patricia Riddle. Particle Swarm Optimization Based Clustering of Web Usage Data. In *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 451–454. IEEE, December 2008.
2. Panagiotis Antonellis, Christos Makris, and Nikos Tsirakis. Algorithms for clustering clickstream data. *Information Processing Letters*, 109(8):381–385, March 2009.
3. Arindam Banerjee and Joydeep Ghosh. Clickstream Clustering Using Weighted Longest Common Subsequences. *Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining*, May 2001.
4. Richard Ernest Bellman. *Dynamic programming.* Princeton University Press, 1957.
5. José Borges, Mark Levene, Brij Masand, and Myra Spiliopoulou. *Web Usage Analysis and User Profiling*, volume 1836 of *Lecture Notes in Computer Science.* Springer Berlin Heidelberg, Berlin, Heidelberg, June 2000.
6. R. Cooley, B. Mobasher, and J. Srivastava. Web mining: information and pattern discovery on the World Wide Web. In *Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence*, pages 558–567. IEEE Comput. Soc, November 1997.

7. Alan Dix, Janet E. Finlay, Gregory D. Abowd, and Russell Beale. *Human-Computer Interaction (3rd Edition)*. Prentice Hall, 2003.
8. Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, 2005.
9. Gang Kou and Chunwei Lou. Multiple factor hierarchical clustering algorithm forlarge scale web page and search engine clickstream data. *Annals of Operations Research*, pages 1–12, February 2010.
10. Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data. *ACM Transactions on Knowledge Discovery from Data*, 3(1):1–58, March 2009.
11. R.J. Kuo, J.L. Liao, and C. Tu. Integration of ART2 neural network and genetic K-means algorithm for analyzing Web browsing paths in electronic commerce. *Decision Support Systems*, 40(2):355–374, August 2005.
12. Pawan Lingras and Chad West. Interval Set Clustering of Web Users with Rough K-Means. *Journal of Intelligent Information Systems*, 23(1):5–16, July 2004.
13. Christopher D. Manning. *Foundations of Statistical Natural Language Processing [Gebundene Ausgabe]*. Mit Press, 1999.
14. Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic personalization based on Web usage mining. *Communications of the ACM*, 43(8):142–151, August 2000.
15. Jakob Nielsen. *Usability Engineering [Taschenbuch]*. Morgan Kaufmann; Auflage: New edition, 1994.
16. Tan Pang-Ning, Steinbach Michael, and Kumar Vipin. *Introduction to Data Mining [Hardcover]*. Addison Wesley; 1 edition, 2005.
17. RProject. The R Project for Statistical Computing, 2011.
18. Jaideep Srivastava, Robert Cooley, Mukund Deshpande, P.N. Tan, Birgit Hay, Geert Wets, Koen Vanhoof, Alan Dix, Janet E. Finlay, Gregory D. Abowd, and Russell Beale. Web usage mining: Discovery and applications of usage patterns from web data. In *Proc. Intelligent Techniques for Web Personalization: 17th Int. Joint Conf. Artificial Intelligence*, volume 1, pages 1–6. Citeseer, 2000.
19. Thomas Tullis and William Albert. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Morgan Kaufmann, 2008.
20. Tak Woon Yan, Matthew Jacobsen, Hector Garcia-Molina, and Umeshwar Dayala. From user access patterns to dynamic hypertext linking. *Computer Networks and ISDN Systems*, 28(7-11):1007–1014, May 1996.
21. Kanwalpreet Sandhu Yongjian Fu. Clustering of Web Users Based on Access Patterns. In *In Proceedings of the 1999 KDD Workshop on Web Mining*. Springer-Verlag, 1999.
22. Tian Zhang, Raghu Ramakrishnan, and Miron Livny. BIRCH: A New Data Clustering Algorithm and Its Applications, June 1997.

Session VI

Multimedia & Image Processing

Chairman: João Silva, Diogo Pratas

A Comprehensive Taxonomy for Three-dimensional Displays
Waldir Pimenta

Automatic Visual Speech Animation
José Serra, João Freitas, Miguel Dias and Verónica Orvalho

Classification, Implicit Segmentation and a Chronological Prediction Model for Cinematic Sound
Pedro Silva

On the architecture of a real-time vision system for a SPL humanoid robot
Alina Trifan, António Neves, Nuno Lau and Bernardo Cunha

Surface modelling and prototyping using a touch interface
Tiago Marques

Motion Capture Fundamentals - A Critical and Comparative Analysis on Real World Applications
Pedro Nogueira

An Enhanced Indoor Positioning Technique For Ubiquitous Computing Applications
Samih Eisa

A Comprehensive Taxonomy for Three-dimensional Displays

Waldir Pimenta[1]

Departamento de Informática
Universidade do Minho
Braga, Portugal
waldir@email.com

Abstract. Even though three-dimensional (3D) displays have been introduced in relatively recent times in the context of display technology, they have undergone a rapid evolution, to the point that equipment able to reproduce three-dimensional, dynamic scenes in real time are now becoming commonplace in the consumer market.
This paper presents an approach to (1) provide a clear definition of a 3D display, based on implementation of visual depth cues, and (2) specify a hierarchy of types and subtypes of 3D displays, based on a set of properties that allow an unambiguous classification scheme for three-dimensional displays.
Based on this approach, three main classes of 3D display systems are proposed –screen-based, head-mounted displays and volumetric displays– along with corresponding subtypes, aiming to provide a field map for the area of three-dimensional displays that is clear, comprehensive and expandable.

Keywords three-dimensional displays, depth cues, 3D vision, survey

1 Introduction

The human ability for abstraction, and the strong dependence on visual information in the human brain's perception of the external world, have led to the emergence of visual representations (that is, virtual copies) of objects, scenery and concepts, since pre-historical times. Throughout the centuries, many techniques have been developed to increase the realism of these copies.

At first, this focused on improving their visual appearance and durability, through better pigments, perspective theory, increasingly complex tools to assist the recreation process, and even mechanical ways to perform this recreation (initially with heavy manual intervention but gradually becoming more automatic).

As these techniques matured, there was a gradual shift to center efforts in providing these virtual representations with more lifelike features, by making them dynamic, and audible:[1] television, cinema, stereo sound and surround sound are straightforward examples of this.

Finally, recent years have revealed yet another shift, this time aiming at ways to recreate the sensation of depth, or three-dimensionality. 3D displays thus emerged as an active area of research and development.

Despite this being a relatively recent field, many different approaches have been already proposed and implemented as 3D displays, and new ones surface with some regularity. Moreover, these implementations provide different sets of approximations for the depth cues that our visual system uses to perceive the three-dimensionality of a scene.

This profusion of implementations has plagued attempts to define a nomenclature system for 3D displays. Despite many taxonomies having been proposed, a definitive, exhaustive and unambiguous categorization system for 3D displays has been lacking in the literature, which hinders the classification and evaluation of different implementations, especially hybrid ones.

The persistent hype around 3D displays shows that there is a great interest in this technology from public, media and industry. Volumetric displays are a common staple in movies (though usually called "holograms"), and currently various techniques have already reached mainstream use, most notably through 3D gaming, cinema and television (for instance, in 2005 already about half the world's IMAX cinemas could project stereoscopic movies with the viewers using either Polaroid or shutter glasses. [1]). Even web browsers are attempting a democratization of 3D through WebGL, while the popular video sharing website Youtube has enabled tools for stereo video.

The current hype, however, also means that the distinction between pseudo-3D and actual 3D displays, and among different 3D displays, is severely blurred and obscured by marketing and promotion claims. This hides various fundamental flaws that these 3D display systems contain, which consist mainly in their inability to provide all natural depth cues available with natural vision. An unambiguous and clearly-defined taxonomy would thus be helpful to both the scientific community and to the industry.

The approach presented in this paper bases the definition and categorization of 3D displays in a thorough understanding of their fundamental properties and functional characteristics, rather than in implementation details, as is the case with most current classifications.

For this, we present, in Section 2, a comprehensive overview of the depth cues used by human visual system to perceive three-dimensionality. With this knowledge, we can then, in Section 3, determine the specific subset of these that clearly mark the frontier between 2D and 3D displays, and define basic properties of 3D displays. Section 4 then goes into detail inside the 3D display realm, defining a hierarchy of classes and sub-classes for 3D displays, based on

[1] Attempts to stimulate other senses have been tried as early as in the 1960s, with machines such as the Sensorama[5]. However, such devices (termed "multi-modal") haven't achieved the same commercial success as audiovisual technology.

the depth cues they implement, the physical properties of the system, and the usage type.

By taking all these characteristics into account, we expect the outcome to be a logical, systematic and extensible taxonomy that will facilitate comparison of different approaches, and the evaluation of appropriate techniques for a given application. Section 5 assesses the degree to which this objective was fulfilled, and illuminates what further work is to be performed to enhance the proposed taxonomy.

2 Visual Cues to Three-Dimensionality

The origins of the Human species, from primates living and moving around primarily in trees, made perception of depth a very important feature of our vision. As artistic depictions of reality developed, several techniques were devised to simulate the three-dimensionality of the original scenes being recreated. Therefore, even media that are usually considered 2D often reproduce some of the cues that our visual system uses to interpret the location of objects. These hints, commonly called **depth cues**, can be divided into two main groups: *psychological* cues, which depend on our previous knowledge of the visual aspect of familiar objects, and *physiological* cues, which manifest through the anatomy of our visual system [7].

The psychological depth cues are:

Occlusion the overlap of some objects by others that are closer to us. Some imaging techniques take this further by making the screen itself transparent, such as the mirror-based Pepper's ghost illusion, and the more recent glass-based projections popular in entertainment shows (often deceptively marketed as "holograms").

Linear perspective given prior knowledge of shapes and/or sizes of objects, we interpret perceived distortions in their shape (parts farther away from us appear smaller), differences in size between objects, and variation of the angular size (how much of our visual field they cover) as indicators to their location in three-dimensional space.

Atmospheric perspective commonly known as "distance fog", it refers to the fading in contrast and detail, and shift to bluish colors, of objects located at a great distance. This happens because the light we get from them had to travel an increased distance through air and thus underwent more scattering from the atmospheric gases and particles.

Shading and shadow projection effects caused by the relationship between objects and light sources. The brightness, color and patterns seen in an object's surface provide information about (among other things) its shape and position relative to the light sources that illuminate it. Also, the location, format and intensity of shadows projected into the object (due to parts of it or other objects obscuring the light) and into its vicinity allow us to interpret its 3D form and relative position to other objects and/or the environment.

The above are all static cues. There are two more psychological cues, which are dynamic; that is, they manifest when there is movement either of the observer or of the observed object (or both):

Motion parallax [2] relative changes in perceived position between two objects when we move. For example, during a car trip a tree seems to be "travelling past us" faster than the distant mountains.

The kinetic depth effect changes in the appearance of an object due to its own motion. For example, when a spherical object –say, a football– is uniformly illuminated so that no shadows give away its round shape, a slow rotation around itself is sufficient for our visual system to infer that it is a solid body and not a flat disk facing us, due to the relative motions of features in its surface.

The physiological depth cues consist of:

Accommodation the effort made by the muscles in the eye that control the shape of its lens in order to bring the image into focus in the retina. Even though we usually do not consciously control these actions, our brain uses this information as an indicator of the distance of objects we are observing. Since the focusing effort varies much more for distance changes near the eye, the effect is particularly notable for nearby objects (less than 2m, according to [6]).

Convergence when both eyes rotate inwards to aim at the object of interest, thus aligning the different images they receive, so they can be more effectively combined by the brain. As with accommodation, this rotation manifests itself with greater amplitude when differences in distance occur closer to the eye, so it is also a cue that is more strongly perceived for nearby objects (less than 10m, according to [9]). If they are close enough, one can clearly feel the eyes "crossing" so that they can keep aiming at the same point.

Binocular disparity (or **stereo parallax**)[3] differences in images received by each eye. [4] Studies indicate that for a moderate viewing distance, binocular disparity is the dominant depth cue to produce depth sensation, through a process called stereopsis, which is the effort made by the brain to fuse the images together into a 3D perception of the scene. This effort is always necessary because convergence of the eyes only produces perfectly aligned points in the line perpendicular to the eye-to-eye distance (this line is called a horopter).

The depth cues described above are summarized in Table 1.

[2] from the Greek *paralaxis* (change).

[3] "Binocular" comes from the Latin *bini* (pair) + *oculus* (eye). "Stereo" comes from the Greek *stereós* (solid).

[4] This cue has been known and studied for a long time: Euclid in 300 B.C. was the first recorded scientist to suggest that depth perception in human vision is related to the fact that we have two eyes which collect different simultaneous perspectives of the same object [2].

Table 1. Summary of visual depth cues for three-dimensional vision

	static	**dynamic**
psychological	occlusion (overlap); linear perspective; atmospheric perspective (distance fog); shading.	motion parallax; kinetic depth effect.
physiological	accommodation (focus); binocular disparity (stereo parallax); convergence.	

2.1 The Accommodation-Convergence Mismatch

Although most representational media reproduced in 2D displays implement the static psychological cues (some, like video, even implement the dynamic ones, albeit partially – only due to scene movement, but unresponsive to viewer movement), the fact that the physiological cues are not implemented isn't a serious problem, because either the scenes represented are meant to take place (or be viewed from) a distance where these cues aren't relevant (especially convergence and accommodation), or because we can cognitively ignore the lack of cues as our abstraction ability allows us to understand their 3-dimensionality regardless.

But when there is a mismatch in physiological cues, we may feel actual physical discomfort. Namely, every display that provides stereoscopy (one view for each eye) is theoretically able to implement proper convergence cues for each object in the scene depending on their location. But accommodation (proper focus of the light, that is, the ability to make the light rays diverge not from the screen, but from the virtual positions of the scene objects) is much harder to achieve; as a result, most of these displays force the eye to always focus at the screen to get a sharp image. Therefore, if the scene is placed at a virtual distance within the range of operation of the accommodation depth cue, we suffer a phenomenon called accommodation-convergence (A-C) mismatch.

This mismatch is more serious than the aforementioned ones, because it provides the brain with conflicting physical sensations, which causes discomfort, as the brain tries to make sense of conflicting data. This effect is similar to the way mismatch between visual and vestibular (from our balance system in the inner ear) perception of movement causes motion sickness. The effect is thus very physical and may cause headaches, fatigue or disequilibrium [10], preventing continued use of these displays. This, of course, in addition to the reduction it causes in the realism of the 3D visualization.

During ocular accommodation (focusing), the amount of contraction of the muscles that control the shape of the eye lens is sent to the brain as information about the distance of the object being observed. This usually occurs paired with the convergence, which also sends muscular contraction information to the brain as an indicator of the proximity of the object. A mismatch between these two sources of information causes discomfort, as the brain tries to make sense of conflicting data, just like mismatch between visual and vestibular perception of movement causes motion sickness. This mismatch happens with conventional stereo systems, since the eyes do rotate to line-up the different images each receive, but otherwise keep the accommodation constant, since to get a sharp image they need to focus at the screen, which does not move.

2.2 Ranges of Operation

All the psychological depth cues can be reproduced by traditional flat media, such as paintings, photographs or movies. Therefore, if we are in restricted situations that prevent us from applying the physiological cues, such media can be enough to produce a realistic depiction of a three-dimensional scene. These situations usually manifest if the scene is distant enough from the observer, out of the range of operation of most depth cues.

Firstly, accommodation stops giving useful feedback at distances over 2m (the eyes essentially relax the muscles that control the lens, and focus on infinity) and convergence is similarly virtually unaffected for objects at a distance of over 10m, when the light “rays” coming from them to each eye have such a small angle between them that they can be assumed to be parallel.

Our eye separation distance also cannot provide any perceived binocular disparity for great distances: even though we still get different images for each eye at distances of over 1 km [8], the minimum depth separation between objects in the scene (relative distances from the observer) becomes increasingly greater if we are to interpret relative depth between objects in the scene without other cues such as occlusion or atmospheric perspective.

Finally, if the scene is static, even motion parallax (which we can provoke by moving our heads, and as such has a greater range than binocular disparity) cannot provide cues to the relative positions of objects at great distances, since the motion of our head produces a fairly uniform displacement in all parts of the perceived image.

This was the reason why in the past some believed that the sky was a painted dome placed well above our heads – visually, there was no way to tell otherwise.

2.3 Demand for 3D Displays

Apart from the situations described above, in most cases the human visual system will easily detect the illusion of 2D reproductions, which prevents realistic 3D visualization to occur. However, this has never posed a serious problem either for informational content or for art and entertainment, both due to the several depth cues that can be represented in traditional “2D” media, and to the high degree of abstraction achievable by the human intellect.

Nevertheless, since imaging devices started to appear, expectations of realism in 3D have constantly grown, as demonstrated by the great diversity and fantastic features of such fictional devices devised in science fiction. Recent advances in computer graphics, display technology and data transfer rates led to not only the manifestation of increasingly realistic 3D experiences that make such expectations to become more plausible, but also to the exhaustion of the ability to display complex sets of three-dimensional data, when this is the best approach for visualization and/or manipulation of complex information that overcome our abstraction abilities [4].

When there is a need to display dynamic three-dimensional data, conventional 2D displays lack the ability to convey true three-dimensional perception, even though (and despite their name) they do support several 3D depth cues (occlusion, perspective, apparent size, “distance fog”, focus, etc.). Realistic 3D perception, however, is only achieved by providing further depth cues, motion parallax and binocular disparity being the most common. As such, many devices capable of reproducing also the physiological depth cues have started to become common in many fields. Below we make an overview of these 3D displays.

3 Definition of a 3D Display

The line separating 3D displays from 2D displays is not always clearly defined, despite what the 2D/3D dichotomy seems to suggest. In one hand, the static psychological 3D depth cues can, in fact, be reproduced in media traditionally considered “2D”; on the other hand, most so-called “3D” displays are actually flat screens (which means that the images are projected in a 2-dimensional surface), with the notable exception of volumetric displays.

With these limitations in mind, **we define 3D displays as any devices that are able to reproduce the dynamic psychological depth cues** (motion parallax and kinetic depth) **and the physiological ones** (stereoscopy, accommodation and convergence).

The Projection Constraint

Another important characteristic of 3D displays (of all displays, in fact) is the universality of the **“projection constraint”**, which means that the image can never exceed the boundaries of the display medium as these are perceived by the observer. This property prevents free-viewing of the scene, limiting both the **look-around** freedom (the observer cannot look away from the display) and the **move-around** freedom (the observer cannot circle the virtual object to see it from all sides). These limitations manifest in different ways and degrees for different displays, but are present for virtually all of them.

This constraint on the virtual viewing area can be compared to the way a window only allows a glimpse of the scene outside. [4] describes the projection constraint in the terms that “a display medium or element must exist in the line of sight between the viewer and all parts of the image.”

It is worth noting that this does not apply only to planar displays: volumetric displays also can only display images inside their volume. Figure 1 makes this principle clear.

Countering this effect may be done by increasing the absolute size of the display (for example, a cinema screen), shaping it in order to surround the viewer (as is done in the CAVE virtual reality environment), or increasing its relative size by bringing it closer to the observer (such as in head-mounted displays). Of

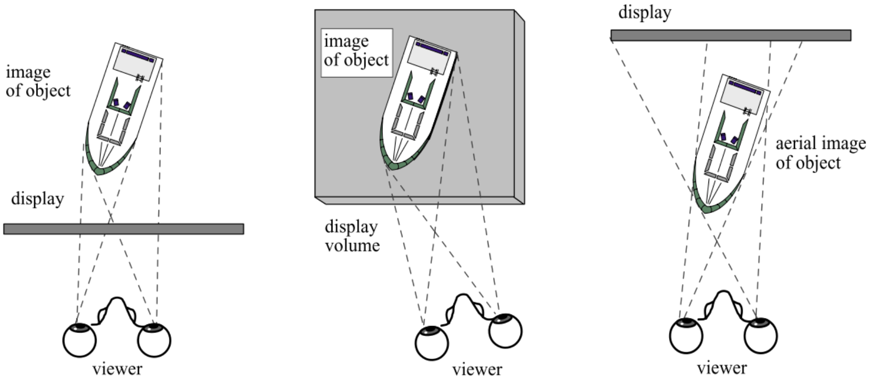


Fig. 1. Illustration of the projection constraint. (Reprinted from [4]).

course, the consequences in terms of the ability of the system to support multiple simultaneous users must be considered for each of these approaches.

4 Proposed Taxonomy for Imaging Techniques

The first 3D display, in the sense described in Section 3, was the stereoscope, invented by Sir Charles Wheatstone in 1840. The binocular-like device allowed viewing of a double-photograph in a way that produced unprecedented depth sensation. The images were taken with a special camera with two lenses with approximately the same displacement as the average interpupillary distance of human eyes. When placed in the device, the pictures were viewed through a setup of prisms and lenses, which helped approximate the conditions of the real scene, rather than the actual conditions of photographs placed immediately in front of the eyes.

Since then, many other devices were invented as mechanisms for 3D viewing. The proliferation of such devices and techniques, especially hybrid approaches, and common misconceptions and misnomers spread by under-researched science-fiction or over-enthusiastic marketing departments, led to some confusion in the definitions used today, even (though to a lesser degree) in the scientific literature.

To sistematize the review of 3D displays, we will classify them into three main types:

1. screen-based displays
2. head-mounted displays (HMDs)
3. volumetric displays

4.1 Screen-based Displays

Screen-based 3D displays are the most popular 3D displays used currently, with commercial use now common in movie theaters and domestic displays. They work mostly by providing stereoscopy (different images for each eye), which, as mentioned in Section 2, is the main depth cue for 3D vision at moderate distances. These displays can be further divided in two main groups: **stereoscopic** displays, which work in conjunction with glasses, and **autostereoscopic** devices, that don't require any headgear.

The glasses-based 3D displays can display one image to each eye by combining separate streams of images in one device (i.e, multiplexing of light). This can be done in three ways:

wavelength multiplexing separating the left-eye and right-eye images in different colors, the most well-known example of which is the anaglyph, with its characteristic "red-green" glasses;

temporal multiplexing using shutter glasses synchronized with the screen and a doubled frame-rate that displays the images for the left and right eye alternatively;

polarization multiplexing the most common glass-based 3D technology currently in use, achieved by emitting images for each eye with different light polarizations, and filtering them with polarized-filter glasses.

Autostereoscopic devices include parallax barrier displays, lenticular screens and holography. Parallax barriers work by interlacing both images together in alternated vertical strips, and employing a fence-like barrier to block the light from the left strips from reaching the right eye, and vice-versa. Lenticular displays do this filtering by using an array of cylindrical lenses that direct each part of the image to the correct direction. Holography, on the other hand, works by storing the shape of the wavefront of the light emanating from the scene into an interference pattern, and reconstructing it.[5] All these techniques guarantee that only light meant to reach each eye actually does.

Despite being planar surfaces, screen-based displays can reproduce the depth cues of stereoscopy (and, as a consequence, convergence), which allows images to appear not only in the screen, but also "floating" behind, or in front of it.

The images produced by glass-based displays (and early versions of parallax barrier displays and lenticular screens) are considered **stereograms**, since they only consist of two perspectives of the scene (one for each eye).

Parallax barriers and lenticular screens can display various perspectives simultaneously if more of these are stacked in the screen (this of course is limited by the screen resolution, and the minimum size of the lenses or barriers that are allowable without creating aberration or diffraction effects that would distort the final image).

Holography can also reproduce multiple perspectives, but since this is enabled through a different mechanism than direct view packing, the limits to the number of views lie in the recording and reproduction setup rather that on

[5] Holograms store the entirety of the information from a scene – hence their name, which derives from the Greek "holo", the same root that the word "whole" came from.

the screen's information storage ability, and are typically higher than the other autostereographic screen methods.

The ability to produce multiple perspectives leads to the images produced by these three approaches being called **panoramagrams**, which provide motion parallax to a certain degree, and, by extension, stereoscopy.

Motion parallax is not necessarily supported by glasses-based displays, but they can be enhanced to support it (only for a single user) by employing head tracking [1]. Very fast temporal multiplexing using shutter glasses would technically enable multi-user motion parallax, but the hardware and software requirements for this makes the approach impractical.

However with the exception of holography, none of these displays (either glasses-based and autostereoscopic) can provide the accommodation cue, which results in the limitations discussed in Subsection 2.1. Of course, such media can avoid this problem by representing scenes beyond the range of operation of ocular accommodation as a depth cue. A good example of an application that works well this way is flight simulators [1]. But for applications concerning visualization, manipulation and close inspection of complex 3D objects or datasets, a virtual location within arm's length distance is the most (or only) appropriate setup, allowing direct manipulation of the displayed graphics.

4.2 Head-Mounted Displays (HMDs)

HMDs include the once popular stereoscope, current popular devices such as virtual reality (VR) or augmented reality (AR) glasses, and techniques still largely embryonic, such as retinal projection, contact lens displays and brain-computer interfaces.

These solve the "look-around" problem by having the display coupled with the eyes (assuming they can dynamically update the image according to the direction the user is facing), and overcome the projection constraint by placing the display close enough to the eye to allow its relative size to easily cover the field of view of the human visual system.

However, precisely because of these properties, they prevent multi-user applications unless each user wears their own device, and all of them are synchronized. They are also invasive, some more than others, but all considerably more than the remaining systems (except glasses-based stereoscopy).

4.3 Volumetric Displays

Volumetric displays use several techniques to display an image in real 3D space. This means that each point of the image is actually located at the position they seem to be, as if we are seeing a light sculpture. This can be achieved by two main methods: static volume displays, and swept-volume displays.

Static volume displays use a substrate (solid, liquid, or gas) that is transparent or near-transparent in its resting state, but becomes luminous, or opaque, when excited with some form of energy. If specific points can be selectively addressed inside a volume of space filled with such a material, the activation of these points (called volumetric pixels, or voxels) forms a virtual image within the limits of the display.

Naturally, gasous substrates are preferred, and displays have been made using artificial haze to produce unobtrusive, homogeneous clouds suspended in the air that make light beams visible. Purely air-based displays have also been proposed, using infrared laser light to produce excited plasma from the gases in the air, at the focal points of the laser. Advanced forms of such displays are common in science fiction, often mistakenly referred to as "holograms" in popular culture. However, the actual visual quality of such displays is very far from their imagined counterparts, and even quite low compared to other current methods of 3D vision.

Swept-volume displays use a two-dimensional surface that cyclically sweeps through a volume (either moving from one extremity to another, or rotating around an axis) and display, at each point of this path, the corresponding slice of the virtual object. Due to the temporal persistence of vision, this results in what resembles a 3D object.[6]

The main problem with volumetric displays is that, since most of the substrates used become bright when excited, rather than opaque, each point of the virtual object won't block light from the other points,[3] which undermines the very basic depth cue of occlusion; that is, observers would see the back side of objects as well as their front side. Such devices are therefore better-suited to display hollow or naturally semi-transparent objects – for example, wireframe 3D models.

5 Conclusions and Future Work

3D displays are increasingly popular choices to provide new, more immersive, intuitive and interesting tools for education, entertainment (especially in gaming, television and cinema), telepresence, advertising, among others.

Moreover, as the technology advances, more demanding uses of such displays have started becoming feasible or expectable in the near future. Such uses require high-fidelity 3D reproductions of objects, and include areas as diverse as professional design, medical imaging and telemedicine, 3D cartography, scientific visualization, industrial prototyping, remote resource exploration, professional training, and architecture.

Such wide appeal has led to the rapid development of many techniques for 3D visualization, and sometimes this has resulted in poorly-defined boundaries between techniques – especially hybrid ones. This work presented a comprehensive taxonomy of 3D displays, focusing on fundamental characteristics rather than implementation details or practical application. This property should make the taxonomy robust and expansible to include new techniques and innovations.

[6] A one-dimensional point-like light source moving in 3D space can also be used to produce a 3D display, as evidenced in the photographic art of "light sculptures", in which light-emitting objects are photographed with very long exposure times, producing a trail that draws the desired shape in 3D space.

Future work entails a exhaustive listing of implementations and their calatoguing in a table or database that will allow manual or automatic filtering and comparison of different approaches based on the depth cues they implement and other relevant restrictions such as the support for multiple users, the ability to operate without headgear, and others.

References

[1] Dodgson, N.A.: Autostereoscopic 3D displays. Computer 38(8), 31–36 (2005)

[2] Euclid: Optics (300 BC)

[3] Favalora, G.E.: Volumetric 3D displays and application infrastructure. Computer 38(8), 37–44 (2005)

[4] Halle, M.: Autostereoscopic displays and computer graphics. ACM SIGGRAPH Computer Graphics 31(2), 58–62 (1997)

[5] Heilig, M.L.: El cine del futuro: The cinema of the future. Espacios pp. 23–24 (1955), reprinted in Presence: Teleoperators and Virtual Environments, Vol. 1, No. 3, pp. 279-294, 1992

[6] McKenna, M., Zeltzer, D.: Three dimensional visual display systems for virtual environments. Presence: Teleoperators and Virtual Environments 1(4), 421–458 (1992)

[7] Okoshi, T.: Three-Dimensional Imaging Techniques. Academic Press (1976)

[8] Palmisano, S., Gillam, B., Govan, D.G., Allison, R.S., Harris, J.M.: Stereoscopic perception of real depths at large distances. Journal of vision 10(6) (2010), `http://www.journalofvision.org/content/10/6/19.full`

[9] Widjanarko, T.: Brief survey on three-dimensional displays: from our eyes to electronic hologram. Media pp. 1–27 (2001)

[10] Zschau, E., Missbach, R., Schwerdtner, A., Stolle, H.: Generation, encoding and presentation of content on holographic displays in real time. Instrumentation 7690 (2010)

Automatic Visual Speech Animation

José Serra[1], João Freitas[2,3], Miguel Dias[2,3], and Verónica Orvalho[1]

[1]Instituto de Telecomunicações
[2]Microsoft Language Development Center, Tagus Park, Porto Salvo, Portugal
[3]ISCTE-Lisbon University Institute/ADETTI-IUL, Lisboa,Portugal
jserra@dcc.fc.up.pt, i-joaof@microsoft.com,
miguel.dias@microsoft.com, veronica.orvalho@dcc.fc.up.pt

Abstract. Visual speech animation, also known as, lip synchronization is the process of matching a speech audio file with the lips' movements of a synthetic character. Visual speech is a very demanding task, being done either fully manual, which is very time consuming, or with automatic methods based on data analysis. Currently there is still no automatic method that generates any sequence of visual speech without requiring further fine tuning. This research focused on the problem of automatically achieving lip-sync and led to a system that relies on speech recognition to obtain the words from audio and maps them to the visual poses, thus automatically obtaining visual speech animation. The system also supports language translation, which allows animation in a language different from the one spoken. Automatic visual speech animation has great impact in the entertainment industry, where it can reduce the time required to produce the animation of talking characters.

Keywords: Computer animation, Automatic Speech Animation, Speech Translation, Speech Recognition

1 Introduction

Speech is a key element to convey the ideas and thoughts of the person behind any 3D talking character. However, the speech is not composed only by sounds, but also by their poses, which in fact, contribute decisively to the believability of the virtual characters. If the speech animation is not done correctly, i.e. the lips movements do not resemble the ones of a person or if these are not synchronized with the audio, the viewers will find the animation awkward.

A speech sequence can be discretized as a sequence of sounds, also known as phones, and silences. Each phone has associated facial poses, which are positions of the visible articulators of the vocal track, being composed by lips, teeth and tongue. All vocal track articulator can influence the production of a phone, however not all are visible, therefore different phones have the same pose. Visemes are a key concept in visual speech animation. It is important to note that it is not possible to directly generate visual speech by concatenating the visemes, as the produced animation will be over-articulated. This results from the co-articulation, which refers to the effect one phone, and its corresponding viseme, has over the ones around it. Therefore, speech animation has two major components, the synchronization of the audio-lips movements and the co-articulation simulation, which a digital artist has to take into account when animating a speech. Traditionally, an artist has to decompose the audio in its phones, reduce them, as not all influence visual speech, mark them in the sound wave, create the time-line with the visemes and finally control the influence of each in the animation [1]. This is an extremely slow and time consuming process, which takes about 25 to 30 minutes to animate a sentence.

Several techniques have emerged that try to automate different stages of the speech animation pipeline, such as lips-audio synchronization, also known as data-driven techniques. These have the goal of reducing the speech animation costs. It is within data driven techniques this research made its contribution, with the goal of creating a fully automatic visual speech animation system, capable of generating the 3D animation data from a speech audio signal. Language translation was also added to the equation, as a consequence of the need in the videogame and film industry to animate their products in different languages. This is a problem because each time, e.g. a film, is translated all the animation has to be done from scratch for each language, as it usually is not possible to reuse it in different languages. Thus , automatic speech animation and translation largely helps to reduce the time required to animate a film or video-game.

The proposed system is based on automatic speech recognition, which is used to obtain the words both in Portuguese and in English. The audio is analysed to get the orthographic and phonetic transcriptions, the text is then translated if necessary also generating the new audio and the visemes that map to the found phones are chosen. With the system it is possible to automatically generate visual speech. The following sections are organized as it follows: state of the art with the description of what has been done in speech animation, system description, which contains a more detailed description of how the system was implemented, results & discussion, with the main results obtained, how they were obtained and their analysis and finally the conclusion and future work.

2 State of the art

Automatic speech animation, in contrast to traditional animation, does not rely on direct work by the animator. It is based on different sources of information, which are then mapped to a model and processed to generate the animation. Based on the sources it is possible to divide data-driven speech animation in two types: performance-based or MoCap and speech based. Performance based animation tracks the movements of a person and maps them to face model, using video or infra-red signal analysis. The tracking can be based on marker placed on the actors [4], or markerless, such as [6]. However, to be able to correctly track the actor, special equipment and very controlled environments are required, thus having high costs. Using Mocap all the facial movements of the actor are captured, therefore the lips movements and audio synchronization are automatically achieved. Fine tuning is still required as the tongue and inner lips movements are not captured. On the other hand, speech based animation relies on the analysis of text and/or audio to generate the animation data. One of the main advantages, compared to MoCap, is the reduction of special equipment needed, as only a microphone is required. The present work falls within the second category.

Visual speech animation research is divided in two major areas, depending on the mapping between the input and the visual poses, being them: phone-to-viseme map and sub-phonetic map. In the first, the phones are obtained from text or audio analysis, mapped to the visemes and finally assembled in a timeline. However, the speech animation process is composed by two parts: lip-sync and co-articulation and with this example the co-articulation is not simulated, therefore the animation will look over-articulated. One common approach is to use triphones [2, 3] instead of only phones. As the triphones contain all the motion information, the coarticulation is automatically modelled. The major problem with triphones is the creation of the database that is very costly, e.g. with a 40 phone language there will be roughly 40^4 triphones. Another approach is to use a model to simulate this effect and one of the most commonn is the Cohen and Massaro [5] model, where each phone has a curve of influence that can be larger than the duration of the phone, thus simulating the co-articulation effect. The major problem with this model is that it is language dependent. It is important to note that to date there is still not any model capable of simulating all the possible effects. Regarding the lip-sync, most approaches today use a text-to-speech (TTS) systems that take a text as input and generates the audio, with the goal of getting the phones and their durations, e.g. in [7]. However, with a TTS, it is not possible to create a fully automatic speech animation system as the text has to be manually inserted and the audio will have a slightly robotic tone. Another approach is to use automatic speech recognition (ASR) to get the phones and their timings, which solves this problem. Unfortunately ASR is still far from perfect, at least for general dictation, however if the context is well defined the results of the recognition tend to be very good.

Sub-phonetic mapping tries to simulate the behaviour of continuous speech. By directly mapping the sub-phonetic features to the visemes, more detail can be obtained from a sound wave, as the in-betweens of the phones can also be used to generate data for the animation, thus automatically simulating the co-articulation. Another advantage is the increase of performance, as no phonetic discretization is required. The major problem with sub-phonetic methods is the high sensitivity to noise. Here the map between the sound features and the poses is learned via machine learning, such as in [8] where the sound features were associated to the facial features using PCA and nearest-neighbour mapping. [9] used a set of classifiers to create a real-time speech animation system.

The proposed system uses ASR to get the orthographic transcription of what was spoken. Using the audio as input it is possible to create a fully visual speech animation system. It is based on a previously created phone-to-viseme map that will allow the generation of the animation data from the words recognized.

3 System Description

The created system is divided in two major components: 1) a speech processing tool, responsible for all the input analysis and choice of data that will drive the animation, and 2) a plugin inserted in an animation engine, which has the goal of connecting both. These components lead to a logical division between processing and animation as all the animation data is generated in an entirely independent tool, thus making the system independent of the animation engine. Translation was also added to the created system mainly due to its application in the entertainment industry, allowing instant translation of both the audio and facial animation. It is necessary to note that, when translating a speech, all the input audio is made useless as it can be used in the animation. Therefore, it is necessary to use a text-to-speech to produce the new audio that will drive the animation data. The two components of the system are now described with the whole pipeline in the Fig. 1.

3.1 Speech Processing Tool

The initial behaviour of the tool varies according to the fact that it is translating the input or not. All the system is controlled by a core, responsible for the flow of information between the different modules, which made the system modular enough to allow a person to change the approach of any specific module and keep the system work as intended.

Speech Recognition

The input module was implemented using Microsoft Speech API (SAPI 5.4) [13], more precisely the SR engine available within SAPI. It can be used either in dictation or command and control mode. The first allows the recognition of any utterance, however it has higher error rates, compared to the second, which allows the definition of commands, sentences, being the only ones recognized by the system. Both modes can be used by the system in Portuguese (pt-pt) or English (en-us). However, the design of the pt-pt language model for the dictation scenario, included essentially unigrams bigrams and trigrams of common words of the Portuguese vocabulary, telephone numbers, proper names of people and some business names and addresses of the Portuguese culture and market, and was not yet trained with a general purpose large text corpus. Therefore, the word and sentence error rates of the pt-pt SR system, when working in dictation mode, are poor, where the same rates, for the dynamic context-free-grammar based command and control scenario, are more approximated to the en-us case.

As soon as the system starts running and all the information is loaded, the SR engine starts to analyse the audio from the microphone, trying to recognize any utterance. This analysis is done until the system is closed. Each time a recognition event is detected, a list of words is created and their audio stored in files, one word per file. If the pt-pt engine is used, it also provides a list of phones in the IPA format [10], corresponding to the phonetic transcription of the resulting recognized word or phrase. This is the information passed to the core, using a thread, so the engine keeps recognizing the words. It is not possible to

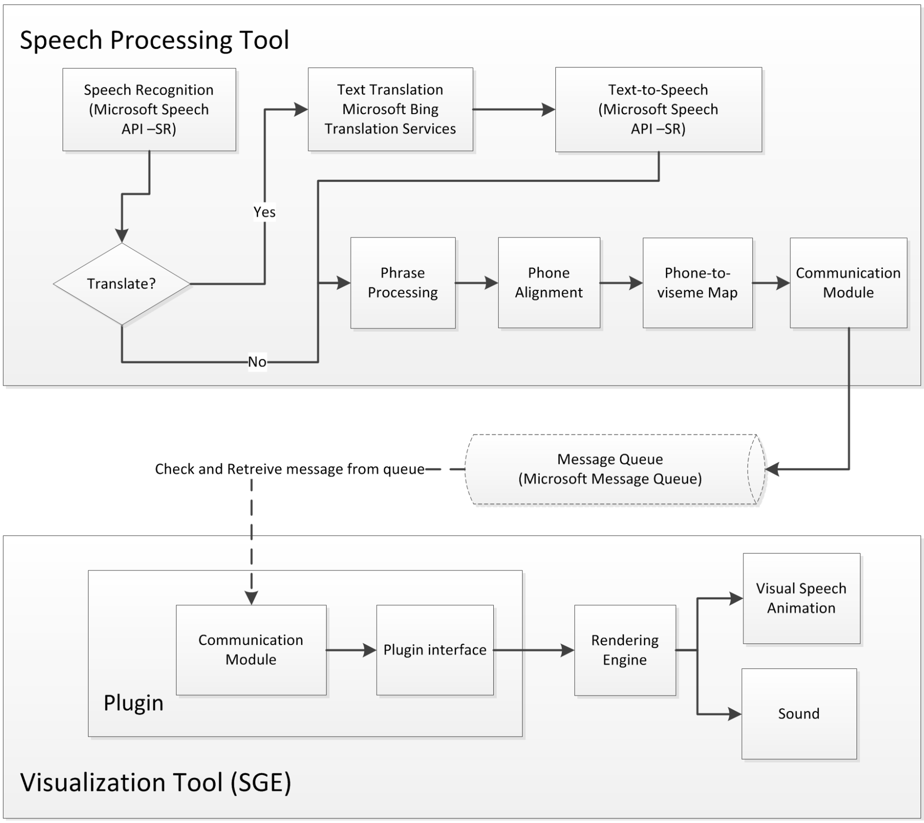


Fig. 1. Pipeline of the proposed system composed by the main modules: the speech processing tool and the plugin inserted in the visualization tool. Here, it is possible to see the different components, the main technology used in each, if any used, and the flow of information until the animation and sound are played. First the words are recognized in the speech processing module, then if in translation mode, the new words and audio are obtained, respectively by the translation services and the TTS, then the words are analysed to obtain their phones in the phrase processing module. After that the phones are aligned with the audio to get their timings and durations. Finally, the visemes, obtained from the phone-to-viseme map are passed to the visualization tool together with the timings, which forms the data required to play a speech animation.

obtain the phones duration and timing from the SAPI, therefore this information has to be obtained in the other modules.

Text Translation

To translate a sentence, the Microsoft Bing Translator [14] web service was chosen. This service was implemented with a SOAP interface, used by the module to make a translation request. The whole process works in an asynchronous way. When the message is translated, a method within the module is called, used to pass the information to the core.

Phrase processing
Splitting the words into their basic sound units, i.e. the phones, is the task of the phrase processing module. The splitting is done using a phonetic transcription dictionary (a phonetic lexicon) provided by Microsoft, both for English and Portuguese. The first is complete, therefore it is only necessary to split the sentence into its words and then get the phones from a dictionary structure that has the words as keys. The phones are represented in the Universal Phone Set (UPS) [12]. Regarding the Portuguese language, the provided lexicon is not complete. When a word or sentence is not available, the phones, obtained from the SR engine, are analysed and converted from the IPA format to UPS. The output is the list of phones in the UPS format.

Phone alignment
The phone alignment module is based on the statistical duration of each phone. The average duration is normalized so the sum of all phones' duration, in a word, has the same length as its corresponding audio file. The pt-pt audio durations were obtained from the Microsoft, collected from the analysis of a database with 100 hours of speech. The en-us durations were obtained using Microsoft TTS, where a set of 500 words were spoken and the phones' duration obtained. The time value is the average per phone. This module receives the audio and the phones and returns the durations per phone.

Speech Synthesis
Like the input module, the speech synthesis is also based on the SAPI [13], in this case the TTS engine is used, instead of the SR engine. It takes a text as an argument and generates the utterance from it. As it is only required to generate the audio, instead of playing the utterance, the TTS is used to directly create a file per words' utterance. The output is the list of audio files.

Viseme Selection
The visual selection module contains the map between the phones and visemes, thus responsible for the viseme selection. The first step is to load a file that contains the mapping between the phones and the visemes. Currently this is a XML file that maps each phone to a viseme, one mapping per language. It is important to notice the phones are in the UPS format and the visemes' names are the same as the ones used by the visualization tool. The selection of the visemes currently results of directly using the phones, obtained from the previous modules, and just get their corresponding viseme name. Their animation weights are always 1. The output is a list of the visemes and their weights grouped by word.

Communication
The Microsoft Message Queue (MSMQ) [15] was used to allow the communication between different programs. A system queue is created that all applications can use to send and read messages, thus being an easy way exchange information. A message containing the audio files, duration and weights is passed to this module that will place it in the MSMQ. One message per word is created.

3.2 Visualization Tool

The full automatic pipeline was implemented in the Serious Game Engine (SGE), which is the current engine used in Porto Interactive Center. Its first version was created under the LIFEisGAME project [11], whose prototype is completely implemented using the engine. It is undergoing active development and currently is in a beta phase. The SGE is a highly modular engine, consisting of a small core and a set of plugins/modules that can be added to perform specific tasks. The modules' behaviour is defined by interface classes that establish the methods each module has to implement. A developer only has to define these methods and they will seamlessly integrate in the engine. To create the speech animation system, it was only necessary to add the plugin to the SGE and make it query the plugin interface for new speech animation data. Unfortunately after implementing the whole system, some problems arose from this choice, mainly related with the frame-rate. Using the SGE with allowed only visualization of the speech animation at 25 fps, which is not enough for speech, where 30/35 fps are the absolute minimum. Therefore, the plugin was also implemented within Maya, which even though it does not allow a full automatic pipeline, permits the creation of the animation with the desired frame-rate. Both cases, plugin for SGE and for Maya, are now briefly described

Plugin

The plugin only has the function of retrieving the data from the message, i.e. it parses the data string and obtains the audio file, visemes, their durations and weights. The message itself is obtained from the communication module, which accesses the MSMQ and retrieves the data. It is the engine that controls when the plugin verifies the queue.

Maya Extension

A script for Maya was created that read the animation data from a *csv*, and from this created the time-line. To generate the file, the communication module, within the speech processing tool, was changed to store the data in a file, instead of the Microsoft Message Queue

4 Results & Discussion

This system consists of automatically generating the speech animation from an utterance, therefore it is extremely important to validate the visual results. The animation was created using the Nene model, created by Xenxo Alvarez (technical director), that has 99 joints. A total of 80 visemes were created, 11 plus neutral for the minimum English viseme set and two sets for the Portuguese, 15 plus neutral for the minimum set and 51 plus neutral for the extended version, all done by Pedro Bastos (digital artist). The data necessary to create both minimum and complete pt-pt sets were obtained from Microsoft, while the en-us was based on the Preston Blair extended phoneme series [16], which defines the mouth, teeth and tongue shapes for most phones. The Preston Blair visemes are used in manual mouth animation, which in fact does not require all the visemes to generate the animation, however this is a problem when using a phone-to-viseme approach, therefore the missing visemes were mapped to the Portuguese minimum set. It took about 8 hours of work to complete the creation of all visemes.

The whole system was implemented in a laptop with an Intel Core 2 Duo T9600, 4Gb of memory and an ATI 4650 1Gb. The system was integrated in the SGE, thus obtaining fully automated speech animation system that works near real-time. If the recognition delay is ignored, it takes about one second from the recognition event until the animation data is on the SGE. All the audio was recorded using the Logitech ClearChat Pro USB Headset at 8KHz and mono stereo. The results can be seen in a video uploaded to youtube in http://www.youtube.com/watch?v=82jid8HPZFo. The validation was done for two case studies, which are now described.

The system was tested only for two utterances, thus serving only as a proof of concept. The two utterances were "olá a todos" and "animar a fala" (Figure 2). Both cases were animated using the Portuguese minimum phone set. The figures show the visemes for the same utterance's word. The audio for both utterances is also shown.

4.1 Qualitative Validation

The qualitative validation was done by showing the video of the character speaking to members of the research group, and, as expected, animating a speech by directly concatenating the visemes does not result in realistic speech animation. The animation is clearly over-articulated. It appears however, that the vowels visemes's weight are somewhat near the correct value, at least for the ones that are more pronounced. This results from the fact that the louder the phones are spoken, the wider the visemes are. The speech animation was also integrated with basic head movements such as nod. And in this case the resulting animation looked as it was expected, where there was no interference between the speech animation and the head movements.

4.2 Quantitative Validation

The quantitative evaluation was done in terms of time required to animate a sentence. The system generated all the audio synchronization information, used to place the audio and the visemes in the correct position of Maya's timeline, thus reducing the time to manually synchronize all the visemes. The weights had to be fine-tuned in both utterances by changing their weights and animation curves, used to control the variation of the weights. About 5 minutes per sentence were required to fine-tune the animation. Based on Xenxo's previous experiences, he estimated it would take about 25/30 minutes per utterance. Thus, a reduction of 500/600 % of the time required to animate each utterance was achieved.

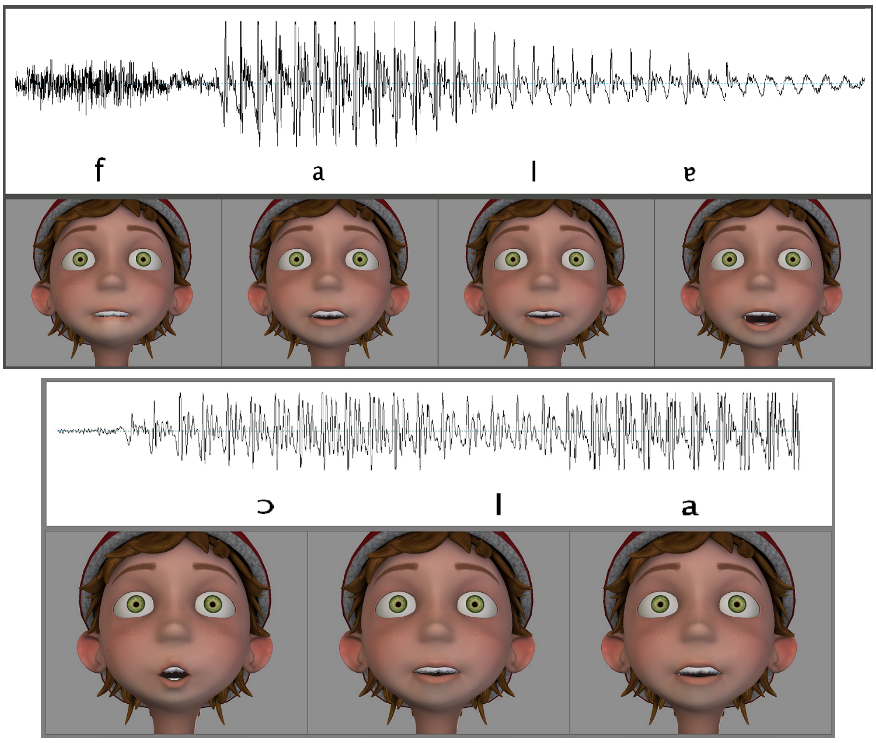


Fig. 2. Visemes and respective sound waves first, for the utterance "olá" and second for "fala". All automatically generated.

4.3 External Dependencies

The quality of the output animation is greatly influenced by each module and method used in it. Even though most modules contribute decisively to the animation generation, there are some factors that are not controlled by the implementation. It is also important to note that the animation itself also contributes to the quality of the results. These are referred as system's external dependencies and are:

- **Speech recognition**. If the words are not recognized correctly, the resulting animation will be completely wrong, as the audio and animation will not be same. Furthermore, any audio analysis will produce non usable results.
- **Phone alignment**. The phone alignment module is based on the statistical duration of phones, which should be obtained from the analysis of considerable size databases. If not done correctly it may influence the audio-visual synchronization. Another possibility is the use of a force alignment technique, which requires the availability of a speech recognition engine for the language

- **Translation**. The translation does not directly affect the animation quality. However, it can, and will influence the users' expectations as non-coherent utterances may be animated.
- **Model and Animation**. The quality of the rig influences how the in-betweens from two poses are calculated. Furthermore, the blending of different poses contributes to the smoothness of the resulting animation. If not done properly, the viewer will find the animation awkward.
- **Visemes**. The system is model independent, which means the visemes can be created in any character, but for the animation to be correct, these visemes should resemble to the poses of a real person. When creating these, it is also necessary to be careful, as a viseme may be mapped to more than one phone, therefore they have to be created accordingly the mapping that was chosen. If the poses are too exaggerated the animation will look over-articulated even if the correct poses and weights are provided.

The main limitation of the system is, without a question, the absence of co-articulation simulation, which prevents the resulting animation from being fully realistic. It is important to note that co-articulation was never meant to be addressed, at least until now. There is, however an intrinsic limitation with the way phones are aligned, especially with English that is related to the statistical duration of each phone. The Portuguese set is capable of fully represent the statistical duration of the phones, however in English this is far from true, as the values were only obtained from a 500 words corpus of synthesized speech. Therefore, before trying to simulate the co-articulation it would be necessary to solve this issue.

It is also important note the choice of the model. As the model is not realistic, a person has lower sensitivity to errors that may occur, therefore it would be important, in future experiments, to use a human-like model to better understand the synchronization.

5 Conclusion

The purpose of this work was to create a fully automatic system capable of automatically generating the synchronization of the audio and the lips from a speech signal and at the same time support speech animation generation in different languages. That was succefully accomplished, however this is not enough to generate realistic speech animation, being necessary to simulate the co-articulation effect. This is the natural evolution of the system, where the goal is to change the visemes according to the audio.

Some work has already been made in this direction, where from the analysis of th audio it is already possible to change the weight of the viseme, e.g. when a person speaks louder the viseme is wider, i.e. its weight is higher. Such already reduces the problem of over-articulation, even though it is not enough. As a result, the next steps are to completely implement the weight variation system and to add a co-articulation model such as Cohen-Massaro's with the purpose of creating a fully automatic visual speech animation system capable of generating life-like speech animation with little to no fine tuning.

References

1. Osipa, J.: Stop Staring: Facial Modeling and Animation Done Right. John Wiley & Sons, 2010.
2. Bregler, C., Covell, M., Slaney, M.: Video Rewrite. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97, pages 353-360, New York, New York, USA, August 1997. ACM Press.
3. Zhou, Z., Zhao, G., Pietikäinen, M.: Synthesizing a talking mouth. In Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing - ICVGIP '10, pages 211-218, New York, New York, USA, December 2010. ACM Press.
4. Guenter, B., Grimm, C., Wood, D., Malvar, H., Pighin, F.: Makingfaces. In Proceedings of the 25th annual conference on Computer graphics and interactive techniques, volume 205, pages 55-66. ACM, July 1998.
5. Cohen, M., Massaro,D.: Modeling coarticulation in synthetic visual speech. Models and techniques in computer, page 139?156, 1993.
6. Zhang, Z., Snavely, N., Curless, H., Seitz, S.: Spacetime Faces: High-Resolution Capture for Modeling and Animation. Data-Driven 3D Facial Animation, pages 248-276, 2007.
7. Yotsukura, T., Morishima, S., Nakamura, S.: Modelbased talking face synthesis for anthropomorphic spoken dialog agent system. In Proceedings of the eleventh ACM international conference on Multimedia - MULTIMEDIA '03, page 351, New York, New York, USA, November 2003. ACM Press.
8. Gutierrez-Osuna, R., Kakumanu, P., Esposito, A., Garcia, O., Bojorquez, A., Castillo, J., Rudomin, I.: Speech-driven facial animation with realistic dynamics. IEEE Transactions on Multimedia, 7(1):33-42, 2005.
9. Liu, J., You, M., Chen, C., Song, M.: Real-time speechdriven animation of expressive talking faces. International Journal of General Systems, 40(4):439-455, 2009.
10. International phonetic alphabet. http://www.langsci.ucl.ac.uk/ipa/.
11. Porto Interactive Center - Learning of facial emotions using serious games. https://www.portointeractivecenter.org/lifeisgame/
12. Microsoft universal phone set white paper. http://download.microsoft.com/download/speechSDK/Patch/1/NT5XP/ENUS/ UPSWhitePaper(3.2).doc.
13. Microsoft speech api. http://msdn.microsoft.com/enus/ library/ee125663
14. Microsoft translator. http://msdn.microsoft.com/enus/ library/dd576287.aspx.
15. Microsoft message queue. http://msdn.microsoft.com/enus/ library/ms71147220 11.
16. Preston blair phoneme series. http://www.garycmartin.com/mouth shapes.html.

Classification, Implicit Segmentation and a Chronological Prediction Model for Cinematic Sound

Pedro Silva

pedro.alexandre.silva@fe.up.pt
Faculdade de Engenharia da Universidade do Porto
Departamento de Engenharia Informática
Rua Dr. Roberto Frias, s/n
4200–465 Porto
Portugal

Abstract. This report presents work done on classification and segmentation of cinematic sound employing support vector machines (SVM) with sequential minimal optimization (SMO). Speech, music, environmental sound and silence, plus all pairwise combinations, are considered as classes. A model considering simple adjacency rules and probabilistic output from logistic regression is used for segmenting fixed-length parts into auditory scenes. Evaluation of the proposed methods on a 44-film dataset against k-nearest neighbor (KNN), Naive Bayes and standard SVM classifiers shows superior results on all performance metrics. A subsequent meta-study is presented, which proposes optimizations to the building of similar datasets with regard to sample sizes. Finally, it employs sound classes resulting from classification as descriptors in a chronological model for predicting period of production of a given soundtrack. Decision table classification is able to estimate the year of production of an unknown soundtrack with a mean absolute error of approximately 7 years (actual → predicted correlation coefficient = 0.6.)

1 Introduction

This report addresses the broad subject of content analysis of cinematic sound. That is, it details a set of techniques for inferring meaning out of the use of sound in a cinematic setting. *Cinematic sound* is defined as sound used either diegetically (produced by agents in the narrative) or non-diegetically (external to the narrative) in a visual presentation. In the field of film theory, there has been little effort to systematize this topic through the use of computational methods so far. As a result, the so-called semantic gap, that is, the gap between low-level machine-computable features and high-level human-perceptible semantics [4], is larger in this than in other fields. For example, the questions of the importance of the introduction of sound in film in the late 1920s, and its subsequent evolution in terms of scene length, music predominance over speech and other factors across different periods, regions and-or producers, are long standing. Arguably, automating content analysis in cinematic sound would help address them.

Amongst other approaches [3, 2, 24], [27] has proposed to address this gap through a model of media encoding (i.e. synthesis, production) and decoding (i.e. analysis, criticism) based on a hierarchy of low to high level “aesthetic elements”. The method proposed here is also hierachical, with feature extraction, classification, audio segmentation, meta-feature extraction and meta-classification all building upon one another toward the high level.

Audio is a low-bandwidth medium in the context of digital audio-visual media, in which video comparatively uses more storage and communication bandwidth. Yet, *semantically*, audio is arguably as complex, and its information potential as high, as video. The analysis of cinematic sound through computation offers efficiency to information retrieval and recommendation systems that purport to consider the semantics of audio, to name two examples. To that end, it would be useful to optimize the sample sizes taken when considering content analysis, since the manual creation of labeled datasets is expensive, and computer efficiency is essential. Although such research has been done for other kinds of media (namely print media) [22, 17, 9, 18, 20, 19], audio has been neglected. The study reported here proposes to address that need.

Although typically only simple, pure sound classes (e.g. speech and music) are considered in the context of high-accuracy classification (i.e. true positive rate $> 90\%$) [11], this study attempts to expand those into a larger set, including hybrid classes combining the simple ones. [12] have shown that a SVM classifier, together with simple adjacency rules for building auditory scenes (see [23] also), is very effective in achieving high-accuracy of classification and segmentation. The use of timbral features is well established in the literature as providing the necessary discriminatory power to differentiate not only simple music and speech classes, but also for high-level classification tasks employing complex classes [25].

The tools resulting from this study are: (a) a method for segmentation and classification of cinematic sound segments into auditory scenes (Sects. 4 and 5), (b) a new visualization tool for representing cinematic sound segmentation: the sound class map, which shows the distribution of sound classes in a single cinematic presentation (Sect. 6), (c) a sampling strategy for minimizing dataset size and complexity and maximizing generalizability (Sect. 7), and (d) a chronological model for prediction of period of production (Sect. 8) .

This report is organized as follows: Sect. 2 characterizes the dataset. Section 3 details pre-processing, features, and rationale. Section 4 defines the labeled dataset, sound classes and classifier. Section 6 proposes a model of auditory scenes, and a novel way of visualizing audio segmentation. Section 5 presents the results of cross-validation on the dataset, features and classifier. Section 7 proposes sample size optimization for similar datasets, and a chronological model for predicting the period of production of a soundtrack. Section 9 discusses the effectiveness of the chosen feature space, classifier, auditory scene model, sampling strategy and chronological prediction model. It concludes by offering potential improvements to be applied to future research on this topic.

2 Dataset Characterization

2.1 Population

The population is comprised of the top-grossing American feature films from 1970 to 2006 (population size = 104) as reported by the Internet Movie Database as of 19 November 2006. *Top-grossing* is defined as being in the top-250 list of all-time U.S. box office revenue, without adjustment for inflation. *American* means that the production was financed by at least one major American production company, and thus excludes independent productions. *Feature film* is defined as a production at least 40 minutes long released for the theatrical market.

2.2 Sample

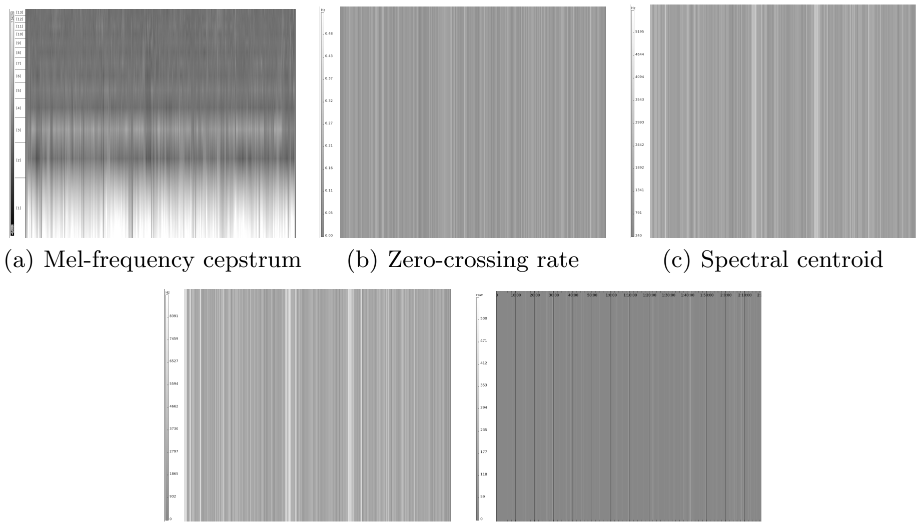
Table 1 lists the result of stratified weighted random sampling on the population (sample size = 44), where each stratum represents one decade. There are 9 films for 1970–79, 10 for 1980–89, 12 for 1990–99, and 13 for 2000–06. The imbalance of this distribution reflects the contribution of later decades to the population.

Table 1. Stratified weighted random sample filmography, ordered chronologically.

1. Patton (1970)
2. Harold and Maude (1971)
3. The Godfather (1972)
4. The Sting (1973)
5. The Conversation (1974)
6. One Flew Over the Cuckoo's Nest (1975)
7. Taxi Driver (1976)
8. Apocalypse Now (1979)
9. Manhattan (1979)
10. The Elephant Man (1980)
11. Raiders of the Lost Ark (1981)
12. Blade Runner (1982)
13. Scarface (1983)
14. Amadeus (1984)
15. Once Upon a Time in America (1984)
16. Platoon (1986)
17. The Princess Bride (1987)
18. Glory (1989)
19. Indiana Jones and the Last Crusade (1989)
20. Goodfellas (1990)
21. Reservoir Dogs (1992)
22. Schindler's List (1993)
23. Pulp Fiction (1994)
24. The Shawshank Redemption (1994)
25. Se7en (1995)
26. Fargo (1996)
27. Sling Blade (1996)
28. American Beauty (1999)
29. Magnolia (1999)
30. The Straight Story (1999)
31. Toy Story 2 (1999)
32. Memento (2000)
33. Snatch. (2000)
34. The Lord of the Rings: The Fellowship of the Ring (2001)
35. The Lord of the Rings: The Two Towers (2002)
36. Kill Bill: Vol. 1 (2003)
37. The Lord of the Rings: The Return of the King (2003)
38. Mystic River (2003)
39. Pirates of the Caribbean: The Curse of the Black Pearl (2003)
40. Eternal Sunshine of the Spotless Mind (2004)
41. The Incredibles (2004)
42. Sin City (2005)
43. Borat: Cultural Learnings of America for Make Benefit Glorious Nation of Kazakhstan (2006)
44. Little Miss Sunshine (2006)

3 Feature Selection

The features used included 13 Mel-frequency cepstral coefficients (MFCC) [13], the zero crossing rate (ZCR) [21], and spectral centroid [6], flux [5] and rolloff (see Fig. 1). The use of these timbral features is well established in the literature as providing the necessary discriminatory power to differentiate not only simple music and speech classes, but also broad and general classification tasks involving music genre, user preferences and broadcast categories, for example [25]. As such, these features were considered adequate to discriminate hybrid sound classes.



(a) Mel-frequency cepstrum (b) Zero-crossing rate (c) Spectral centroid

(d) Spectral rolloff (e) Spectral flux

Fig. 1. Spectral features for Federico Fellini's *8 and* $\frac{1}{2}$. x-axes: soundtrack time-line. y-axis: 1(c) center of gravity of the magnitude spectrum of the short-time Fourier transform (STFT); 1(d) frequency below which 85% of the magnitude distribution is concentrated; 1(e) rate of change of the power spectrum through a distance comparison of adjacent normalized frames; 1(b) rate of change of the signal's sign; 1(a) discrete cosine transform of the logarithm of the magnitude spectrum of the Mel STFT.

The means and variances of each of these features were calculated for each 3 s segment during training. The basic analysis window was 512 samples long with no overlap, keeping a buffer of 40 windows for computing a moving average for the features of 129 frames.[1] The resulting feature vector was composed of the buffer and frame means and variances of each of the actual 17 features. resulting in a final 68-feature vector for each sound segment.

[1] Which, at a sampling rate of 22050 Hz, represent $512 \times 129 \div 22050 \approx 3\,\mathrm{s}$

4 Classification

4.1 Labeled Set

The labeled set was composed of 3560 distinct audio segments randomly sampled from the list in Tab. 1, each 3 s long. These segments were manually classified by three human coders into any of the classes in Sect. 4.2, with an inter-coder agreement coefficient $\kappa = 0.83$[2] [1]. Conflicts were resolved by majority voting.

4.2 Classes

The simplest (first-order) classes considered were speech, music, environmental sound, and silence. Although used throughout the literature, first-order classes only were not enough to build the meta-feature set used in the second part of this study. Therefore, these were subdivided into combinations of two-element hybrid (second-order) classes, corresponding to all possible combinations of two first-order classes. Table 2 describes the working operational definitions for each class used in this study.

Table 2. Sound classes and corresponding working operational definitions.

Speech Dialog, perceptually unaccompanied by music or environmental sounds, noise, or other transient sounds.
Music Harmonic composition with a varying degree of subjective dissonance, perceptually unaccompanied by speech, noise, or other environmental sounds.
Environmental sound Foley (i.e., clothes rustling, footsteps), true environmental sounds (i.e., wind blowing, traffic), and noise in the strictest sense (i.e., white noise and other types of random or pseudo-random inharmonious sounds).
Silence Perceptual absence of any sound.
Speech with music background Speech that is perceptually dominant over music.
Speech with environmental sound background Speech that is perceptually dominant over environmental sound.
Music with environmental sound background Music that is perceptually dominant over environmental sound.

4.3 Classifier

The classifier used in this study was a fast implementation of a support vector machine (SVM) [26, reprint from 1966 original] using sequential minimal optimization (SMO) [15]. A linear kernel was used, since the data was linearly separable in the binary classification problem. Logistic regression models were built during training and prediction to obtain a posterior probability for each instance classification [16]. All classification tasks were done using the WEKA machine learning framework [7], including its specific SMO implementation.

[2] "Almost perfect" agreement, according to [10]

5 Validation

Ten-fold cross-validation was used to obtain performance metrics of the SMO classifier against ZeroR, Naive Bayes, KNN, and SVM classifiers. Table 3 shows that SMO performed better than its direct alternatives ($\mathit{\Delta}_{\mathtt{TP}} > 10\%$), denoting "substantial" agreement [10] with the baseline labels ($\kappa = 0.7$). Table 4(a) shows that most misclassifications shared a first-order constituent; the four circled numbers represent approximately 43 % of total misclassifications. Table 4(b) shows that performance was consistently high across classes, with the exception of music, due to a high actual music with environment sound class versus predicted music class error rate.

Table 3. Cross-validation summary for SMO classifier performance versus a baseline classifier that predicts the mode (ZeroR), a classifier that simply employs Bayes' theorem while assuming independence between samples (Naive Bayes), a common clustering classifier based on close samples in the feature space (k-nearest neighbor), and a support vector machine classifier without the SMO optimizations (standard SVM).

			Classifier		
Performance	ZeroR	Bayes	KNN	SVM	SMO
Correctly classified	32.32%	64.19%	64.34%	65.73%	**76.01%**
Kappa statistic	0	0.56	0.55	0.57	**0.70**
Mean absolute error	0.23	0.10	0.10	0.13	**0.10**
Root mean squared error	0.34	0.30	0.32	0.26	**0.22**
Relative absolute error	100%	45.23%	44.87%	58.99%	**41.97%**
Root relative squared error	100%	89.52%	94.08%	76.19%	**66.11%**

Table 4. Performance of SMO classifier. env: environmental sound, mus: music, sil: silence, sp: speech, m-e: music over environmental sound, s-e: speech over environmental sound, and s-m: speech over music. TP: true positive rate, FP: false positive rate, Pr: precision, Re: recall, Fm: F-measure, ROC: receiver operating characteristic.

(a) Confusion matrix

Class	env	mus	m-e	sil	sp	s-e	s-m
env	**242**	2	22	2	8	32	2
mus	2	**254**	44	0	4	4	10
m-e	28	58	**180**	0	2	16	26
sil	10	0	0	**36**	0	0	0
sp	6	0	0	6	**470**	78	26
s-e	28	2	18	6	92	**788**	66
s-m	0	8	26	18	18	90	**382**
All	316	324	290	68	594	1008	512

(b) Cross-validation accuracy

Class	TP	FP	Pr	Re	Fm	ROC
env	0.78	0.03	0.77	0.78	0.77	0.98
mus	0.78	0.03	0.78	0.80	0.79	0.97
m-e	0.58	0.04	0.62	0.58	0.60	0.94
sil	0.78	0.01	0.72	0.78	0.80	0.99
sp	0.80	0.05	0.80	0.80	0.80	0.96
s-e	0.79	0.11	0.78	0.79	0.79	0.93
s-m	0.73	0.05	0.75	0.73	0.74	0.95
All	0.76	0.06	0.76	0.76	0.76	0.95

6 Audio Segmentation

An auditory scene is a coherent collection of sound sources, only a few of which are perceptually dominant. In this model, a scene change occurs when the majority of these dominant sources change. [23, p. 2441]. There is a natural equivalency between objects (sound sources) and characteristics (sound classes) [14, pp 26–27]. Each scene was therefore modeled as a collection of adjacent segments that share a common class, under the rules in [12, p. 6]. Further, each scene was annotated with its cumulative posterior probability, based on the estimates of the classifier. Finally, basic descriptive statistics were computed for each soundtrack, namely: average and standard deviation of scene length (in seconds), average and standard deviation of scene frequency (scenes per minute) and, for each class, its coverage coefficient (the proportion of the duration of a soundtrack that that class represents).

Figure 2 exemplifies the result of this process. There are (a) seven time series, one for each sound class; (b) N discrete auditory scene points (the centered differently shaped marks); (c) individual scene lengths (the horizontal ticks—a single center mark is the basic unit, a three-second long scene); and (d) the cumulative posterior probability, denoted by the vertical ticks, varying from a simple point (low probability of prediction being correct) to a line spanning the distance to the adjacent time series (high probability of the prediction being correct).

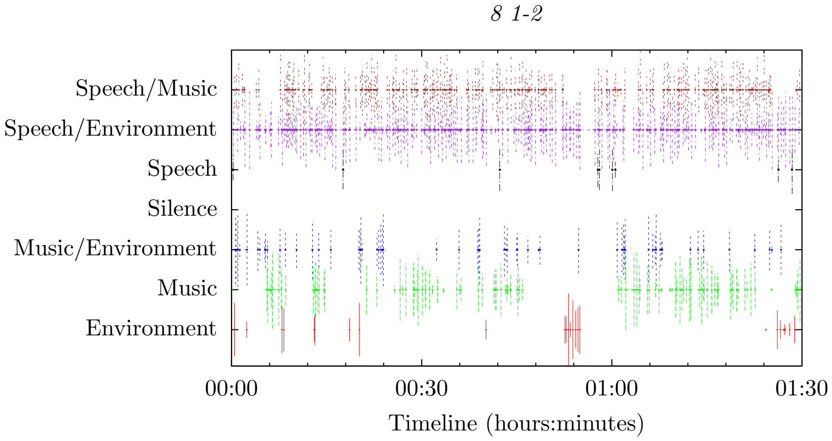


Fig. 2. Class segmentation for Federico Fellini's *8 ½*. Each mark represents an auditory scene. Horizontal ticks show the scene's length. Vertical ticks show the cumulative posterior probability that the scene has been correctly classified.

7 Meta-analysis

Figure 3 shows cross-corrrelation in variation of changes in class coverage, scene frequency and average scene length from 1970–2006, with progressively higher sample sizes per decade. For each decade and for each class, random sampling with substitution was done with sample sizes 1 through 9. Cross-correlation was computed for each sub-sample with the entire sample, and its results averaged into three final series, one per meta-feature. All three series are highly cross-correlated ($cc > 0.94$). Assuming that the sample size is representative of the population, this is evidence that the anchor weigths used for the stratified sampling in Sect. 2.2 can be reduced from 10 to 8 films per decade with no loss of power.

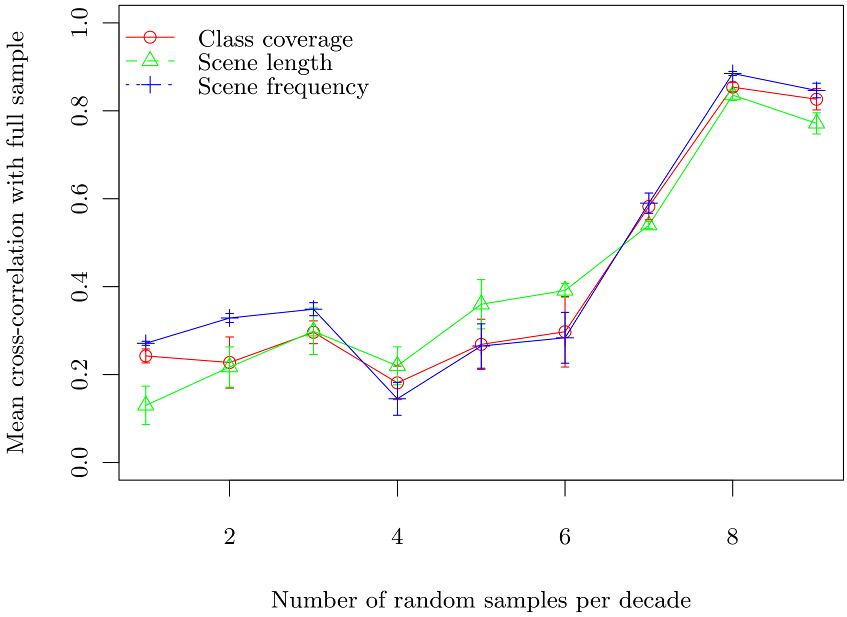


Fig. 3. Meta-sampling summary of cross-correlation in class coverage, scene frequency and scene length, 1970–2006, between different sample sizes and full sample. Error bars show standard error at 0.95 confidence level.

8 Chronological prediction model

The last step was to train a meta classifier with the task of predicting the year of production of unlabeled instance soundtracks. The classifier used was a rule-based decision table [8] with a greedy hillclimbing search space strategy. The features used in this step were the output of the previous stages of classification and segmentation. That is, the feature space extracted from each instance soundtrack was the class-specific and global means and standard deviations of length, scene frequency and coverage coefficient, for a total of 58 features. The correlation coefficient between predicted and actual years was 0.6, with 7.037 years mean absolute error and 9.114 years root mean squared error. Figure 8 shows the output of this classifier.

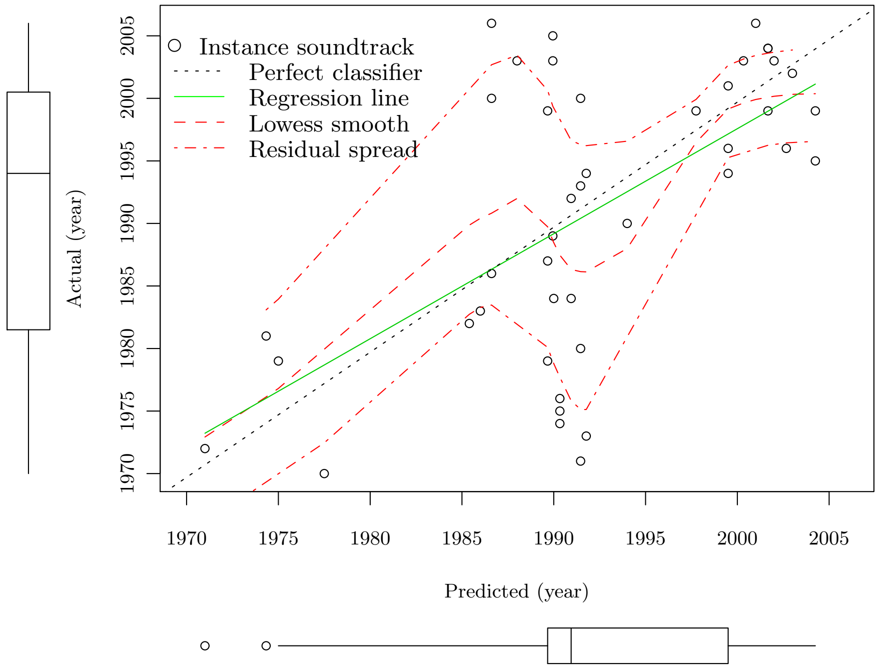


Fig. 4. Scatter plot of actual versus predicted years. Points show each instance soundtrack, the dotted line shows a hypothetical perfect classifier, the solid line shows linear regression on the predictions ($p < 0.001$, $r^2 = 0.36$), the dashed line shows loess non-parametric regression line on the predictions, and the dashed and dotted lines show smoothing applied to the root-mean-square positive and negative residuals from the loess line to display conditional spread and assymetry.

9 Conclusion

The combination of spectral characteristics (MFCC, spectral centroid, flux and rollof), and the zero-crossing rate as features for short segments provided sufficient discriminatory power for selecting between the seven sound classes defined in this study. The use of additional features, namely stereophonic ones (such as apparent source width [AWS]) could, in principle, decrease the rate of misclassification, in particular of hybrid classes involving environmental sound, which proved to be the source of most misclassifications. Environmental noise, aside from an overly broad definition, is especially heterogeneous. One commonality might involve its fundamental binaural decorrelation (wherefore most of its defining characteristics would be lost during pre-processing of a soundtrack to one channel only).

Amongst various classifiers, SMO provided the best predictive capabilities. At 74% accuracy, the model could certainly see improvements when compared with the state-of-the-art, which tends to hover around 90%. However, the use of three hybrid classes is rare in the literature; as argued in Sect. 5, the majority of errors involved combinations of classes that would not exist in a simple four-first-order-classes model, where misclassifications confusing a hybrid class with one of its constituent classes would disappear.

The use of logistic regression to obtain posterior class call probabilities aided in estimating the effectiveness of implicit audio segmentation based on adjacency rules. As a result, the sound class maps exemplified by Fig. 2 provide a novel way of visualizing segmentation results and their correctness.

Under the limited scope of the dataset, this study presented evidence that a sampling frequency of eight instances per decade is strongly correlated with the full sample. A tentative conclusion, therefore, is that for the purposes of segmenting and classifying cinematic soundtracks in the 1970–2006 period, it is likely that sampling eight instances per decade, for a total of 32 data points, is justified. It remains to be investigated whether these figures apply also to the full sound film period of 1930–present, or to other locales or agents of production.

The specific scope of this study disallowed further investigation of production locales and agents, but chronological prediction was scrutinized. For the three meta-features specifically employed (class coverage, scene frequency, and average scene length), it was difficult to build an accurate chronological model, although the system was able to learn how to place an unknown instance soundtrack in the 1970–2006 timeline, within an experimental error of approximately 7 years.

In the future, the feature space could be refined to include not only spectral characteristics, but also stereophonic ones. The SMO classifier appears to be adequate for the subsequent task of training and predicting unknown instances. The meta-feature space should be expanded, as it is clear that, while a step in the right direction, judging from the experimental error and correlation coefficients in the meta classification part of this study, these could be improved. Specifically, the meta-feature space should most likely include in-soundtrack variations of scene frequency, length, and class coverage.

References

1. Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20(1), 37–46 (1960)
2. Dorai, C., Mauthe, A., Nack, F., Rutledge, L., Sikora, T., Zettl, H.: Media semantics: Who needs it and why? In: Proceedings of the Tenth ACM International Conference on Multimedia. pp. 580–583. ACM Press, New York (2002)
3. Dorai, C., Venkatesh, S.: Bridging the semantic gap in content management systems: Computational media aesthetics. In: Proceedings of the First Conference on Computational Semiotics for Games and New Media. pp. 94–99 (2001)
4. Dorai, C., Venkatesh, S.: Bridging the semantic gap with computational media aesthetics. IEEE Multimedia 10(2), 15–17 (2003)
5. Grey, J.M.: Multidimensional perceptual scaling of musical timbres. The Journal of the Acoustical Society of America 61(5), 1270–1277 (May 1977)
6. Grey, J.M., Gordon, J.W.: Perceptual effects of spectral modifications on musical timbres. The Journal of the Acoustical Society of America 63, 1493 (May 1978)
7. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations 11(1) (2009)
8. Kohavi, R.: The power of decision tables. In: Proceedings of the Eight European Conference on Machine Learning. pp. 174–189. Springer-Verlag, London (1995)
9. Lacy, S.: Sample size in content analysis of weekly newspapers. Journalism and Mass Communication Quarterly 72(2), 336–45 (1995)
10. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics 33(1), 159–174 (1977)
11. Lu, L., Zhang, H., Jiang, H.: Content analysis for audio classification and segmentation. IEEE Transactions on Speech and Audio Processing 10(7), 504–516 (2002)
12. Lu, L., Zhang, H., Li, S.: Content-based audio classification and segmentation by using support vector machines. Multimedia Systems 8(6), 482–492 (2003)
13. Mermelstein, P.: Distance measures for speech recognition, psychological and instrumental. In: Chen, C.H. (ed.) Pattern recognition and artificial intelligence, pp. 374–388. Academic, New York (1976)
14. Metz, C., Gurrieri, G.: Aural objects. Yale French Studies 60, 24–32 (1980)
15. Platt, J.: Sequential minimal optimization: A fast algorithm for training support vector machines. In: Schölkopf, B., Burges, C.J.C., Smola, A.J. (eds.) Advances in kernel methods: Support vector learning, pp. 41–65. MIT Press, Cambridge (1998)
16. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Smola, A.J., Bartlett, P., Schuurmans, D., Schölkopf, B. (eds.) Advances in large margin classifiers, pp. 61–74. MIT Press, Cambridge (2000)
17. Riffe, D.: The effectiveness of random, consecutive day and constructed week sampling in newspaper content analysis. Journalism Quarterly 70(1), 133–39 (1993)
18. Riffe, D.: The effectiveness of simple and stratified random sampling in broadcast news content analysis. Journalism and Mass Communication Quarterly 73(1), 159–68 (1996)
19. Riffe, D., Freitag, A.: A content analysis of content analyses: Twenty-five years of journalism quarterly. Journalism and Mass Communication Quarterly 74, 515–524 (1997)
20. Riffe, D., Lacy, S., Drager, M.W.: Sample size in content analysis of weekly news magazines. Journalism and Mass Communication Quarterly 73, 635–644 (1996)

21. Saunders, J.: Real-time discrimination of broadcast speech/music. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing. vol. 2, pp. 993–996. IEEE (1996)
22. Stempel, G.H.: Sample size for classifying subject matter in dailies. Journalism Quarterly 29(2), 333–334 (1952)
23. Sundaram, H., Chang, S.F.: Audio scene segmentation using multiple features, models and timescales. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. vol. 6, pp. 2441–2444 (Jun 2000)
24. Truong, B.T., Venkatesh, S., Dorai, C.: Application of computational media aesthetics methodology to extracting color semantics in film. In: Proceedings of the Tenth ACM International Conference on Multimedia. pp. 339–342. ACM Press, New York (2002)
25. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing 10(5), 293–302 (2002)
26. Vapnik, V.N., Kotz, S.: Estimation of dependences based on empirical data. Springer-Verlag, New York (2006)
27. Zettl, H.: Contextual media aesthetics as the basis for media literacy. Journal of Communication 48(1), 81–95 (1998)

On the architecture of a real-time vision system for a SPL humanoid robot

Alina Trifan, António Neves, Nuno Lau, Bernardo Cunha

Universidade de Aveiro
IEETA/DETI
3810–193 Aveiro, Portugal
{alina.trifan, an, nunolau}@ua.pt, mbc@det.ua.pt

Abstract. Robotic vision is nowadays one of the most challenging branches of robotics. In the case of a humanoid robot, a robust vision system has to cope with all the constraints imposed by the hardware architecture and the movements of the robot. This paper presents a reliable implementation of a real-time modular vision system for a humanoid robot. The software that we present has been applied to the NAO robot developed by Aldebaran [1], which is currently the robotic platform used in the RoboCup Standard Platform League-SPL [2]. Our vision system can be used efficiently in real time for the detection of the objects of interest for a soccer playing robot (ball, field lines and goals). We also present an algorithm for self-calibration of the robot's camera parameters, as well as two support applications that can run on an external computer for calibration and debugging purposes. The experimental results that we present prove the efficiency of our approach in terms of both accuracy and processing time. Despite having been developed for the NAO robot, the modular design of the proposed vision system allows it to be easily integrated into other humanoid robots with a minimal number of changes, mostly in the acquisition module.

Keywords: Robotics, robotic soccer, computer vision, object recognition, humanoid robots, color classification.

1 Introduction

Humanoid robotics is the branch of robotics that focuses on developing robots that not only have an overall appearance similar to the human body but can also perform tasks that until now were strictly designated for humans. From taking care of the sick and/or elderly people, to playing football or even preparing for inhabiting a space shuttle, humanoid robots can perform some of the most common, yet unexpected tasks that humans undergo daily. Various humanoid robots are fully autonomous, which means that human interaction is needed only for their maintenance.

Similar to humans, the vision system of a robot is responsible for creating an accurate representation of the surrounding world and allowing the classification of objects so that they can be recognized the robot. Implementing a robust vision system for a humanoid robot is not an easy task since its performance is strongly influenced not just by the hardware architecture of the robot but mostly by its body movements. In the particular case of the standard robotic platform used in SPL, the NAO robot [3] comes equipped with a x86 AMD GEODE 500MHz CPU processor and 256 MB SDRAM / 2 GB flash memory. Focusing on the real-time performance of this robot, the poor processing capabilities have to be overcome by using low-computational image processing algorithms that can still deliver accurate results when it comes to object detections.

In SPL, NAO robots play soccer on a green field with an orange ball. The field lines are white and the two teams playing can have either red or blue markers. The red team will defend a yellow goal and the blue team will defend a sky-blue goal. The task of detecting the objects of interest is slightly eased by the color codes of each of the objects of interest. However, an efficient algorithm for object detections cannot solely rely on color information. During a SPL game, the playing field provides a fast-changing scenery in which the teammates, the opponents and the ball move quickly and often in an unpredictable way. The robots have to capture these scenes through their cameras and to discover where the objects of interest are located. Everything has to be processed in real time.

In this paper we provide a detailed description of a real-time modular vision system, based on color classification, for a humanoid robot. Having chosen robotic soccer as the context for achieving physical results of our implementation we present a robust solution for a reliable vision system, with emphasis on the real-time performance. We start by presenting an overview about the vision system, outlining its modularity, which makes it intuitively easy for being exported to other humanoid platforms. Then we propose an algorithm for self-calibration of the parameters of the camera. The algorithm uses the histogram of intensities of the acquired images and a white area, known in advance, for estimating the most important parameters of the camera, such as: exposure, gain and white balance. For the color classification algorithms a lookup table and horizontal or vertical scan lines are used. Finally, we present several validation criteria for a good detection of the objects of interest.

An overview of the work developed so far in this area of robotic vision was needed in order to better understand the context, the challenges and the constraints that robotic vision implies. The structure of the vision system that we are proposing was based on our previous experience in other robotic applications [4] as well as other related papers such as [5] and [6]. We consider that our approach is an important contribution mainly due to the modularity of our proposal, the real-time capability and the practically proven reliability of our system.

2 System Overview

The architecture of the vision system can be divided into three main parts: access of the device and image acquisition, calibration of the camera parameters and object detection and classification. Moreover, apart from these modules, two applications have been developed either for calibrating the colors of interest (NaoCalib) or for debugging purposes (NaoViewer). Figure 1 shows the proposed modular architecture. These two applications run on an external computer and communicate with the robot through a TCP module of the type client-server that we have developed. The current version of the vision system represents the best trade-off that we were able to accomplish between processing requirements and the hardware available in order to attain reliable results in real time.

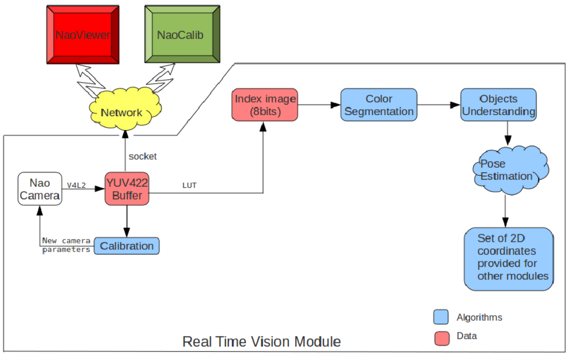


Fig. 1: Block diagram of the proposed vision system.

NAO has 2 identical video cameras that are located in the forehead and in the chin area, respectively. They provide a 640×480 resolution at 30 frames per second. The forehead camera can be used to identify objects in the visual field such as goals and balls, while the chin camera can ease NAO's dribbles during a soccer game. The native output of the camera is YUV422 packed. In the current version of the software only the lower camera of the robots is being used since it can provide more meaningful information about the surroundings. However, the software allows to switch between cameras in a small amount of time (29ms). This can be very useful when more evolved game strategies will be developed.

The camera is accessed using V4L2 API [7], a kernel interface for analog radio and video capture and output drivers. The V4L2 driver is implemented as a kernel module, loaded automatically when the device is first opened. The driver module plugs into the "videodev" kernel module. The access and acquisition module of the system that we are presenting is the only one that might suffer small changes when used with different humanoid robots. Different video devices connected by different technologies to the rest of the hardware can be accessed by making small adaptations to the module that we are proposing. All the other modules can be used as they are on any humanoid robot since their construction is very generic and is not related to any particularities that the NAO robot might have compared to other humanoids.

The calibration module is not continuously running on the robot because of the processing time limitations. It is run just once whenever the environment or the lighting conditions change, having the purpose of setting the parameters of the camera so that the images acquired give the best possible representation of the surrounding world. Details of the algorithm for self-calibration of the camera are presented in Section 3.

For the detection process, with the use of a look-up table, and by means of the OpenCV library [8], [9], the raw buffer can be converted into an 8-bit grayscale image in which only the colors of interest are mapped using a one color to one bit relationship (orange, green, white, yellow, blue, pink and blue sky, while gray stands for no color). The next step is the search for the colors of interest in the grayscale image, which we call an index image, by means of vertical or horizontal scan lines, and the formation of blobs from neighbour pixels that have the same color. The blobs are then marked as objects if they pass the validation criteria, which are constructed based on different measurements computed for each blob (bounding box, area, center of mass of the blob). The color segmentation and object detection are detailed in Section 4.

Having the possibility of running the vision module as a server, the two applications that we have developed, NaoCalib and NaoViewer can act as clients that can receive, display and manipulate the data coming from the robot. Thus, NaoViewer is a graphical application that allows the display both of the original image as well as the corresponding index image, containing validation marks for each object of interest that was found. This application was essential in terms of understanding what the robot “sees” since NAO does not have any graphical interface that allows the display and manipulation of images. NaoCalib is a very helpful application that we developed for the calibration of the colors of interest and it is presented in more details in Subsection 3.2.

3 Calibration of the vision system

Being still a color coded environment, during a SPL game the color of a pixel in the acquired image is a strong hint for object validation. Because of this, a good color classification is imperative. The accuracy of the representation of the colors in an image captured by the camera of the robot is related to the parameters of the camera such as: brightness, saturation, gain, contrast or white balance. By controlling these parameters relatively to the illumination of the environment we can acquire images that accurately represent the real world. Our approach is limited to these 5 parameters mainly because they prove to be sufficient for an accurate representation of the real world and also due to time considerations.

3.1 Self-calibration of the camera intrinsic parameters

The use of the NAO camera in auto-mode has raised several issues which made the segmentation and validation of objects hard to be performed. By using the camera in auto-mode the images acquired were far from being accurate and the colors of interest were not represented in the same way that the human eye could perceive them. Thus, the classification of colors was difficult to perform and the robots’ perception of colors was distorted when compared to the human one.

We propose an algorithm for self-calibration of the camera that is both fast and accurate, and requires a minimum amount of human intervention (presented in Fig. 2). The algorithm uses the histogram of intensities of the acquired images for calculating some statistic measurements of the images which are then used for compensating the values of the gain and exposure by means of a PI controller. The choice of a PI controller relies again on the attempt of making every process or computation as fast as possible. Even though at this stage of our developments the calibration module does not run in real-time, efforts are being made in this direction. Moreover, since there is not any mathematical relation between the statistic measurements of the image that we are using, and the camera parameters, the PI controller allows the compensation of the latter based on the error between the measurement computed and its idealistic value. For the calibration of the white balance, a white area, whose location in the image is known in advance, is used. The human intervention is only needed for positioning a white object in the predefined area and for determining in which situations the calibration algorithm has to be run. The algorithm only needs an average number of 20 frames to converge and the processing time of each frame is approximately 300ms.

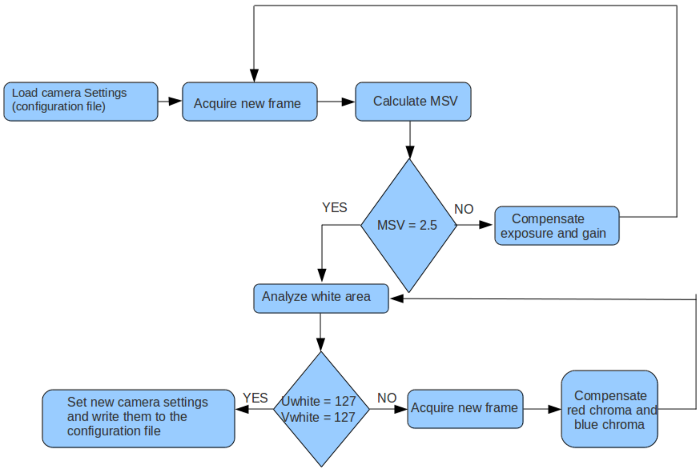


Fig. 2: Diagram of the proposed algorithm for self-calibration of the camera.

The intensity histogram of an image, that is, the histogram of the pixel intensity values, is a bar graph showing the number of pixels in an image at each different intensity values found in the image. For an 8-bit grayscale image there are 256 different possible intensities, from 0 to 255. Image histograms can also indicate the nature of the lighting conditions, the exposure of the image and whether it is underexposed or overexposed. The histogram can be divided into 5 regions. The left regions represent dark colors while the right regions represent light colors. An underexposed image will lean to the left while an overexposed one will be leaning to the right. Ideally most of the image should appear in the middle region of the histogram.

From the gray level histogram the Mean Sample Value (MSV) can be computed based on the following formula and it represents a useful measure of the balance of the tonal distribution in the image: $MSV = \frac{\Sigma_{j=0}^{4}(j+1)x_j}{\Sigma_{j=0}^{4}x_j}$, where x_j is the sum of the gray values in region j of the histogram. The histogram is divided into five regions. The image is considered to have the best quality when the MSV ≈ 2.5. MSV is a mean measure which does not take into account regional overexposures and underexposures in the image. The values for the gain and exposure are compensated with the help of the PI controller until the value of the MSV for the images acquired is ≈ 2.5.

For the calibration of the white balance, the algorithm that we are proposing assumes that the white area should appear white in the acquired image. In the YUV color space, this means that the average value of U and V should be close to 127 when both components are coded with 8 bits. If the white-balance is not correctly configured, these values are different from 127 and the image does not have the correct colors. The white-balance parameter is composed by two values, blue chroma and red chroma, directly related to the values of U and V.

The parameters of the PI controller were obtained experimentally, based on the following reasoning: first, the proportional gain is increased until the given camera parameter would start oscillating. The value chosen for the proportional gain will be 70% of the value that produced those oscillations and the integral gain is increased until the convergence time of the parameters reaches an acceptable value of around 100ms.

An exemple of the use of the proposed algorithm is presented in Fig. 3. As we can see, the image on the right has the colors represented in the same way that the human eye perceives them. On the oposite, in the image on the left the colors are too bright and a distinction between black and blue is difficult to be made.

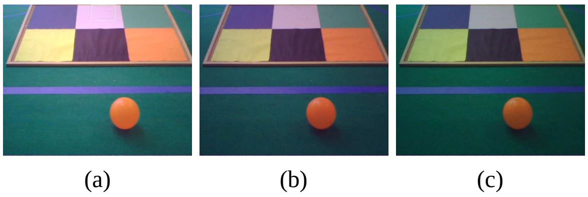


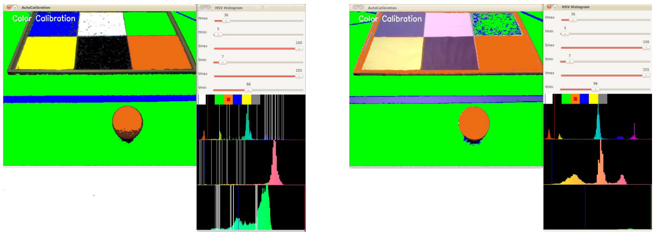
Fig. 3: On the left, an image acquired with the camera used in auto-mode. The white rectangle, in the top middle of the image, represents the white area used for calibrating the white balance parameters. In the middle, an image acquired after calibrating the gain and exposure parameters. On the right, the result of the self-calibration process, after having also the white balance parameters calibrated.

3.2 Calibration of the colors of interest

Along with the calibration of the parameters of the camera (presented in the previous subsection), a calibration of the color range associated to each color class has to be performed whenever the environment or the illumination conditions change. These two processes are co-dependent [10].

NaoCalib is an application created after a model used by CAMBADA, the RoboCup Middle-Size League team of the University of Aveiro [11]. It is used for the manual calibration of the colors of interest and it allows the creation of a configuration file that contains the Hue, Saturation and Value minimum and maximum values of the colors of interest. Figure 4 (a) presents an example of its use. The configuration file is a binary file that apart from the H, S and V maximum and minimum value also contains the current values of the parameters of the camera.

The interface is based on the histograms of the three color components (Hue, Saturation and Value) and it allows the selection of the color range for each color class with the help of sliders. For each of the colors of interest, a set of pixels corresponding to the color class that is being calibrated can be selected with the help of the mouse. The color classes correspond to the colors of interest, which are: white, green, orange, yellow, blue, blue-sky, magenta and black. Based on this and with the help of the H, S and V histograms the user can manipulate the sliders for selecting the maximum and minimum values of the three components for each color class. It is then exported to the robot and loaded when the vision module starts.



(a) (b)

Fig. 4: On the left a color calibration after the intrinsic parameters of the camera have converged. On the right, the result of color classification considering the same range for the colors of interest but with the camera working in auto-mode. Most of the colors of interest are lost (the blue, the yellow, the white and the black) and the shadow of the ball on the ground is now blue, which might be confusing for the robot when processing the information about the blue color.

4 Object detection

For an SPL soccer player robot the objects of interest are: the orange ball, the white lines of the field and the yellow and blue goals. In this section we present our approach for the detection and validation of the objects of interest, based on color segmentation followed by blob formation and measurements computations for the validation of the blobs.

4.1 Look-up table and the image of labels

Color classes are defined with the use of a look-up table(LUT) for fast color classification. A LUT represents a data structure, in this case an array used for replacing a runtime computation with a basic array indexing operation. This approach has been chosen in order to save significant processing time. The image acquired in the YUV format is converted to an index image (image of labels) using an appropriate LUT.

The table consists of 16,777,216 entries (2^{24}, 8 bits for Y, 8 bits for U and 8 bits for V). Each bit expresses whether one of the colors of interest (white, green, blue, yellow, orange, red, blue sky, gray - no color) is within the corresponding class or not. A given color can be assigned to multiple classes at the same time [12]. For classifying a pixel, first the value of the color of the pixel is read and then used as an index into the table. The 8-bit value then read from the table is called the ”color mask” of the pixel.

The resulting index image is a grayscale image with the resolution of 320×240 pixels. A smaller resolution was obtained with the purpose of reducing the classifying time and further decreasing the time spent on scanning and processing the image. The reduced resolution was obtained by using a subsampling approach. By using the YUV422 packed format of the image, we obtained a subsampling of the image across the image line. For the Y sample, both horizontal and vertical periods are 1 while for the U and V samples the horizontal period is 2 and the vertical one is 1. This means that the two chroma components are sampled at half the sample rate of the luma: the chroma resolution is halved. Then, by type casting the YUV422 buffer [5], which is an unsigned char buffer to an integer one, thus making the reading of 4 bytes at the same time possible, we ignore one column in 4 of the image, by reading only half of the luminance information (Fig. 5). Even though for the human eye the luminance is the component of a color that has more significance, this is not valid in the case of robotic vision. Moreover, using this approach we access 4 times less the memory. Further image processing and analysis will be performed on the index image. The algorithm for the type casting and construction of the index image is depicted next:

Algorithm 1 Algorithm of the type casting of the unsigned char buffer to an integer one and the construction of the index image.

```
$unsigned\, int\ *b = (unsigned\, int*)YUVbuf$
for $i = 0; i < nColsIndex \times nRowsIndex; i++$ do
    $lutPos = (b[i]\ \&\ 0x00FFFFFF)\ >>\ 8$
    $IndexImageData[(r \times nColsIndex + c)] = lut[lutPos]$
    $c++$
    if $c \geq nColsIndex$ then
        $i+ = nColsIndex$
        $c = 0$
        $r++$
    end if
end for
```

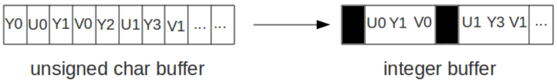


Fig. 5: An illustration of the conversion of the unsigned char buffer to an integer one, allowing thus the reading of 4 bytes at the same time. Using this approach we can obtain a reduced resolution of the images.

4.2 Color segmentation and blob formation

Having the colors of interest labeled, scan lines are used for detecting transitions between colors of interest [13]. For the detection of posts and field lines vertical search lines are used, while for the detection of the ball both vertical and horizontal scan lines can be used. Both for vertical and horizontal scan lines, it is possible to configure the number of scan lines to be used in order to reduce the processing time. Both types of scan lines start in the upper left corner of the image and go along the width and the height, respectively, of the image. Transitions of the type green-color of interest-green are looked for. The information about the green color is used as a validation that we are searching for one of the colors of interest only within the limits of the soccer field.

While scanning the image in search of a color of interest, with every new row/column, a new scan line of length 0 is being initialized. The algorithm processes all the pixels of a row/column and for every green pixel found a counter called GPB is incremented

(GPB stands for "green pixels before"). If a pixel of the color in search is being found, a counter called CP ("color pixel") is incremented. Since the algorithm is based on transitions of the type green-color of interest-green, after finding a given number of color pixels, a counter for the following green pixels (GPA) is also incremented every time a new green pixel is found (Fig. 6).

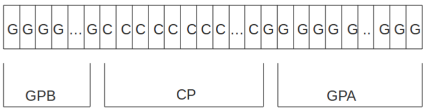


Fig. 6: Transitions between green pixels (G) and pixels of one of the colors of interest (C).

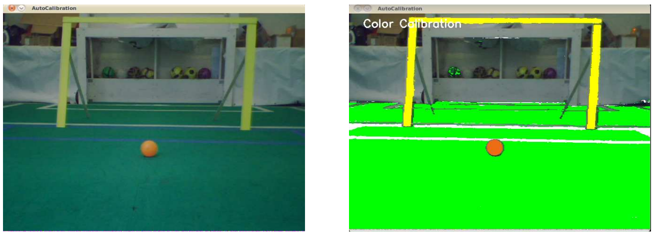
For each scan line, the run length information of each color of interest is saved if all three counters are larger than predefined thresholds. The next step is the formation of blobs from the run-length information about close parallel scan lines. By calculating the distance between the center of mass of consecutive scan lines we can decide whether or not they are parallel. If they are parallel and the distance between them is smaller than a predefined threshold the scan lines are considered as being part of the same blob and they are merged together.

Having the blobs formed, several validation criteria are applied in the case of the orange ball and of the blue or yellow goals, respectively. In order to be considered a yellow goal, a yellow blob has to have the size larger than a predefined number of pixels. In the situation in which the robot sees both posts of the goals, the middle point of the distance between the two posts is marked as the point of interest for the robot. In the case when just one of the posts is seen, its mass center is marked. For the validation of the ball, the areas of the orange blobs are calculated and the blob validated as being the ball will be the one that has the area over a predefined minimum value and it is closest to the robot. In order to calculate the distance between the robot and the orange blobs without having an estimation of the pose of the robot, the center of mass of the robot is considered to be the center of the image.

4.3 Results

In this subsection we present several images that show every step of our algorithms for object detections: from acquiring a frame, calibrating the color of interest, forming the index image with all the colors of interest labeled, to color segmenting and detection of the objects of interest (in this case the objects of interest were the orange ball and the yellow goals).

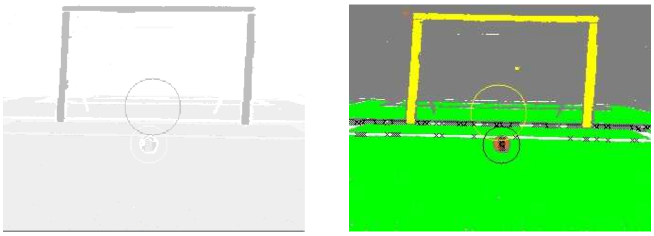
The first step is acquiring an image that can be displayed with the use of our NaoViewer application (Fig. 7(a)). Having an image acquired, we move on to classifying the colors of interest with the help of the NaoCalib application, as it was previously described in Section 3.2. The result of the color classification can be seen in Fig. 7(b).



(a) (b)

Fig. 7: On the left an image captured by the NAO camera. On the right, the same image with the colors of interest classified.

The next step of our algorithm, is the conversion of the RGB image into an index image. Figure 8(a) presents the index conversion of the previous RGB frame while Figure 8(b) represents the equivalent "painted" image according to the labels in the grayscale image.

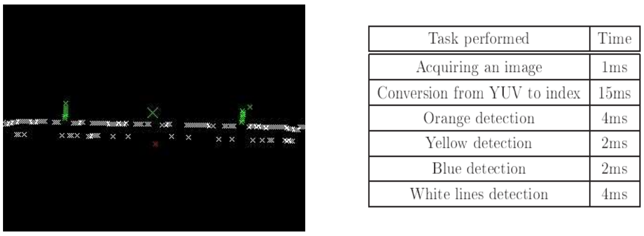


(a) (b)

Fig. 8: On the left, the index image. On the right, the equivalent image "painted" according to the labels in the grayscale image.

The painted image is a 3-channels RGB image of the same resolution as the index image. The index image is scanned and for each pixel labeled as having one of the colors of interest, the color of the corresponding pixel in the RGB image is set as having the respective color of interest. If there are pixels that do not have any of the colors of interest they will be painted as gray. Both images already contain the markers that identify the objects of interest. The black circle stands for a valid ball while the yellow circle is a marker for the yellow goals. The yellow circle is constructed having the center in the middle of the distance between the two yellow goals. The black crosses are markers for the white lines of the field.

The low processing times that we obtained (Fig 9(b)) allow us to use the camera at 30 fps while processing the images in real-time and achieving reliable results. These results are also influenced by the internal structure of the NAO robot.



Task performed	Time
Acquiring an image	1ms
Conversion from YUV to index	15ms
Orange detection	4ms
Yellow detection	2ms
Blue detection	2ms
White lines detection	4ms

(a) (b)

Fig. 9: On the left, the result of the object detection algorithms. On the right, a table with the processing times spent. The total processing time of a frame is 28ms, which allows us to use the camera at 30fps.

The performance of the vision system in terms of percentage of valid balls comparing to the number of frames acquired by the robot is recorded in Table. 1. The percentages represent frames in which the ball is seen and appropriately detected. The percentage of valid balls in almost all of the scenarios is high, the more critical situation being the one in which the robot is moving, as expected. Due to its type of locomotion, still images that give a good representation of the surrounding world are hard to acquire.

Scenario	Percentage
Robot dribbling the ball	93%
Robot stopped observing the ball	99%
Ball sent towards an imobile robot	99%
Robot moving towards the ball	30%

Table 1: Percentage of ball detections compared to the total number of frames acquired under various scenarios.

Regarding the auto-calibration procedure, Table 2 presents the time spent in the performance of the most important tasks of the algorithm. The main steps of the process are: reading the value of an intrinsic parameter (Read), setting the value of an intrinsic parameter of the camera (Write), calculating the MSV (MSV), calculating the error between the MSV value of the acquired frame and the desired value of 2.5 (MSV), compensating gain (Gain), exposure (Exposure), white balance values (Blue and Red) and calculating the average U and V for the white area (U and V).

Mode	Read	MSV	Gain	Exposure	U	V	Blue	Red	Write	Total
Auto	0ms	33ms	15ms	15ms	1ms	1ms	6ms	5ms	0ms	10s
0	0ms	34ms	17ms	18ms	1ms	1ms	6ms	6ms	0ms	46s
Max	0ms	34ms	15ms	16ms	1ms	1ms	6ms	6ms	0ms	1min

Table 2: Processing times spent by the main tasks of the self-calibration module, when the camera is started at different values of the intrinsic parameters. In the first situation, the camera starts in auto-mode, in the second situation the camera starts with all parameters set to 0. Finally, in the third situation the camera starts with all the intrinsic parameters set to their maximum values.

5 Conclusions and Future Work

This paper presents a real-time reliable vision system for a humanoid robot. From calibrating the intrinsic parameters of the camera, to color classification and object detection the results presented prove the efficiency of our vision system.

Future developments of our work include more validation criteria for the ball detection based on circular histograms and classifiers training which are more generic and are not color dependent. Also the algorithm for finding transitions from one color of interest to another will be further applied for the detection of the team markers.

References

1. Aldebaran official website. `http://www.aldebaran-robotics.com/`. Last visited January, 2012.
2. RoboCup Standard Platform League official website. `http://www.tzi.de/spl`. Last visited January, 2012.
3. D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier. The NAO humanoid: a combination of performance and affordability. *ArXiv e-prints*, July 2008.
4. Antonio J. R. Neves, Armando J. Pinho, Daniel A. Martins, and Bernardo Cunha. An efficient omnidirectional vision system for soccer robots: from calibration to object detection. *Mechatronics*, 21(2):399–410, March 2011.
5. Piyush Khandelwal, Matthew Hausknecht, Juhyun Lee, Aibo Tian, and Peter Stone. Vision calibration and processing on a humanoid soccer robot. In *The Fifth Workshop on Humanoid Soccer Robots at Humanoids 2010*, December 2010.
6. H. Lu, Z. Zheng, F. Liu, and X. Wang. A robust object recognition method for soccer robots. In *Proc. of the 7th World Congress on Intelligent Control and Automation*, Chongqing, China, June 2008.
7. Video For Linux 2 API Specifications. `http://v4l2spec.bytesex.org/`. Last visited June, 2011.
8. Open Source Computer Vision Library (OpenCV). `http://sourceforge.net/projects/opencvlibrary/`. Last visited January, 2012.
9. Gary Bradski and Adrian Kaehler. *Learning OpenCV*. O'Reilly, first edition, September 2008.
10. P. M. R. Caleiro, A. J. R. Neves, and A. J. Pinho. Color-spaces and color segmentation for real-time object recognition in robotic applications. *Revista do DETUA*, 4(8):940–945, June 2007.
11. A. Neves, J. Azevedo, N. Lau B. Cunha, J. Silva, F. Santos, G. Corrente, D. A. Martins, N. Figueiredo, A. Pereira, L. Almeida, L. S. Lopes, and P. Pedreiras. *CAMBADA soccer team: from robot architecture to multiagent coordination*, chapter 2. I-Tech Education and Publishing, Vienna, Austria, In Vladan Papic (Ed.), Robot Soccer, 2010.
12. James Bruce, Tucker Balch, and Manuela Veloso. Fast and inexpensive color image segmentation for interactive robots. *2000 IEEE-RSJ Int. Conf. Intelligent Robots and Systems (IROS'00)*, 3:2062–2066, 2000.
13. A. J. R. Neves, D. A. Martins, and A. J. Pinho. A hybrid vision system for soccer robots using radial search lines. In *Proc. of the 8th Conference on Autonomous Robot Systems and Competitions, Portuguese Robotics Open - ROBOTICA'2008*, pages 51–55, Aveiro, Portugal, April 2008.

Surface modelling and prototyping using a touch interface

Tiago Marques

Faculdade de Engenharia da Universidade do Porto,
Rua do Dr. Roberto Frias, s/n
4200-465 Porto, Portugal
pro11013@fe.up.pt

Abstract. The design and prototyping design stage and the final model development stage of any product are still being treated as separate processes with different methodology. This is because people use highly distinct tools in both stages. This barrier causes delays and obstructions in the development of a project. By approximating the tools used, we may be able to enhance this process. In this paper, an application is presented that enables users to model a 2D surface composed of metaballs through a touch-based interface. This application may enable virtual models to be made early in the design process, joining these two stages of development. The application uses the Euclidean Distance Transform to simulate a metaball system and render it. It supports both single- and multi-touch through the Tangible User Interface Objects (TUIO) protocol. It interprets three types of modelling and implements an "undo" action, common in several modelling systems.

Keywords: Multi-touch, Modelling, Early-stage prototyping, Prototype design, Touch-based Interfaces

1 Introduction

Computer generated images have been increasing its popularity among many industries [1]. Nowadays, these techniques are useful, sometimes indispensable, to some of those industries, not just for creating or modifying objects, for example in a film production [1, 2], but also as planning and organising tools. In either case, these tools are mainly modelling tools, called CAD systems, which allow designers and architects to create and iterate virtual scenes.

While developing a product that requires these tools, the modelling stage of the product usually follows a more creative stage. This first stage allows designers the freedom needed to explore different options and ideas with ease [3]. Therefore, modelling using a CAD system is usually only an intermediate or final step towards creating a virtual scene. The development of a prototype is often divided in these two different stages due to the lack of professional standardised and optimised solutions that enable both freedom for rapid prototype development and the use of a CAD system for exact modelling [4].

1.1 Modelling Tools and Natural User Interfaces

The modelling systems have been growing in functionality and effectiveness throughout the years [5–7]. CAD tools are now precise modelling tools designed for sculpting detailed virtual scenes. However, these tools may require some time to comprehend and function with due to their advanced features only being available to professional and knowledgeable users [8, 9].

As a consequence of the associated learning curve, CAD tools do not allow users to concentrate on their creativity but rather on their productivity while building a model. This fact pushes designers to use a more versatile piece of equipment which allows them to focus on their ideas [10] and not so much on the exactness of the model itself. Hence, tools such as the pen and paper are usually used in the initial stages of these projects while modelling systems are only used after the initial paper prototypes are created [11].

The problem of time-wasting and lack of collaboration between the two stages of a model development may be solved using natural user interfaces (NUIs). By reducing the experience needed to operate a CAD system, designers with no modelling experience would be able to create virtual models more freely [11].

The concept of NUI refers to a user interface, that can be, in practical terms, imperceptible and natural to use [12]. In regular user interfaces, devices need to be used to interact with a system and a learning period is usually needed to be able to use that device accurately. By contrast, in natural user interfaces, the device that separates the user and the system should be as unobtrusive as possible or hidden so that the user does not notice it.

NUIs such as touch detection, voice recognition, haptic input or gesture recognition usually try to replace the keyboard and mouse [13, 14]. In this case in particular, these natural interfaces should be used to ease the modelling experience of inexperienced users that wish to focus on ideas rather than virtual modelling.

1.2 Description of the Work

The work that endorses this paper focuses on the importance of creating tools that equally offer modelling functionality and creative freedom to the user. The purpose of this is to aid in early-stage prototyping and remove the need for different tools for different stages of the development.

This project aims to create a 2D multi-touch CAD environment for rapid prototype development based on a Play-Doh modelling approach. In this sense, this work attempts to lessen the barrier of the two different development stages, by means of NUIs.

1.3 Related Work

In terms of virtual modelling interaction, there has been some work performed in order to lessen the load of CAD interfaces, which, by default, can be very overloading.

Döllner *et al* [15] tried to create a framework for modelling and animation that reduces the load of the program's interface. This project, in spite of not involving the use of natural input devices, attempted to create a working environment that quickens the modelling process.

Scali *et al* [16] experimented with a 6 degrees of freedom haptic feedback device for 3D modelling and its comparison in quantitative and qualitative terms. Despite the main objective being the attempt to facilitate the learning curve of current modelling systems, this work ultimately tries to approximate the stages of the development process.

On a completely different approach – paper-designer centred instead of CAD-designer centred – Anderson *et al* [17] were able to put together a system that converts electronic Lego pieces put together into a virtual 3D model. This model can be worked on at a later stage by a less creative but more trained CAD designer.

Schmidt *et al* [18] however, followed a closer methodology to the one used in this paper. In their work, they implemented an object modeller which creates 3D objects out of 2D lines (through linear sweeps and revolutions) and shape them using push, pull and cut operations. Much like the work that this paper endorses, Schmidt *et al* [18] also focus on the combination of virtual modelling on computer at the initial stage of a project, although the main difference lies on the type of usage. While Schmidt *et al* [18] developed a system to be used with a stylus to draw new objects or parts of an object, our work is based on a more Play-Doh-like approach.

1.4 Structure of the Document

This document consists of 6 sections. Section 1 is an introductory part to contextualise the reader and justify the importance of the work at hand. The methodology is presented in Section 2 and focuses on how the work was performed and the tools used. Section 3 is the last of the descriptive part of this paper and shows how the prototype application is implemented and why. A results Section (Section 4) is presented where the developed application is shown, followed by a discussion and conclusions Section (Section 5) in which some alternative paths are explored and conlusions are given. Lastly, Section 6 addresses the future work.

2 Methodology

For the work described in this paper, some key concepts needed to be researched. Namely, the object or objects in the application that form the surface to be modelled by the user and the interaction with the user. The latter implied the study of touch frameworks available for touch detection and processing.

2.1 Surface to be modelled

There are several ways to simulate a Play-Doh-like surface on a computer screen in order to try to mimic the experience of modelling on a tangible (real) object. Specifically, objects whose form changes according to other objects or to some environment settings can be used. In general they are called meta-objects [19], though the instances of these objects may vary so they might be referred to as metaballs or metacubes, for example. These objects are ultimately composed of polygon meshes that may need to be recalculated every time the environment changes so that these changes can be taken into account for each object. Roughly, each object possesses:

- a core position - *x* and *y*, for 2D environments;
- a size value (or several) - such as radius or height, depending on the shape of the meta-object;
- a threshold value - usually to combine the objects, as a curvature or minimum distance for intra-object interaction.

For this case in particular, it was decided to use metaballs with the same size and threshold values. The idea behind this approach would be to create a surface by combining multiple metaballs adjacent to each other. As a consequence, the surface's contours could easily be shaped by changing some metaballs' position through the user's input.

2.2 User Input

After deciding on how to create the surface for the user to model, the interaction method was in due. Interaction through touch was chosen above others to serve as the main input for model editing. This decision was based on the notion that people prefer modelling tangible objects with their own hands rather than through other means [20].

The process that involves detecting a person's touch on a surface and linking it with an application roughly involves the following steps:

- touch detection through hardware;
- sending the information from the input device to the application;
- process the touch information on the application's domain.

Detection and Transmission

Touch can be detected through several mechanisms. In this work, it was decided to use a touch and tangible objects information protocol named TUIO [21]. This protocol eliminated the need for hardware-specific development so this subject did not become a relevant issue.

The TUIO protocol [21] is progressively being used as a standard for sending and receiving touch information. An application that implements a TUIO client and handles its information correctly supposedly works on all devices that implement a TUIO server for sending information. Consequently, it also makes application development and distribution easier by allowing developers to focus more on developing the application itself.

Touch Handling

From the software's point of view, touch interaction can be considered single- or multi-touch. In order to develop the application that would benefit from both single- and multi-touch interaction, a versatile and extensible touch handling framework needed to be chosen. The Kivy framework was then chosen, due to its speed, flexibility and its API's ease.

Kivy [22] is a Python-based framework for multi-touch applications' development that supports both touch and tangible objects interaction through TUIO. Through the Widget class (and its subclasses), the developer can extend three functions that are called when the events 'touch down', 'move' and 'up' are created. Furthermore, the events and functions are called on a top-down approach. This way, the developer has full control over how the application processes each touch. This framework is open-source, cross-platform and it uses OpenGL ES 2 for rendering.

3 Implementations

By using the Kivy framework as the basis of the implementation, a program was developed in Python that rendered the 2D metaball system on an 800 by 800 pixel screen. As mentioned in Section 2.2, the program's interaction is touch-based and allows a straightforward use to any device that implements a TUIO server.

3.1 Metaball System

The metaball system that creates the 2D image of the object to be modelled was initially based on a python algorithm by Hannu [23]. The algorithm was created as an optimisation of the naïve alternative to building a 2D metaball system, also described by Hannu [23]. The basic idea behind the algorithm, and the reason for its performance, is that, for each metaball, it searches for the border of the current metaball and then iterates through the border, considering every other metaball's influence over the part of the border it is drawing.

There are some flaws in Hannu's work [23] which were revealed when implemented in our system. The first being the fact that processing time increases exponentially with the number of metaballs to be processed; the second being that if a metaball does not have a border, *i.e.* it is surrounded by other metaballs (exemplified by metaball A in Figure 1), the program can become computationally heavy and may take too long to realise that its border should not be drawn.

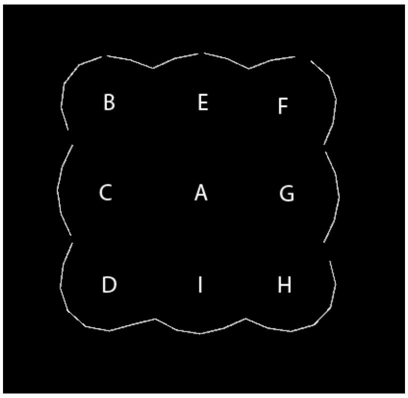


Fig. 1. Location of 9 metaballs which form a single body. In this figure it is visible that metaball A is the only one that does not own part of the joint border. In metaballs C and G, their contribution to the border is clearly seen due to discontinuity of the border.

As the disadvantages to the performance were substantial, it was decided that a different algorithm had to be used, one that mimicked a metaball system and could still solve some of Hannu's [23] issues. The algorithm that was used in the final version of the prototype was the Euclidean Distance Transform (EDT) [24]. Roughly, this algorithm calculates the distance between elements, in this case metaballs, creating a bitmap image with higher values (coloured blue) on spots near these elements and lower when these spots are more distanced (coloured red). An example of the algorithm being used is shown in Figure 2. EDT scans the entire area (in Figure 2, a 1000 by a 1000 pixel area) twice (downwards and then upwards), considering only the metaballs it has seen previously during the pass. The advantage of this particular algorithm is that performance does not fluctuate with the increase on the number of metaballs.

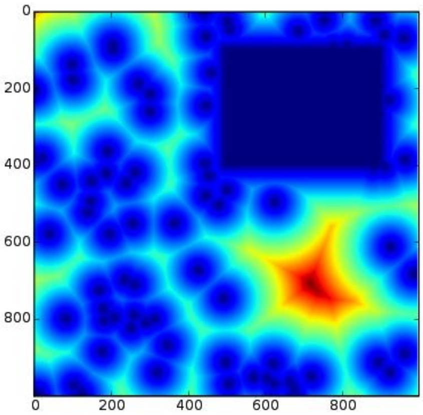


Fig. 2. Example of EDT usage on a 1000 by a 1000 pixel screen. In this figure, it is apparent that each meatball is resided on locations with high concentration of the colour blue.

However, some adaptations to EDT were necessary to implement it. The main code that generates the bitmap is nearly the same and the bitmap is used to create a texture in Kivy. That texture is later applied to the main screen and updated when the framework detects a new touch event. Figure 3 shows the prototype's interface with 400 metaballs being calculated.

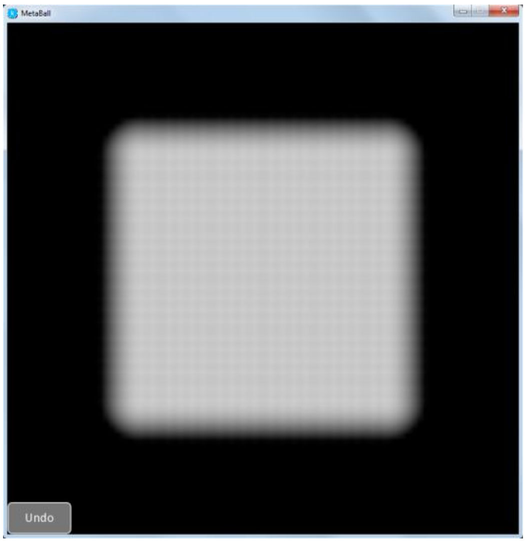


Fig. 3. Final prototype's interface with EDT meatball system rendering. In this figure, a 20 by 20 metaball representation is shown, using a 800 by 800 pixel screen.

3.2 Interaction Mechanisms

The application described in this work mainly focused on a touch-based interface. The user is able to model the surface of the object freely and using single- or multi-touch. The actions implemented in this work include the ability to:

- model the surface without touching it;
- model the surface by dragging a single metaball through the surface;
- drag a single metaball without affecting the surface.

Concretely, if the user simply moves either finger from any empty space towards the surface, the surface will mould according to the position and direction of the touch. If, on the other hand, the user chooses to click a metaball, it attaches itself to the touch. Whether it is a single- or a double-click, the fact will determine if the metaball helps shaping the rest of the surface while moving.

As previously mentioned in Section 2.2, touch is native to the framework Kivy, so the implementation is fairly simple. A touch event is generated when a finger touches the surface, moves or is lifted. For each of these events, their respective functions are called. The following pseudo-code shows what happens when each event generated:

```
# function to be called when
# a touch gets detected
def on_touch_down(self, touch):

  for each metaball:
    if touch is near metaball:
      grab_metaball()

  move_remaining_metaballs()
  build_texture()

# function to be called when
# a touch moves position
def on_touch_move(self, touch):
  move_remaining_metaballs()
  build_texture()

# function to be called when
# a touch is lifted
def on_touch_up(self, touch):
  for each metaball:
    if this_ball_is_grabbed:
      ungrab_metaball()
```

Additionally, an “undo” action was also implemented. Like any tool nowadays, this function allows the user to go back on previous steps. This was done by simply storing the positions of the metaballs that the last lifted touch modified. By doing so, it is ensured that if the user wishes to undo an action, it would not undo the modifications of other simultaneous touches, consequently leading to a much more accurate “undo” action when using multi-touch. The modification to the previously explained functions was simple and it is shown in bold in the code bellow:

```
# function to be called when
# a touch gets detected
def on_touch_down(self, touch):
  record_touch_id_and_metaballs()
  # then perform all other commands
  # shown before in the same function

# function to be called when
# a touch moves position
def on_touch_move(self, touch):
  save_metaballs_state()
  move_remaining_metaballs()
  record_metaballs_moved()
  build_texture()
```

```
# function to be called when
# a touch is lifted
def on_touch_up(self, touch):
  record_all_metaballs_that_were_changed()
  # then perform all other commands
  # shown before in the same function
```

When the user then requests the "undo" action – either by pressing the "undo" button, shown in Figure 3, or by pressing the "backspace" key – the metaballs associated with the last touch lifted will be returned to their original positions.

4 Results

In this work, an approach to modelling 2D objects was presented, based on a virtual Play-Doh-like object. A prototype was successfully implemented, based on these premises and on further research conducted, using the framework Kivy and EDT for rendering the object. An example usage of the application is shown in Figure 4 where the trace of the user's fingers modelling the surface can be seen. It is possible to see on the right side of the figure, the user's finger moving the location of one metaball.

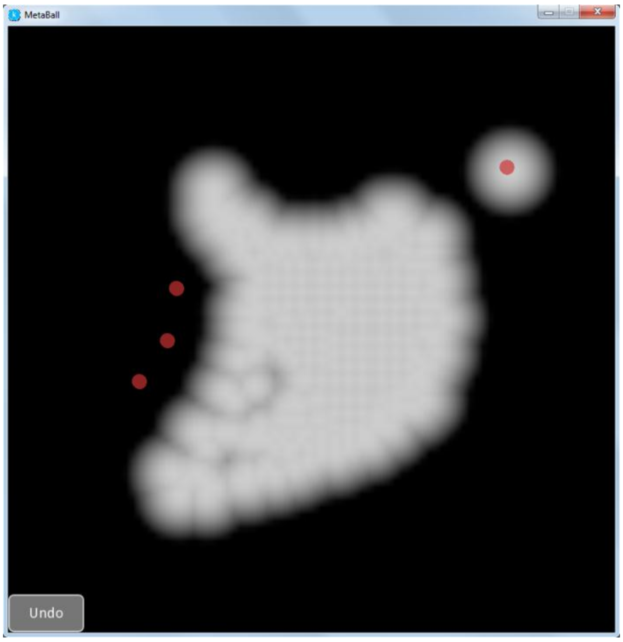


Fig. 4. Example usage of the final version of the prototype. In this figure, we can see the trace of the user's fingers through the red circles. The user is modeling the surface with his left hand, as seen in the left side of the figure, while simultaneously dragging one meatball, to adjust the surface, as seen on the right side of the figure.

5 Discussion and Conclusions

During the execution of this work, decisions were made regarding the software tools and protocols, the actions to be implemented by the prototype and some implementation decisions.

In respect to the Kivy framework, it was chosen due to some previous work performed by the author [25] that compares this Python-based multi-touch framework to other similar frameworks. Specifically, the comparison showed that this framework was the most flexible and future-oriented.

The actions implemented on the final prototype were suggested by the author through experience on modelling tools such as Blender, 3D Studio Max, SketchUp and AutoCAD.

As for the implementation itself, the decision to approximate the initial metaball system by an EDT implementation was purely for performance reasons. Nevertheless, due to the similarity of the visual results obtained, this decision seems to have improved the quality of the results obtained.

As a conclusion, the multi-touch capabilities of the framework and of the application developed on this project, may allow modellers to try out ideas in a more intuitive and button-free way. This prototype may enable early stage prototyping through virtual technologies. Additionally, the type of interaction created can be considered as suitable for this type of work, considering the author's experience.

6 Future Work

In addition to the ones implemented in this application, other instructions can be added to the prototype, such as the "cut" function, to easily cut part of the surface. As such, a study and evaluation of the proposed interaction should be considered as future work, in order to improve the user experience and power designers with more tools.

This system could also be evolved into a 3D environment. However, this approach would require a carefully designed interaction to suit the users' needs. For instance, camera control would be crucial. In this scenario, the interaction mechanisms would benefit from a gesture recogniser, like Kinect [26]. Additionally, tangible objects, representing specific functions in the application, can also be considered for future work. This type of environment may have the potential to become a stepping stone to developers aiming to build modelling tools based on natural user interfaces.

Acknowledgements

I would like to acknowledge the contributions made by Professor Rui Rodrigues PhD. to this project in terms of inspiration, know-how and some optimisations of the final prototype.

References

1. Antonsen, T.: Sublime Pixels: Exploring the Audience Experience in Digital Special Effects Cinema, http://researcharchive.vuw.ac.nz/handle/10063/1075, (2009).
2. Mao, C., Qin, S.: Sketching-out virtual humans: from 2D storyboarding to immediate 3D character animation. Proceedings of the 2006 ACM SIGCHI. (2006).
3. Huang, C.C.S.: Conceptual Modeling Environment (COMOEN). Computer-Aided Architectural Design Futures (CAADFutures) 2007. 31–43 (2007).
4. Li, W., Ong, S., Fuh, J., Wong, Y., Lu, Y.: Feature-based design in a distributed and collaborative environment. -Aided Design. (2004).
5. Deßloch, S., Harder, T., Mattos, N., Mitschang, B.: KRISYS: KBMS support for better CAD systems. Data and Knowledge Systems for Manufacturing and Engineering, 1989., Second International Conference on. pp. 172–182. IEEE (1989).
6. Kukkal, P., Imaizumi, M., Kobayashi, H.: CAD tool integration for ASIC design: an end-users perspective. System Theory, 1991. Proceedings., Twenty-Third Southeastern Symposium on. pp. 363–368. IEEE (1991).
7. Culler, D.E., Burd, W.: A framework for extending computer aided process planning to include business activities and computer aided design and manufacturing (CAD/CAM) data retrieval. Robotics and Computer-Integrated Manufacturing. 23, 339–350 (2007).
8. Boone, T., Ganeshan, R., Hicks, R.L.: Learning and Knowledge Depreciation in Professional Services. Management Science. 54, 1231-1236 (2008).
9. Xue, S., Kou, X.: Natural Voice-Enabled CAD: Modeling via Natural Discourse. Computer-Aided Design and Applications. 6, 125-136 (2009).
10. Lane, D., Seery, N.: Promoting Creative Discovery and Mental Synthesis through Freehand Sketching. Visualizing Change: Graphics on the. (2010).
11. Sharma, A., Madhvanath, S., Shekhawat, A.: MozArt: A Multimodal Interface for Conceptual 3D Modeling. System. 3-6 (2011).
12. Lee, J.C.: In search of a natural gesture. The ACM Magazine for Students. 16, 9–12 (2010).
13. Câmara, A.: Natural user interfaces. Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction-Volume Part I. pp. 1–1. Springer-Verlag (2011).
14. Jain, J., Lund, A., Wixon, D.: The future of natural user interfaces. Virtual Reality. 211-214 (2011).
15. Döllner, J., Hinrichs, K.: Object-oriented 3D Modelling, Animation and Interaction. The Journal of Visualization and Computer Animation. 8, 33-64 (1997).
16. Scali, S., Wright, M.: 3D Modelling is not for WIMPs. Human-Computer Interaction,. 2-6 (2003).
17. Anderson, D., Frankel, J., Marks, J.: Tangible interaction+ graphical interpretation: a new approach to 3D modeling. Proceedings of the 27th. (2000).
18. Schmidt, R., Wyvill, B., Sousa, M.C., Jorge, J.A.: Shapeshop: Sketch-based solid modeling with blobtrees. ACM SIGGRAPH 2006 Courses. p. 14. ACM (2006).
19. Giacchetta, G.: Genetic Selection of Parametric Scenes (Scene Generation) Honours Project Report. System. (2003).

20. Massie, T.: A tangible goal for 3d modeling. Computer Graphics and Applications, IEEE. 18, 62–65 (1998).
21. Kaltenbrunner, M.: TUIO, www.tuio.org.
22. Virbel, M., Hansen, T., Denter, C.: Kivy: Crossplatform Framework for NUI, www.kivy.org.
23. Kankaanpää, H.: Metaball math, http://www.niksula.hut.fi/~hkankaan/Homepages/metaballs.html.
24. Harrison, P.: Euclidean Distance Transform in Python, http://logarithmic.net/pfh/blog/01185880752.
25. Marques, T.: Development platform for elderly-oriented tabletop games, (2011).
26. Johnson, S.: E3 Report on Microsoft Kinect. Tech News Today (This Week in Tech netcast). (2010).

Motion Capture Fundamentals

A Critical and Comparative Analysis on Real-World Applications

Pedro Nogueira,
Faculdade de Engenharia da Universidade do Porto
Programa Doutoral em Engenharia Informática
Instituto de Telecomunicações
18 de Novembro de 2011
pro11014@fe.up.pt

Abstract. In this paper we provide the reader with a simple, yet thorough, overview of what is motion capture, its history and evolution so far and processes used in acquiring the tri-dimensional data of the recorded scene, take or animation. A brief look at the "mocap" process's (short for motion capture) structure is given followed by an extensive list of the technologies behind the most popular systems used today. We dive into the inner works of acoustical, mechanical, magnetic and optical mocap systems, also discussing the differences between active, passive and markerless optical systems, since these are the most popular of the above referred. Later on we also provide some insight into facial motion capture, right after we compare the data acquisition systems and quickly overview the generic data file structure. Finally we provide examples of real-world applications and some possible research fields in the area along with our conclusions.

Keywords: Motion capture, movement reconstruction, markerless capture.

1 Introduction

Animation gave its first steps in the early 20th century, when in 1911, cartoonist Winsor McCay drew a character in multiple sheets of paper with slight changes between these and then sampled them at a constant rate to create the illusion of motion [1].

Animation processes did not witness considerable innovation until computers started to take place in the process. With the birth of keyframing, which reduced the amount of samples needed to create an animation animators saw their work a lot more simplified. This process was time consuming because, at the time, every artist was forced to individually animate each pose/frame. With the introduction of keyframing the artist specified the initial and ending frames of the animation and the intermediate frames of the movement where automatically generated [2]. However some animations were still impossible to recreate due to their inherent complexity, for example the human walking animation, which is terrifyingly complex due to our articulations.

To speed up the animation process further, motion capture was invented, a means by which we capture the movements of objects in the real world and then insert the data of the captured movement in a tridimensional model of the world in a virtual environment. The process first evolved with mechanical systems that were quite cumbersome and limited the amount of freedom the actor could experience, limiting severely the animation spectrum that could be captured. This happened mainly because these were mechanical systems that resorted to very restrictive suits and large amounts of cable that hindered the actor's movements. Mocap has evolved much since then and today, as a response to these early issues, we have a broader range of options (Section 3). They include acoustical, mechanical, optical and magnetic systems, further divided in marker and markerless systems.

Today, motion capture is widely used in the gaming, movie and animation industry [3] as a means to provide quick, budget adapting body and/or facial animations in order to animate one or various characters. We provide insight into these methods and processes, and also the data processing and data formats that most systems use. Lastly we introduce some future work and research in motion capture. Research we believe would be highly beneficial and would enable future developments and breakthroughs in the area.

2 State of the Art

Motion capture has come a long way since its birth a few decades in the past. Still, there are a number of pending issues waiting a solution. In this section we provide some systems that represent today's state of the art in motion capture and that have dealt and overcome some of the limitations of mocap. We start by introducing, Ascension's Motion Star system [4]. This system tries to overcome some of the shortcomings of magnetic mocap systems, such as degree of freedom offered and electro-magnetic interference. Stanford University has also developed state of the art systems for markerless motion capture [5], that provides great accuracy, full body calculation of joint angles and joint centres, as well as a model matching followed by kinematic extraction. Carnegie Mellon University has also developed similar systems [6], which provide fully articulated 3D models by means of a joint skeleton and body shape acquisition. Finally Image Metrics has developed a truly astonishing markerless motion capture system, based in a spherical lighting rig equipped high resolution digital cameras and their proprietary markerless mocap technology that captures the model's animations to a degree where it is almost impossible to differ between the real and the virtual actor. We refer and council the reader to witness this work in [7] and [8], for this system is truly a noteworthy feat.

3 Motion Capture Systems

3.1 Marker-based Motion Capture

3.1.1 Acoustical Systems

In this type of system a set of sound transmitters are placed on the actor's main articulations, while three receptors are positioned in the capture site. The emitters are then sequentially activated, producing a characteristic set of frequencies, that the receptors pick up and use to calculate the emitters' positions in three-dimensional space. The computation of the position of each transmitter is as follows: Using as data the time interval between the emitting of the noise by the transmitter, the reception of this one by the receptor and the travelling speed of sound in the environment, one can calculate the distance travelled by the noise. To determine the 3D position of each transmitter, a triangulation of the distances between the emitter and each of the receptors is computed [9].

Some problems with these systems are; firstly, the difficulty in obtaining a correct description of the data in a certain instant. This is due to the sequential firing nature of the transmitters, which can create a non-fluid, or not so fluid as desirable, description of the movement. Another downside of these methods are the restrictions to the freedom of movement the cables induce on the actor, reducing the scope of movements available. Finally, the amount of transmitters that can be used is also limited, a factor that can hinder the quality of the animation [9]. Although these systems do not suffer of occluding or metallic object interference issues, typical in optical and magnetic systems, they are very susceptible to sound reflections or external noise, which being quite frequent and uncontrollable in most situations make them a "last resource" choice.

3.1.2 Mechanical Systems

These systems are made out of potentiometers and sliders that are put in the desired articulations and enable the display of their positions (Figure 1). Despite being underdeveloped, mechanical motion capture systems have some advantages that make them quite attractive. One advantage of these systems is that they possess an interface that is similar to stop-motion systems that are very popular and used in the film industry, thus permitting an easy transition between the two technologies. A final advantage is that they're not affected by magnetic fields or unwanted reflections, not needing a long recalibration process, which makes their use easy and productive.

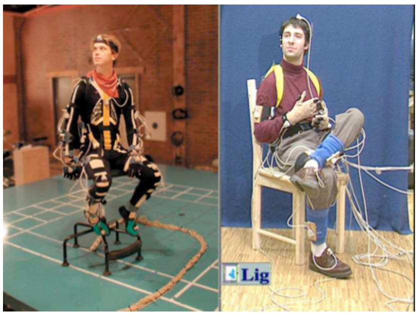


Figure 1: Left: Actor using full mechanical motion capture suit. Right: Actor using just some mechanical sensors instead of a full suit.

3.1.3 Magnetic Systems

Using a set of receptors that are placed in the actor's articulations it's possible to measure the positioning and orientation of the articulations relative to an antenna. Magnetic systems aren't very expensive, in comparison with other systems for motion capture. The workstation used for data acquisition and processing is cheap as well and the precision of data is quite high. With a typical sampling rate of ~100 frames per second, magnetic systems are perfect for simple movement capture.

The disadvantages of these systems include the huge number of cables that connect to the antenna, reducing the freedom degrees of the actor. Systems that do not require the use of cables are currently being developed [10], effectively eliminating this drawback. The possible interference in the magnetic field caused by various metallic objects and structures poses a restriction to the surrounding material, which can be of some gravity. Some systems are highly sensible even to the building's own structure presenting some interference, making this a critical flaw in magnetic mocap systems.

Nowadays we observe an effort by these product's manufacturers, which are investing in new and improved versions of their systems that exhibit a reduced vulnerability to these issues. Nonetheless, the issue has not been completely eradicated. Further reading can be done in [11].

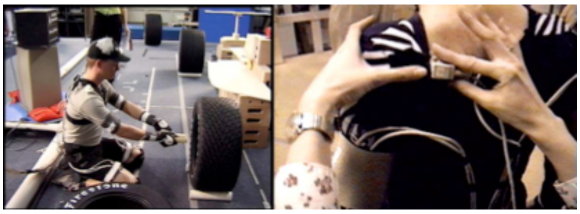


Figure 2: Left: Actor using magnetic mocap suit. Right: Close-up of a magnetic mocap sensor.

3.1.4 Optical Systems

With this kind of system the actor wears an especially designed suit, covered with reflectors that are placed in their main articulations. Then, high-resolution cameras are strategically positioned to track those reflectors during the actor's movement. Each camera generates the 2D coordinates for each reflector, obtained via a segmentation step. Proprietary software is then used to analyse the data captured by all of the cameras to compute the 3D coordinates of the reflectors. These systems are the most expensive ones in the market due to their cutting-end technological nature, such as the high-resolution cameras and sophisticated proprietary software. The values can hit the USD 250,000 bar.

The advantages of using these systems are mainly the very high sampling rate, which enables the capture of fast movements such as martial arts, acrobatics and gymnastics, among others. The sampling rate usually depends on the cameras used, meaning that the higher the resolution, the higher the sampling rate will be. Sampling rates of up to 200 frames/second are achievable.

Another advantage is the freedom offered by these systems, since, unlike the other systems, there are no cables or limited workspace and the reflectors pose no restrain or cumbersome effect on the actor. Also, since the reflectors offer no resistance there is virtually no limit to the number used in the capture process, which enables a very high theoretical level of detail.

The disadvantages of this method are the occlusion of some transmitters, especially in small objects such as hands or closely interacting objects, a problem that can compromise the entire process if the occluded data is unrecoverable. This problem can be somewhat overcome with the deployment of additional cameras and/or reflectors but this implies a higher processing time for the CPU during the tracking. Increasing the reflector numbers is no panacea, since the ”tracking-confusion” problem will eventually arise. This issue consists on the difficulty in identifying reflectors that are too close to one another and is directly influenced by the resolution of the cameras, being thus solved, by using cameras with higher resolution capabilities. This obviously leads to budget issues as these cameras are, as previously explained, extremely expensive. One final problem we can encounter is the lack of interactivity, since the data gathered must be processed (and sometimes undergo filtering and noise reduction) before it is usable. This gives us little feedback, an inconvenience when production costs are involved, the studio is booked for only a few hours and time is of the essence. Further reading on these systems can be done in [12].

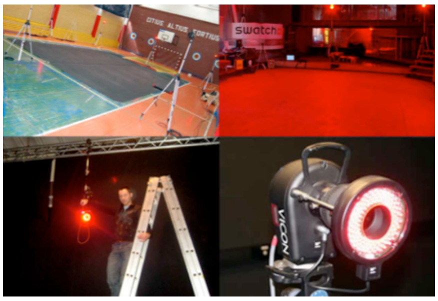


Figure 3: Upper left: A mocap studio. Upper right: Same mocap studio during capture. Lower left: A motion capture camera. Lower right: Close-up of a motion capture camera.

Active Markers

This kind of optical motion captures use LED’s that, instead of reflecting the light emitted by the high-resolution cameras, emit their own light, being powered by a small battery. Since according to the Inverse Square Law [13] we can achieve 1/4 the power at two times the distance, this can increase the usable capture volume.

Passive Markers

These markers, unlike the active markers, are coated with a retro-reflective material that reflects the light back to the cameras, that must first be calibrated so only the markers are identified, ignoring other materials. Using multiple cameras they are calibrated using an object with reflectors for which the positions of these are known. Waving a "wand" imbued with a series of reflectors, across the capture volume, is how this calibration process is usually performed. Normally a system will incorporate anywhere from 6 to 24 cameras, but some systems with over 300 cameras exist to reduce marker swap or confusion issues in complex captures.

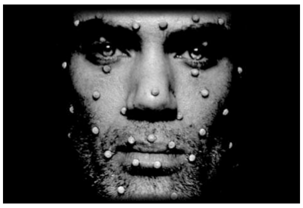


Figure 4: Passive markers placed on an actor's face.

3.2 Markerless Motion Capture

The ever-growing research in computer vision fields is quickly enabling the development of markerless optical motion capture techniques. These systems do not require special equipment for tracking of the actors' movement. The actors' movement is recorded in multiple video streams and computer vision algorithms analyse these streams to identify human forms, decomposing these into single, isolated parts used for tracking. The motion capture process is, thus, completely done via software, removing all the physical limitations while introducing computational constrains. A perfect example of such a system is Microsoft's Kinect, which has successfully introduced a solution for low-cost motion capture to the masses [14]. We refer the reader to [15], [16] and [17] for further reading on some of these systems.

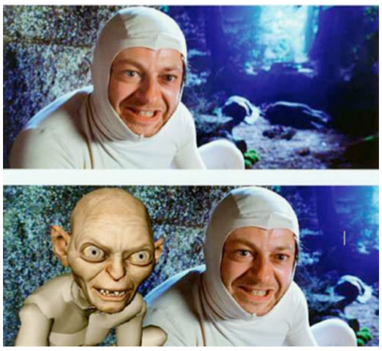


Figure 5: Markerless based motion capture used in the Lord of the Rings trilogy: Upper image: The actor whose movements are going to be captured. Lower image: The computer generated character side-to-side with the actor imitating his movements.

4 Acquired Data Processing

In what regards the data acquisition type, mocap systems are classified in two categories; direct and indirect acquisition. Direct acquisition systems are the ones that do not require any type of post-processing cycle, such as magnetic, acoustical and mechanical systems. Although this is, in fact, an upside to these systems it does not come without drawbacks. This is because these systems are more obtrusive and offer a lower sampling rate [18].

Indirect acquisition systems, in which optical systems are included, offer more freedom to the performer and a higher sampling rate. These systems group the captured data in data sets that are processed *a posteriori* by dedicated software; responsible for generating the final data set that represents the captured movement. This workflow scheme makes the use of these systems impossible in projects that require interactivity and near instantaneous viewing of the captures. They are especially suited for the accurate and seamlessly perfect capture of complex and fast movements [18].

Although not all of the methods described earlier require a data processing cycle, all of them, require the data to be cleaned, filtered and then mapped (rigged) to a skeleton, which then follows the positions of the markers in every frame of the animation by binding the markers to the skeleton's own articulations.

4.1 Data Formats

Data formats are normally proprietary formats, owned by the company that developed the software. These usually follow the same structure, only differing in some minor nuances like commentaries, special symbols, data and time information and additional data that the manufacturer deems useful.

The body parts are segmented into key regions and the information for each segment is stored separately. The segment structure consists of an object vector, with each vector representing one sample of the segment, stored in some file format (e.g. 3DS). Every segment object is represented by a 3D model with a bounding box and contain 9 information channels relevant to the segments X, Y and Z positions, as well as orientation and scale factors for each coordinate. This data structure makes it possible to transfer the animations between different models, given that they have the same scale and pivot positions in every segment [18]. There is research in this field to create algorithmic procedures that adapt the data to other structures without producing unrealistic or faulty animations. This would be ideally done by adjusting the pivot positions of the segments, the scale factors and the weights associated within each segment in the skeleton of the mode.

5 Applications

5.1 Films

5.1.1 King Kong

The mocap techniques used in the animation of the King Kong creature in Peter Jackson's film adaptation of the novel with the same name, involved optical marker based mocap with around 70 to 80 cameras in a huge warehouse, adapted for this function. Every animation of King Kong was played out by one actor (Andy Serkis) and then later heavily edited to compensate for the physiological differences between him and the skyscraper-climbing ape. In the capture about 60 optical markers on the body and 12 on the face were used. Out of the 70 to 80 cameras, 52 where used to track the body movement and around 20 for the facial expressions, the number however varied slightly from scene to scene [19].

The greatest challenge, however, came with the fact that the producers didn't want human-like movement, but ape-like movement. So, even though, the actor was acting like an ape, this just couldn't compensate for the fact that his body wasn't that of an ape, much less that of a five-story building one. Having taken that into consideration, a specific piece of software was used to translate the movement of the actor into a more convincing one, compensating for all the physical discrepancies. The software mapped every marker in the actor's body to a corresponding point in the gorilla's body and then adjusted the movement as it was being recorded. After this phase, the animation still went through minor adjustments done by some technicians to re-touch and polish some nuances [19]. Unfortunately the software used was proprietary, so no more details other than this basic workflow specification were released to the general public.

5.1.2 Lord of the Rings: The Return of the King

The visual effects of the Lord of the Rings trilogy were put in charge of WETA Digital [20], which assembled a team of computer artists, motion editors, and software engineers, among others. They also created a never-before-seen database to store every single shot frame in the trilogy. This digital library was able to track, cross-reference and analyse every item appearing in the film, making every single element susceptible to digital manipulation. This included characters, animals, props, and so on, meaning that the motion capture process had no need for extra assets, which reduced occlusion and movement restriction issues introduced by these items. Such an approach to these problems was a novelty in the area.

Despite this, the most astonishing achievement by WETA was the creation of fully digital creatures, featured in the game, such as Gollum (Figure 5), Treebeard, the Balrog and the Eye of Sauron. Gollum is one of the most surprising achievements in the mocap process of the Lord of the Rings trilogy as it was created by a mixture of motion capture technology and state-of-the-art computer animation. Also, in order to distance themselves from normal computer-generated characters Gollum was animated using joint movement based on organic muscle and bones, which can be seen in action in the film as the muscles and bone move under his translucent skin [21].

Citation: "WETA developed vast amounts of code to create Gollum", "They developed new modelling codes, new skin codes, new muscle codes. He is amazingly life-like and we were able to give him a range of expressions from the evil of Gollum to the sympathy of Smeagol." - Peter Jackson in LotR Making Of.

Optical motion capture techniques were used to record the actors and proprietary software to animate the creatures and battle scenes. These animations created via software were constantly subjected to synchronization with the mocap actor's movements (and also between the digital creatures), in order to create a believable battle or dialog scene [22].

5.2 Videogames

5.2.1 Tiger Woods PGA Tour 07

Tiger woods used a specifically developed facial motion capture technique, called Universal Capture, by Digital Air [24]. The process starts with a head scan, through the use of high-definition cameras, to obtain a 3D model of the actor's head. Then a large amount of markers (from 80 to 90) are placed in key regions of the actor's head and the movements of the markers are recorded as animations in a three-dimensional space by the cameras. The movement, opposed to FIFA's, which was captured in a spacious room, is recorded in a confined booth with limited movement. This is justified by the fact that in this process we only want facial animation and not full-body animation, where the players show virtually no facial animation. About 18 mocap cameras, 3 video cameras, 2 microphones and a huge amount of floodlights surround the booth. The three video cameras are used to capture the actor's face in three different angles, and the recorded video is then used to wrap around the modeled face from phase 1 of the capture process and thus create a final animated 3D model of the actor's head, which is animated via the recorded data by the mocap cameras and markers, in the booth [24]. A video demonstration and walkthrough of the system can be seen in [25].

5.2.2 Lord of the Rings: The Return of the King

The Return of the King used the same mocap systems (marker-based optical motion capture) that the production crew of the film trilogy choose. They used available animations, borrowed from the films and even the same actors that participated in the film were recruited. This was done in order to achieve maximum synchronization between the original animations and the game's animations, such as the combat or cutscenes' movements. We can therefore conclude that the mocap process from the films didn't need to be adapted because the new animations to be recorded were doable in the already used system. This was because, being an optical mocap system, gave the actors enough liberty to act out the desired animations.

6 Conclusions

Mocap systems, as shown throughout the paper, have evolved from simple, highly restricting, user un-friendly systems (software wise), to very mobile and specialized ones. The types of system discussed clearly all have their optimal case scenarios for deployment. However, optical mocap has evolved much more than its brethren systems. This is, for the most part, due to its ability to adapt very well to the major requirements of the film and videogame industry, which have invested and thus aided in this technology's development.

Despite its advantages, there are still improvements that can be done in this field (some of which proposed in the Future Research section). Other than these improvements, some mocap systems are still very limited in terms of the area where they can capture movements, being restricted to adapted warehouses or studios. These systems are also very high budget, which, in some cases, rules them out of question. They could benefit from lower budget versions, more accessible by the public and smaller companies.

Synthesizing, acoustical mocap systems are a last resort due to their cumbersome, external sound or sound reflection inflicted noise and non-fluid results. Mechanical systems are best suited for movements that don't require a high degree of freedom and there may be magnetic or reflection as these cause no noise in the capture process. Magnetic systems are useful in situations where the level of detail needed is high and the budget is low. However their great sensibility to the surrounding magnetic fields proves to be a critical flaw. We assumed the use a non-wired magnetic suit; otherwise, the captured movement's choices would be reduced to low level of freedom ones. Optical systems, although being expensive and having a processing/noise reduction stage, prove to be the best solution in most of the cases. This is because they offer a very high level of detail, freedom and working area enabling the sampling of virtually any movement.

In conclusion, mocap systems are very useful tools and provide a solution to a very important problem as well as enable truly amazing effects, as shown in the article. We have introduced the basic methods behind the various mocap systems and explored their advantages and disadvantages, concluding that every system has its best use in a specific scenario (and budget). The reader was also introduced to some of this technology's uses in real life and research topics, which we hope has provided a comprehensive overview of the area.

6.1 Future Research

After thoroughly studying the area and writing this paper we believe the following subjects would prove interesting fields of research.

Captured Movement Modification:

Since every data file represents a limited and closed data set of animations these cannot be manipulated after the caption process. An interesting field of research would be how to modify or derive new movements from an already captured movement.

Captured Movement Fusion and Concatenation:

As an extension to the prior research field one could try to create a fusion of any number of movements or concatenate them in order to create new movements from these original ones, while maintaining their fluidity and naturality, through algorithmic approaches.

Improvements in Actual Tracking Techniques:

As previously discussed, marker occlusion in optical mocap systems is critical. Improved tracking techniques could be developed to eliminate this problem.

Marker Mapping Techniques for Non-Human Beings:

Sometimes we want to capture movements that don't belong to human beings. Studying how to position the markers in the most effective way in these beings could prove beneficial, especially if the procedure could adapt itself to every object, thus becoming general and optimal and if it could help solve the problem stated in the above suggestion for future research.

Markerless Mocap in the Creation of High Definition Realistic Animations:

A highly advanced system such as the one in [3] and [4] constitutes an important milestone in highly realistic animation. We find it a very interesting and promising research field, which can help new breakthroughs in motion capture. Would it be in the development of even more realistic games, movies featuring indistinguishable virtual actors, personalized helpdesks, AIs with a human interface that creates credible and high definition expressions, amongst others.

Higher interactivity in mocap systems that involve a pre-processing stage:

With more interactive systems we would overcome the issues present in the systems that employ a data processing stage, enabling their use in projects that require nearly immediate access and validation of the gathered data.

Transference of motion captured animations between models:

Although mocap is somewhat transferable between models there are still a lot of limitations as to which animations can be transferred, given the source and destination model. For example, the animation of a normal person cannot be transferred to a giant, since by physical laws they move in a very different manner, or between models with very different body constitutions. An interesting case study would be how to adapt certain parameters of the animation so that it would be applicable to other models while maintaining its credibility.

Markerless Motion Capture for Psychological Analysis:

With the development of markerless mocap we could research methods to analyze an individuals inner feelings through expressions for psychological research and treatment. One application example of this technology would be an interactive tool that would respond, via a programmed AI, in accordance with an autistic child or mentally handicapped individual in order to further the healing process, since a human being while qualified cannot evaluate the person’s mental status as accurately or quickly as a well programmed computer.

References

1. J. Canemaker. "Winsor McCay - His Life and Art”. Abbeville. 1987.
2. Takeo Igarashi, Tomer Moscovich, John F. Hughes, "Spatial Keyframing for Performance-driven Animation", ACM SIGGRAPH / Eurographics Symposium on Computer Animation, 2005.
3. Technabob. "Image Metrics Emily CG Facial Animation Blows My Mind". Technabob Online. 2008. Available: http://technabob.com/blog/2008/08/20/image-metrics-emily-facial-animation-blows-my-mind/.
4. Ascencion Technology. "MotionStar (Tethered Model)". 2011.
5. S. Corazza, L. Mundermann, A. M. Chaudhari, T. Demattio, C. Cobelli, T. P. Andriacchi. "A Markerless Motion Capture System to Study Musculoskeletal Biomechanics: Visual Hull and Simulated Annealing Approach". Annals of Biomedical Engineering, Vol: 34, No: 6, pp: 1019-1029. 2006.
6. G. K. M. Cheung, S. Baker, T. Kanade. "Shape-From-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estimation and Motion Capture". Computer Vision and Pattern Recognition, Vol: 1, pp: 77-84. 2003.
7. E. O'Brien. "Image-Metrics Facil Animation". 2008.
8. Meet Emily - Image Metric Tech Demo. 2008. Available: http://www.youtube.com/watch?v=bLiX5d3rC6o.
9. O. Gabai, H. Primo. United States Pattent Application PCT/IL08/01578. 2008.

10. S. Yabukami, H. Kikuchi, M. Yamaguchi, K. I. Arai, K. Takahashi, A. Itagaki, N. Wako. "Motion capture system of magnetic markers using three-axial magnetic field sensor". IEEE Transactions on Magnetics, Vol: 36, Issue: 5, pp: 3646-3648. 2000.
11. L. Dickholtz. "Magnetion Motion Capture Systems". MetaMotion. 2009. Available: http://www.metamotion.com/motion-capture/magnetic-motion-capture-1.htm.
12. L. Dickholtz. "Optical Motion Capture Systems". MetaMotion. 2009. Available: http://www.metamotion.com/motion-capture/optical-motion-capture-1.htm.
13. E. Hill, J. Faller, H. Hill. "New Experimental Test of Coulomb's Law: A Laboratory Upper Limit on the Photon Rest Mass". Physical Review Letters, Vol: 26, pp: 721-724. 1971.
14. J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio. "Real-Time Human Pose Recognition in Parts from Single Depth Images". Computer Vision and Patter Recognition. 2011.
15. A. J. Davison, J. Deutscher, I. D. Reid. "Markerless motion capture of complex full-body movement for character animation". Proceedings of the Eurographic workshop on computer animation and simulation. 2001.
16. C. Chu, O. C. Jenkins, M. J. Matarie. "Markerless Kinematic Model and Motion Capture from Volume Sequences". Proceedings of IEEE Computer Vision and Pattern Recognition, Vol: 2, pp: 475-482. 2003.
17. B. Rosenhahn, T. Brox, H. Seidel. "Scaled Motion Dynamics for Markerless Motion Capture". IEEE Conference on Computer Vision and Pattern Recognition. 2007.
18. B. Bodenheimer, C. Rose, S. Rosenthal, J. Pella. "The Process of Motion Capture: Dealing with the Data". Eurographics CAS. 1997.
19. A. Serkis. King Kong - The Full Motion Capture Experience. 2011. Available: http://www.youtube.com/watch?v=XonxHXxlKnw.
20. WetaDigital - New Zealand. Available: http://www.wetafx.co.nz.
21. D. Salvator. "The Making of Gollum". ExtremeTech. 2004. Available: http://www.extremetech.com/gaming/55934-the-making-of-gollum.
22. P. Jackson. "Lord of the Rings: Return of the King Motion Capture". 2007. Available: http://www.youtube.com/watch?v=dnOeixfWLcs.
23. GameTrailers. "FIFA 09 - Motion Capture". 2008. Available: http://www.gametrailers.com/user-movie/fifa-09-motion- capture-video/254786.
24. DigitalAir. Universal Capture. 2011.
25. EA Sports. Tiger Woods PGA Tour 2007 Motion Capture. 2008. Available: http://www.gametrailers.com/video/motion-capture-tiger-woods/12922.
26. F. Besse, N. Heynen, B. Jamet, P. Lacour. "Cheap Motion Capture". 2008. Available: http://motion.capture.free.fr/index.htm.

An Enhanced Indoor Positioning Technique for Ubiquitous Computing Applications

Samih Eisa

Map-i Doctoral Program in Informatics Engineering,
Faculty of Engineering (feup), University of Porto,
Portugal

dma11043@fe.up.pt

Abstract. Indoor positioning is an exciting research area that promises many new applications in ubiquitous computing area. Knowing the exact position of a person in a building becomes a common requirement for many applications including person tracking, fire fighting, rescue, healthcare and patient monitoring. Several techniques and technologies have been developed for indoor positioning e.g. WiFi, RFID and Fingerprinting. Fingerprinting technique can be considered as a suitable candidate for many location-aware applications due to the low-cost and high-accuracy localization features. However, existing fingerprinting technique does not always give accurate localization especially in dynamic environments in which the measured signals are distorted by unexpected environment changes or other obstacles that might lead to position outliers. Within my PhD, we would like to identify which kind of obstacles cause position outliers in indoor environments and how these outliers can be handled in order to provide an effective indoor positioning method that causes trivial amount of localization errors. Moreover, we believe that ubiquitous computing environments bring new positioning requirements that are not supported by the existing positioning solutions and therefore, we would like to identify these new requirements.

Keywords: Indoor positioning, fingerprinting, signal strengths, location-based, WiFi positioning, localization accuracy, ubiquitous computing

1 Problem Statement

Getting accurate in-building position of a person is become a common requirement for many location-based applications. It helps in providing services for end-users anywhere and at anytime. Fingerprinting technique has emerged as a widely used indoor positioning technique based on WiFi technology [1]. It provides a better way for indoor positioning with low-cost and high-accuracy localization based on the measurement of the signal strengths collected from WiFi access points (APs) [2]. Two phases can be indentified in fingerprinting technique: the offline phase(training phase) in which the signal strengths in a building are collected from APs and stored in a database together with their corresponding coordinates or places' names. The other phase is the online phase (positioning phase) in which the located object samples the received signal strengths (RSS) from APs at its position and then search for similar location pattern in the offline database. The RSS matching process can be implemented using pattern recognition approaches (e.g. k-nearest-neighbor (KNN) or neural networks (NN)) or probabilistic approaches (e.g. Bayesian estimation) [3, 4]. The problem in this method is that the RSS could be affected by diffraction, reflection or any other obstacles that might lead to localization errors and position outliers. The main focus of my research lies in understanding the main causes of these position outliers and finding the best way to handle it. Moreover, the research will focus on identifying the positioning requirements of ubiquitous computing applications in order to identify the suitable positioning solution for these applications.

The reminder of this paper is organized as follows. Section 2 gives a brief state-of-the-art. Section 3 presents a preliminary research approach and Section 4 concludes the paper.

2 State-of-the-Art

Various techniques and technologies have been developed for indoor positioning. Authors in [4-9] provide comprehensive survey studies to identify and compare the major features of the existing indoor positioning solutions. They define the main positioning techniques that used for location estimation and based on these techniques they classify the existing positioning technologies and provide some properties and metrics for evaluation purposes e.g. accuracy, precision, cost and robustness. In this section, we will briefly review the main measuring principles and algorithms that used for location sensing.

2.1 Location Sensing

In general, there are three main principles used for signal location sensing: Triangulation, Scene analysis, and Proximity [4] . These principles are used to provide Physical, Symbolic, Absolute, or Relative location information for different applications. In the following we will briefly explain these three principles:

Triangulation.

Triangulation is a geometric-based location sensing technique. It estimates the position of an object by measuring its distance from multiple reference points using the Received Signal Strengths (RSS), Time of Arrival (TOA) or Time Different of Arrival (TDOA). Fig.1 shows the Triangulation positioning based on RSS. Alternatively, the position can be determined by computing the angles relative to multiple reference points as shown in fig.2.

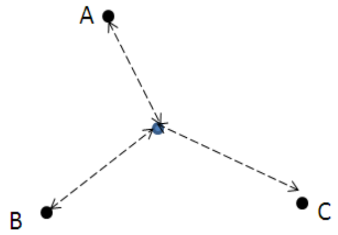


Fig. 1. Triangulation Positioning based on RSS (Received Signal Strength)[4]

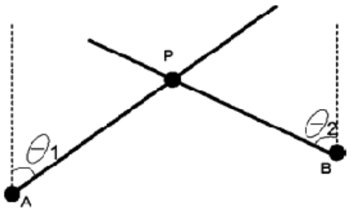


Fig. 2. Angulation Positioning based on AOA (Angle Of Arrival) [4]

Proximity.

This location sensing method provides symbolic relative location information by determining when an object is near to a well-know reference point. This method is relatively simple to implement because it entails that a target object is detect by a single reference point, for example, monitoring when a mobile device is in range of one or more access points.

Scene analysis.

Scene analysis technique refers to the type of algorithms that use the features of a scene for location estimation. In this method, an initialization or training phase (offline phase) is carried out firstly to collect features (fingerprints) of a scene and store them in a database. During the online phase, the location of an object can be estimated by matching the current measurement with the priori collected features and then the best fit is selected. Fingerprinting method uses this technique .The disadvantage of this method is that we need to have access to the features of the environment firstly before measuring locations. Moreover, changes to the environment might need reconstruction of the pre-collected features [4, 9].

3 Approach and Methodology

As a preliminary research methodology, in this thesis work we would like firstly to identify the main obstacles and constraints that might face the location determination of a person in indoor environments. We intend to start by surveying the existing indoor positioning systems with more focus on fingerprinting and WiFi-based technologies. Secondly, an experimental performance analysis of the existing fingerprinting algorithms is needed to clearly identify the drawbacks of these solutions. Thirdly, we believe that, the existing positioning solutions do not exactly address the localization needs of ubiquitous computing area and therefore, Identifying the possible constrains, issues and localization requirements that might face the deployment of indoor positioning in this area is needed. Finally, propose a robust and effective indoor positioning method that accurately handles position outliers and causes trivial amount of localization errors.

4 Conclusion

Indoor positioning is an emerging and a promising research area. In this paper, we opened a potential research direction in this area. We believe that, a more robust and effective indoor positioning method is needed to handle the drawbacks of the existing solutions while considering the new requirements of ubiquitous computing area. This paper is a preliminary outline step towards a more focused research topic in this area.

5 References

1. Altıntaş, B. and T. Serif, Location-aware Patient Record Access: Patient Identification Using Fingerprinting. 2011.

2. Kaemarungsi, K. and P. Krishnamurthy. Properties of indoor received signal strength for WLAN location fingerprinting. 2004. IEEE.

3. Meng, W., et al. Secure and robust Wi-Fi fingerprinting indoor localization. 2011. IEEE.

4. Liu, H., et al., Survey of wireless indoor positioning techniques and systems. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2007. **37**(6): p. 1067-1080.

5. Al Nuaimi, K. and H. Kamel. A survey of indoor positioning systems and algorithms. 2011. IEEE.

6. Cypriani, M., et al. Wi-Fi-based indoor positioning: Basic techniques, hybrid algorithms and open software platform. 2010. IEEE.

7. Gu, Y., A. Lo, and I. Niemegeers, A survey of indoor positioning systems for wireless personal networks. Communications Surveys & Tutorials, IEEE, 2009. **11**(1): p. 13-32.

8. Koyuncu, H. and S.H. Yang, A survey of indoor positioning and object locating systems. IJCSNS International Journal of Computer Science and Network Security, 2010. **10**(5): p. 121-128.

9. Mautz, R., Overview of current indoor positioning systems. Geodesy and Cartography, 2009. **35**(1): p. 18-22.

Session VII

Software Engineering

Chairman: Pedro M. Teixeira

Dataflow Programming Concept, Languages and Applications

Tiago Boldt Sousa[1,2]
tiago.boldt@fe.up.pt

[1] INESC TEC (formerly INESC Porto)
[2] Faculty of Engineering, University of Porto
Campus da FEUP Rua Dr. Roberto Frias, 378 4200 - 465 Porto, Portugal

Abstract. Dataflow Programming (DFP) has been a research topic of Software Engineering since the '70s. The paradigm models computer programs as a direct graph, promoting the application of dataflow diagram principles to computation, opposing the more linear and classical Von Neumann model. DFP is the core to most visual programming languages, which claim to be able to provide end-user programming: with it's visual interface, it allows non-technical users to extend or create applications without programming knowledges. Also, DFP is capable of achieving parallelization of computation without introducing development complexity, resulting in an increased performance of applications built with it when using multi-core computers. This survey describes how visual programming languages built on top of DFP can be used for end-user programming and how easy it is to achieve concurrency by applying the paradigm, without any development overhead. DFP's open problems are discussed and some guidelines for adopting the paradigm are provided.

Keywords: dataflow programming, visual programming, end-user programming, programming languages, parallel computing

1 Introduction

Dataflow programming (DFP) introduces a new programming paradigm that internally represents applications as a directed graph, similarly to a dataflow diagram. Applications are represented as a set of nodes (also called blocks) with input and/or output ports in them. These nodes can either be sources, sinks or processing blocks to the information flowing in the system. Nodes are connected by directed edges that define the flow of information between them. Most visual programming languages that use a block-based architecture for representing their workflow are indeed based on DFP [3]. Several advantages are inherited with such model, as presented in this paper.

[3] Although UML may seem an obvious candidate, it should not be regarded as a programming language, but rather as a specification language. Methods for making UML executable exist [20], although they are mainly ad-hoc solutions [10] and not part of the core standard, hence, not making UML a visual programming language.

1.1 Motivation

DFP is a commonly forgotten paradigm, despite its ability to successfully solve certain scenarios, from which the author highlights two.

A first advantage is the existence of visual programming languages[4], easing the work of programmers in a tool that, due to its simplified interface, can provide rapid prototyping and implementation of certain systems. Visual programming languages are also known to ease the process of providing end-user programming, where the user of an application is able to modify the behavior of the application in some way. Many languages exist providing such capabilities, as described in section 3. Visual programming has been successfully adopted both by experienced programmers and non-technical computer users (while still experienced), who are able to use those language as a tool to either extend an existing application or to build one from scratch.

A second point in favor of DFP is the implicit achievement of concurrency [16]. In the internal representation of an application, each node is an independent processing block, producing no side-effects, that is, working independently form any others. Such execution model allows nodes to execute as soon as data arrives to them, without the possibility of creating deadlocks, as there are no data dependencies in the whole system. This is a core feature of the dataflow model, removing the need to have programmers handle concurrency issues such as semaphores or manually spawning and managing threads. Such feature can greatly increase the performance of an application when executed on a multi-core CPU, a common architecture nowadays, without introducing any additional work for the programmer.

These two key points from DFP let the author believe that this paradigm should be part of the knowledge of any developer, empowering him to use it in scenarios were it best fits. This survey paper is expected to introduce readers with DFP, describing its historical background, introducing existing languages and open problems, guiding the reader in the right direction to adopt the paradigm.

1.2 Structure

This survey is composed by five sections, from which this first one is the introduction. The history and concepts of Dataflow Programming are described in the next section. Section 3 gives examples of DFP languages, frameworks for implementing the dataflow paradigm and know usages from it. Section 4 argues about some well-known issues over DFP, as well as describing some common answers for some of those questions. In section 5 the author argues on why DFP is relevant knowledge for any developer. Section 6 details future work and the paper is then finished with a last section detailing the conclusions gathered in this survey paper.

[4] Most visual programming languages are based on DFP [25]

2 Dataflow Programming Overview

Dataflow Programming is a programming paradigm whose execution model can be represented by a directed graph, representing the flow of data between nodes, similarly to a dataflow diagram. Considering this comparison, each node is an executable block that has data inputs, performs transformations over it and then forwards it to the next block. A dataflow application is then a composition of processing blocks, with one or more initial source blocks and one or more ending blocks, linked by a directed edge.

2.1 History

DFP has been subject of study in the area of Software Engineering for more than 40 years, with its origins being traced back at at the Ph.D. thesis of Bert Sutherland [30]. Sutherland used a light-pen and a TX-2 computer to create a visual programming language, on top of the SKETCHPAD framework. He also contributed with patterns for graphical representation of procedures that are still used in visual languages today.

In figure 1 Sutherland shows how arithmetic instructions can be represented in both textual and visual forms. In that example, extracted from Sutherland's thesis, we can understand how parallel operations occur and why they result in a reduction of the computation time in even such a small code snippet. We can observe that the calculation of the value of W can be processed simultaneously with the other arithmetic operations occurring in the two vertically aligned nodes, as there are no data dependencies between them. In a DFP language, such parallel computation is achieved automatically by the compiler. The compiler analyses the source and creates an internal dataflow representation of it, based on connected notes, commonly, with each node being processed by an individual thread. DFP compilers exist to create such binaries from either textual and visual languages.

2.2 Architecture

With the increased need to compute large datasets and enable common computers to process more than a single thread at the same time, both in the industrial and scientific world, the need for multi-core processor systems arose [9]. Despite that, multi-threaded programming was still an error prone task to achieve, as it was subject to race conditions, very complex scenarios to debug. The disadvantages and common problems with using threads were well summarized by Ousterhout [24]. Dataflow programming was able to provide parallelism without the increased complexity involved in the management of threads.

In dataflow programming, computation nodes are connected between themselves whenever a node as a dependency on the value processed from another node. Values are propagated as soon as they are processed to the dependent nodes, triggering the computation on them.

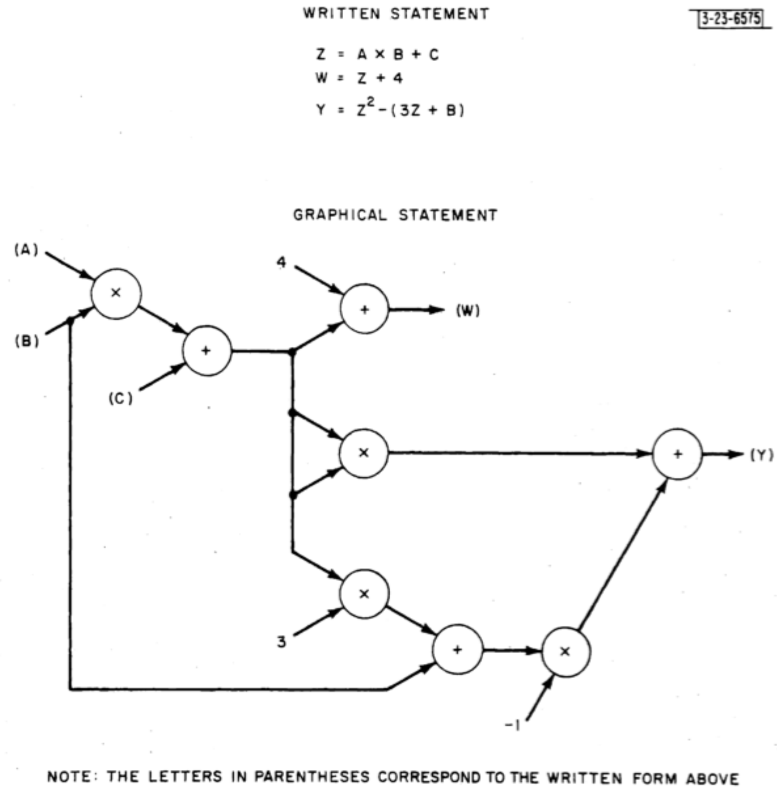


Fig. 1. A comparison of the textual and graphical representation of an arithmetic calculation, from Sutherland's Ph.D. thesis [30].

An initial approach to dataflow programming, by Dennis [6], started by suggesting the use of an architecture able to execute these applications at the hardware level, by giving static memory positions to each node to fill with values that could be read by the remaining nodes that were connected to it. With the introduction of multi-core CPUs and processing farms, languages evolved into supporting this more common architectures for portability reasons and provided developers with the necessary tools to parallelize their computations on common computers [2].

Introduced by Gilles Kahn, the *Kahn Process Networks* approached this problem by having sequential processes (nodes) to communicate via unbounded FIFO queues as message passing protocol [17]. Whenever the entry FIFO queue of a node was not empty, the first value would be processed by the node and outputted into the FIFO belonging to the next node in the chain.

DFP has evolved into a resourceful method to exploit modern computer architectures, composed by multi-core CPUs, as well as computation farms, while reducing the development complexity.

3 Languages and Usages

The dataflow paradigm has been used in a wide range of contexts, supporting either massive computation of data or being the basis for visual languages providing end-user programming capabilities. The *Journal of Visual Languages and Computing* [5] is a reference point in the novel researches being held in this topic.

This section introduces DFP languages and relevant implementations using them. The section describes a textual and a visual dataflow language, particularly, SISAL and Quartz Composer. Although, many more exist, with some relevant names such as LabVIEW [31], VHDL [29] or LUSTRE [12].

3.1 Visual and Textual Dataflow Languages

Independently of the representation style adopted by a the language, it is up to its compiler to analyze the provided source and generate an internal dataflow representation that will define how information will flow between nodes. Several architectures for generating the internal model were researched by Johnston et al [16].

Despite this common comparison to dataflow diagrams, as previously stated, DFP is not a synonym of visual programming, although most visual programming languages are based on the dataflow paradigm. In fact, many early dataflow languages had no graphical representation.

The applications achievable with textual and visual languages do not differ, although, choosing the best language for each situation is a key factor to achieve success. Visual programming languages favor the simplicity of a visual representation. Visual programming can also be used to provide an end-user programming interface. Textual languages require more knowledge but are usually faster to work with, as well as provide a more scalable organization of the source code [7].

SISAL SISAL, acronym for Streams and Iteration in a Single Assignment Language, is a derivative of the Val language and it is a text-based functional and dataflow programming language from the late 80's, introduced by Feo and Cann [19,8,9]. The language is strongly-typed, with a Pascal-like syntax for minimizing the learning curve and enhancing readability.

The language intended to compete in performance with Fortran while using the dataflow model to introduce parallel computation in the first multi-core

[5] Available online at *http://www.journals.elsevier.com/journal-of-visual-languages-and-computing*.

machines. It still provided a micro-tasking environment that supported the dataflow architecture on traditional single-core machines.

In order to increase its performance, SISAL's compiler was able distribute computation between nodes in an optimized way. The management of the internal dataflow was fully automatic — the compiler was responsible to create both the nodes and connections between them. In runtime, each node was executed by an independent thread that was always either running or waiting for data to arrive to the node. Data was processed upon arrival and the result forward along the dataflow chain.

In some benchmarks, SISAL was able to outperform Fortran in computation performance [5].

Quartz Composer Part of XCode, the development environment suite from Apple, Quartz Composer is a node-based visual programming language. The language was developed for quick development of applications for processing and rendering graphical data by non-technical users, as it doesn't require programming knowledges [15].

Quartz Composer stands out from other dataflow languages due to its superior graphical editor, as seen in figure 2. The editor provides an intuitive way for users to add, configure and connect nodes in their dataflow. Each node can be either a source, sink or transformation of data and the editor manages type casting automatically.

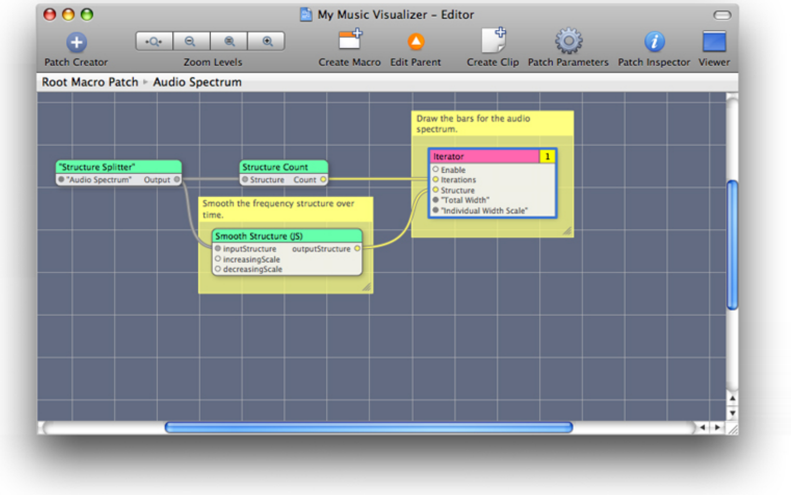
The language has a very extensive library of components that interacts with the operative system out of the box. Transformation blocks can be connected between any two blocks of information to provide computation over the flowing data.

The editor allow users to create modules without having to write a single line of code and allows these modules to be integrated with applications developed in Cocoa with the XCode suite. It always allows the creation of animations that can either be used as screen savers or played with Quicktime.

3.2 End-User Programming

DFP is behind most Visual Programming languages based on dataflow diagrams. Such languages not only target experienced developers but also non-technical users, providing them with a simplified interface for building applications. In fact, end-user programming is a common usage for dataflow applications, both by using visual dataflow-based editors, such as Apple's Quartz Composer (previously referred) or with spreadsheets, also a form of end-user programming, empowered by the DFP paradigm.

Graph-based Empowered by intuitive interfaces, such as the one provided by Quartz Composer, users are able to extend or create applications without the need to know how to program. This approach usually relies on the use of a set of pre-defined blocks that can be used to compose the diagram, connected by directed edges.



Draw the bars for the audio spectrum.

Smooth the frequency structure over time.

Fig. 2. The Quartz composer editor. Blocks and the connection between them are clearly visible. The interface is visually attractive and easy to use.

Spreadsheets Spreadsheets are probably the most common example of DFP and widely adopted by every type of computer users.

On a spreadsheet, each cell represents a node that can either be an expression or a single value. Dependencies can exist to other cells. Following the dataflow model, whenever a cell gets updated, it sends its new value to those who depend on it, that update themselves before also propagating their new values. This specific type of application is commonly denominated as Cell-Oriented DPF or Reactive programming.

At a more advanced level, tools exist that can extract a visual dataflow model from spreadsheets [13]. These are useful for many scenarios, such as debugging complex expressions or simplify the process of migrating a spreadsheet to a new software.

3.3 The Actor Model

The actor model is a very popular concurrency model by Carl Hewitt from MIT introduced in the '70's. With his team, he researched a method that allow developers not only to simplify the process of parallelizing their computations, but also to increase the confidence on the concurrent behavior of their programs [14].Twitter as adopted it for scaling their computations [21].

An *Actor* is an agent that receives and sends messages, behaving independently from other actors in the system. On each message, the actor is able to start new actors, compute data or reply with messages to other existing actors. In the dataflow paradigm, an actor is the equivalent to the node and the messages past are equivalent to the connections between nodes.

This architecture perfectly fits the dataflow model when an actor is used as a processing node and the massages between them as communication channels. In cases where there's the need to use an imperative or functional programming language, the actor model could be applied to port the concepts of dataflow programming into those languages, as it has been done by [27,18,11,23].

Many implementations of the actor model are freely available for several languages [26,28,32,1].

4 Open Problems

Dataflow programming is an area still open to further research, with some open issues to answer. In fact, most of the open questions today have long been identified and despite the improvements, patterns for answering them are yet to be achieved.

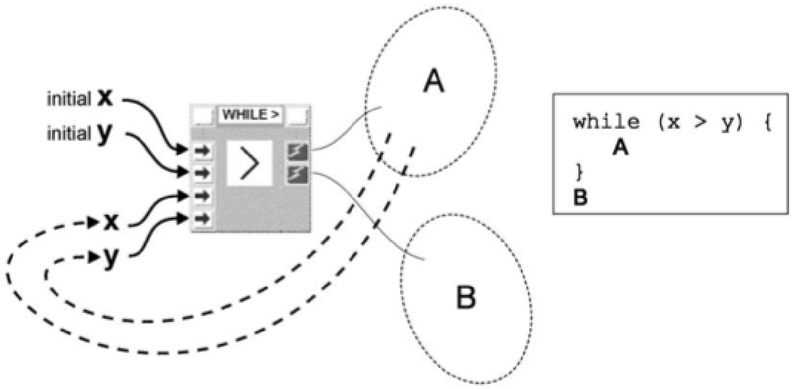
4.1 Visual Representations

Despite DFP being achievable without a visual programming environment, a graphical representation of how nodes connect in a dataflow-based application provides the user with a better understanding of what the application is supposed to do, providing the possibility of end-user programming. Although, representing conditions and iterations, as well as more complex algorithms or applications might result in a graph with an huge number of nodes with tangled connections, hard to read and maintain. Bellow, some solutions for this problem are proposed.

Iteration and Conditions To represent conditions or iterations as a set of nodes can easily result in a complex graph, nontrivial to understand, if the proper abstractions are not adopted.

Mosconi [22], summarized techniques adopted by the languages Show-and-Tell and Labview, while also introducing his approach to iteration and conditions using the VIPERS language, another dataflow visual programming environment based on the Tcl language [3]. In his paper, Mosconi describes viable implementations of the loop expressions *For* and *While* and explains how index-based iterations can be represented, as well as how to handle ending conditions using blocks with that sole purpose. The representation of a *While* block in VIPERS is shown in figure 3. Similarly, he suggests the creation of a single block for each type of loop and condition, native in the language, in order to significantly reduce the size of the graph, removing the large number of elements that would be needed to construct such expressions.

Visual Granularity Another open problem with visual DFP languages also happens with complex applications, when composed by a very large number of



```
while (x > y) {
   A
}
B
```

Fig. 3. A While block in VIPERS. *A* represents a block (or set of blocks) inside the loop that receives and generates new values of x and y. Whenever *A* returns an $x \leq y$ the loop exits and continues execution to block *B*.

nodes. In some cases, for experienced programers, the complexity of interpreting a visual representation can end up being higher than reading textual source code.

An approach to solve this problem it to allow a variation on the granularity of data shown at a given moment. To do so, nodes can be grouped hierarchically, so that they can be reduced into a single block that represents them, only showing the inputs and outputs of the whole group of that node. The amount of data shown for a node at a given time can also be configured. At any time the node can be expanded, enabling the user to alter its containing nodes.

4.2 Debugging

Debugging parallel applications requires tools capable of monitoring everything happening in each concurrent operation. In visual programming languages that process becomes even more complex, as the programmer has no direct control over the parallelism. There is the need to map the execution in the direct graph in order to provide visual feedback to the programmer. Browne et al [4] described an approach to debug these languages as a set of five steps:

1. Identify and select the portions of the graph whose behavior will be monitored;
2. Specify the expected execution behavior for each of the nodes in the specified set to be monitored;
3. Run the application with a test scenario as input and capture the execution behavior of the selected portions of the program;
4. Determine where the actual execution and expected events first diverge;
5. Map the elaborated graph of expectations back to the original graph, signaling where errors were detected.

The steps above can be followed by language designers to guide the development of visual debugging tools for DFP languages using a graph-based representation of the application, obtained from either a textual of visual language.

5 Discussion

This paper introduces the DPF paradigm and presents the two most relevant features within it: DFP as a basis for most visual programming languages, including a as way of providing end-user programming in applications and the ability to seamlessly provide developers with a parallel computational model, without introducing development complexity.

Visual Programming Languages allow experienced users to perform rapid application development and non-technical users to extend their application, what is commonly denominated by end-used programming, or create their own applications, without requiring programming knowledges. A common issue with these languages is the complexity to provide abstractions capable of representing an application without resulting in a huge, unperceivable, dataflow diagram — this paper identifies two patterns that can be applied to prevent this situation. Non visual DFP languages also exist. The textual approaches to DFP have a compiler capable of inferring the internal dataflow representation of the application, defining how parallelism is achieved automatically.

Concurrency is also easily achieved by the lack of side-effects in a DFP processing node. Following the concept that data is transmitted as a message and that these are sequentially processed as they arrive to a node provides DFP languages with parallelism out of the box, a valuable feature for developers looking to increase performance on parallelizable applications and algorithms.

6 Future Work

Despite the advantages in performance provided by DFP and the possibility of providing end-user programming with visual languages, there are no frameworks that provide integration of these features in modern day languages. Future work will consist on the development of such framework, believed to be of interest either for academic and industrial purposes, by using the actor model for implementing the dataflow paradigm, independently form the language chosen for implementation.

7 Conclusions

To conclude, the author believes that dataflow programming is a viable paradigm to be explored today for creating either end-user programming and parallel computation applications. Due to the lack of good quality visual editors and frameworks available for creating such systems, the creation of a generic framework for building end-user programing systems on top of a DFP architecture based on the actor model would be of use in several scenarios and will be pursued as future work.

References

1. Agha, G.: Actors: a Model of Concurrent Computation in Distributed Systems, Series in Artificial Intelligence (Jun 1985)
2. Arvind, D.: IEEE Xplore - Dataflow architectures and multithreading. Annual review of computer science (1986)
3. Bernini, M.: VIPERS (1994)
4. Browne, J., Hyder, S., Dongarra, J.: IEEE Xplore - Visual programming and debugging for parallel computing (1995)
5. Cann, D.: Retire Fortran? A debate rekindled (1991)
6. Dennis, J.B.: Data Flow Supercomputers. Computer 13(11), 48–56 (1980)
7. Erwig, M., Meyer, B.: Heterogeneous visual languages-integrating visual and textual programming pp. 318–325
8. Feo, J., DeBoni, T.: A tutorial introduction to sisal (August 1991), `https://waimingmok.wordpress.com/2009/06/27/how-twitter-is-scaling/`
9. Feo, J., Cann, D.: A report on the Sisal language project (1990)
10. Ferreira, H., Aguiar, A., Faria, J.: Adaptive Object-Modelling: Patterns, Tools and Applications. In: Software Engineering Advances, 2009. ICSEA '09. Fourth International Conference on. pp. 530–535 (2009)
11. Gu, R., Janneck, J., Bhattacharyya, S., Raulet, M., Wipliez, M., Plishker, W.: Exploring the Concurrency of an MPEG RVC Decoder Based on Dataflow Program Analysis. Circuits and Systems for Video Technology, IEEE Transactions on 19(11), 1646–1657 (2009)
12. Halbwachs, N., Caspi, P., Raymond, P., Pilaud, D.: The synchronous data flow programming language LUSTRE. Proceedings of the IEEE 79(9), 1305–1320 (Sep 1991)
13. Hermans, F., Pinzger, M., van Deursen, A.: Breviz: Visualizing Spreadsheets using Dataflow Diagrams. arXiv.org cs.SE (Nov 2011), 9 Pages, 5 Colour Figures; Proc. European Spreadsheet Risks Int. Grp. (EuSpRIG) 2011 ISBN 978-0-9566256-9-4
14. Hewitt, C., Bishop, P.: A universal modular ACTOR formalism for artificial intelligence. 3rd IJCAI-73 (1973)
15. Inc., A.: Quartz composer user guide (July 2007), `http://developer.apple.com/library/mac/#documentation/graphicsimaging/conceptual/QuartzComposerUserGuide/qc_intro/qc_intro.html#//apple_ref/doc/uid/TP40005381`
16. Johnston, W., Hanna, J.: Advances in dataflow programming languages. ACM Computing Surveys (CSUR) (2004)
17. Kahn, G.: The Semantics of a Simple Language for Parallel Programming. In Information Processing 7́4: Proceedings of the IFIP Congress (1974), pp. 471-475. pp. 471–475 (1974)
18. Lee, E., Parks, T.: Dataflow process networks. In: Proceedings of the IEEE. pp. 773–801 (1995)
19. McGraw, J.: The VAL Language: Description and Analysis (1982)

20. Mellor, S.J., Balcer, M.B.J.I.: Executable UML: A Foundation for Model-Driven Architectures. Addison-Wesley Longman Publishing Co., Inc. (Jun 2002)
21. Mok, W.: How twitter is scaling (June 2009), `https://waimingmok.wordpress.com/2009/06/27/how-twitter-is-scaling/`
22. Mosconi, M.: ScienceDirect - Computer Languages : Iteration constructs in data-flow visual programming languages. Computer languages (2000)
23. Oh, H.: Constant Rate Dataflow Model with Intermediate Ports for Efficient Code Synthesis with Top-Down Design and Dynamic Behavior. Quality Electronic Design, 2008. ISQED 2008. 9th International Symposium on pp. 190–193 (2008)
24. Ousterhout, J.: Why threads are a bad idea (for most purposes) (1996)
25. Petre, M.: ScienceDirect - International Journal of Human-Computer Studies : Mental imagery in program design and visual programming. International Journal of Human-Computer Studies (1999)
26. Philipp Haller, F.S.: Actors in Scala pp. 1–139 (Mar 2011)
27. Plishker, W., Sane, N., Bhattacharyya, S.: A generalized scheduling approach for dynamic dataflow applications. In: Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE '09. pp. 111–116 (2009)
28. Scherer, A., Gandhi, R.: Programming Concurrency on the JVM
29. Sjoholm, S., Lindh, L.: VHDL for Designers. Prentice Hall PTR, Upper Saddle River, NJ, USA (1997)
30. Sutherland, W.: On-Line Graphical Specification of Computer Procedures. (1966)
31. Travis, J., Kring, J.: LabVIEW for Everyone: Graphical Programming Made Easy and Fun (3rd Edition) (National Instruments Virtual Instrumentation Series). Prentice Hall PTR, Upper Saddle River, NJ, USA (2006)
32. Vajda, A.: Programming Many-Core Chips - András Vajda, Mats Brorsson, Diarmuid (CON) Corcoran - Google Books (2011)

Using XML Schemas in Parallel Corpora

Sara Fernandes[1]

HASLab / INESC TEC , University of Minho
sarasantos.fernandes@gmail.com

Abstract. Parallel corpora are resources used in Computational Linguistics and Natural Language Processing and are defined as a set of texts, in different languages. These sets are translations of each other. When dealing the sharing resources process, there have been firm intention on the use of XML formats. This is no different when talking about parallel corpora sharing. When visiting different projects in the web that release parallel corpora for download, we can find several different formats. In fact, this abundance of formats has led some projects to adopt more than one format.
This article discusses three of the main formats: XML Localisation Interchange File Format, Translation Memory Exchange format and the Text Encoding Initiative. We provide a comparison of their formal definition and their XML schema. And finally, we present TEI as the best option between these three formats for parallel corpora.

Keywords: corpora, parallel corpora, xml, xliff, tmx, tei

1 Introduction

Natural Language Processing or Computational Linguistics are examples of areas where corpora, where it is included parallel corpora, presents themselves as relevant resources. To best understand some of the concepts we will discuss, we should start by defining this concept.

The *corpus* (plural, *corpora*) term, born in Linguistics, refers to a finite collection of texts, usually from a restricted domain [5]. There are many examples of available corpora. The most well known is the British National Corpus[1].

A *Parallel Corpus* is a collection of texts in different languages, where each of them has a translation. In some cases one of these languages is considered as the *source language*, and its translations as the *target languages*. Although is not consensual, it is usual to consider that a parallel corpus is aligned at the sentence level, meaning that there is a relationship between sentences in different languages.

This alignment process can be defined as: having two parallel texts, U and V, a sentence alignment of these texts is a segmentation of U and V in n segments,

[1] http://www.natcorp.ox.ac.uk/

such that, for each i, $1 \leq i \leq n$, u_i and v_i are mutual translations, and u_i and v_i are, respectively, sequences of sentences from U and V [4].

Note that this definition means that we might have segmentations u_i or v_i that are empty sequences from U and V. Therefore, there might exist sentences in one of the languages that does not have a corresponding translation. Indeed, the creation or removal of sentences during the translation process is common.

This definition can be expanded to a set of languages, instead of just a pair. In this situation, we have a set S of m texts T_i $(1 \leq i \leq m)$, that have n segments each, such that, $\forall i, j \quad 1 \leq i \leq m \wedge 1 \leq j \leq n$, $t_{i,j}$ sequences of sentences are mutual translations.

The Parallel Corpora definition is simply the mapping between segments in different languages. Researchers, that are in the Natural Language Processing or Linguistics field, like to enrich their parallel corpora with some extra information. Note that this information to be added will depend on the corpus objective. Examples encompass the simple annotation of named entities (company name, for instance), morphologic or part-of-speech tagging of each word, syntactic structure, e.g..

This diversity of possible annotations makes it almost impossible to define a standard schema with all the alternatives one might want. Therefore, the adopted solution is the ability to define generic tags that can be personalized by each user.

In this article we will focus on three different formats that have been used by the research community to encode corpora and more specifically, parallel corpora:

- The Text Encoding Initiative (TEI) schema (subsection 2.1);
- The Translation Memory Exchange (TMX) schema (subsection 2.2);
- The XML Localisation Interchange File Format (XLIFF) (subsection 2.3);

We will present the formats origins and the original purpose for which they were created. Their goals are different, and this means that the level and type of annotation they can support is diverse. Nevertheless, they can all encode non-annotated parallel corpora, meaning it should be possible to define computational bridges to convert between these formats.

Section 2 presents each of these formats in particular; Section 3 compares their structure in means of usability and flexibility. And finally, section 4 discusses the directions users who need to encode parallel corpora should follow.

2 Parallel Corpora Encoding Standards

The formats described in this paper are currently used by researchers to release parallel corpora. Very often the researchers make their corpora available in more than one format[2].

[2] One example can be found in the Per-Fide `http://per-fide.di.uminho.pt/site.pl/resources.pt`

In this section we will define the subsets that are relevant to encode parallel corpora and annotate possible language phenomena. In the end, we will perform a qualitative evaluation on their flexibility to encode parallel corpora (see section 3).

2.1 TEI: Text Encoding Initiative

The Text Encoding Initiative (TEI) collection of schemas [8] was created to help in preparation and interchange of electronic texts for most real-world situations. TEI is modular, and depending on the text being encoded the set of schemas to be used is different. TEI includes a big variety of schemas, to encode texts, verses, transcription of speech, standard dictionaries, lists of places and names (toponyms and onomastic indexes), tables, mathematical formulae, graphs, networks, trees and others.

In particular, TEI includes schemas to encode language corpora (chapter 15 of the TEI Guidelines for Electronic Text Encoding and Interchange) [3]and for text segmentation and alignment (chapter 16)[4].

All these schemas share a common schema, known as the TEI header. This header includes typical meta-information, as the name of the document, its authors, the document copyright, editor, publisher, year, e.g. While meta-information is relevant when encoding corpora and parallel corpora, in this article are more interested in the means these schema have to encode the corpora, itself.

Nevertheless, we should stress the relevance of meta-information for corpora construction. It is very relevant to know the genre of the text (journalistic, literary, religious, e.g.), the age of the text (when it was written), its language and sub-languages, its type (oral, written), e.g. All this information can be stored in the TEI header. A good example of the use of the TEI header can be found in [5].

The macro-structure of a TEI corpus can be described as follows:

$$\begin{aligned} teiCorpus &\leftarrow teiHeader,\ (TEI \mid teiCorpus)^{+} \\ TEI &\leftarrow teiHeader,\ text \\ text &\leftarrow front\ ?,\ (body \mid group),\ back\ ? \\ group &\leftarrow (text \mid group)^{+} \end{aligned}$$

With this structure it is possible to have a header for the full corpora, and a separated header for each text. Also, each text might be grouped in different sections.

The *text* element is used by TEI to store all kind of texts. Therefore one can expect all kinds of mark-up to be possible inside this element. Although there are some corpus that might come from well structured data sources, most are processed by automatic tools, that just extract plain text. Therefore we can

[3] http://www.tei-c.org/release/doc/tei-p5-doc/en/html/CC.html

[4] http://www.tei-c.org/release/doc/tei-p5-doc/en/html/SA.html

[5] http://per-fide.di.uminho.pt/

consider that a text is just a sequence of paragraphs (*p* element) or lines (*l* element, often used for verse lines).

Some texts include some other level of segmentation, like the *div* element, that is used to divide text into sections.

For text annotation, TEI provides elements below the line or paragraph level. It includes elements for sentences (*s* element), for clauses (*cl* element), phrases (*phr* element) and words (*w* element). In fact it provides elements below word level, as morpheme, character or punctuation character.

Given the amount of elements to annotate different levels of text, the annotation of a corpus in TEI format can be very detailed. Any one of these elements can have attributes like *type* and *function* for phrases and clauses, *lemma* and *type* for words. Therefore, it is very simple to add all the needed information with these attributes, that have an open content type.

The alignment is implemented as links between elements. Often, parallel corpora are encoded in TEI as three distinctive files: the text in the source language, the text in the target language, and the alignment file. This alignment file includes the TEI header, and a sequence of *linkGrp* elements. These elements have some meta-information, like the documents that are being linked (in the *xtargets* attribute), and includes a list of *link* elements. These elements can include a *type* attribute (that is usually the number of segments form the source-text and from the target-text that are being linked), and a *xtargets* or *targets* attribute that has the identifiers used in the individual text files for the *p* or *l* elements (although this mechanism makes it easy to link sub-paragraph parts, like sentences, clauses, phrases or even words).

As an example for a *linkGrp* element:

```
<linkGrp targType="head p" xtargets="jrc-pt;jrc-ro">
  <link type="1-1" xtargets="28;28"/>
  <link type="1-1" xtargets="30;30"/>
  <link type="1-1" xtargets="31;31"/>
  <link type="1-2" xtargets="32;32 33"/>
  <link type="1-2" xtargets="33;34 35"/>
</linkGrp>
```

Fig. 1. Example of a *linkGrp* element:

This procedure is not bilingual since more than two languages support is easy to perform. To achieve this goal the only thing that needs to be done is extending this mechanism. In fact, we can find two different solutions: first, instead of two text files, we have one per language, and instead of a *linkGrp*, we have a set of groups, one for each language pair; other solution is to have more than two fields in the *targets* or *xtargets* attributes.

This sort of description on the TEI mechanisms for encoding corpora and their alignment is not detailed since we do not intend to write a tutorial, but to compare the referred formats. For more information, we invite the readers to consult the Guidelines for Electronic Text Encoding and Interchange that are available on the web[6].

As an example of project/corpus encoded in TEI, please check the multilingual parallel corpus based on the *Acquis Communautaire*[7], the total body of European Union (EU) law applicable to the EU Member States [6].

2.2 TMX: Translation Memory Exchange

The Translation Memory Exchange format was designed for the interchange of translation memories across different vendors of computer assisted translation (CAT) software. It is a standard, or norm, defined by the Localisation Industry Standard Association[8] (LISA). LISA is an association where some Universities and the major companies with CAT software or localisation offices are represented. Examples of partners are Abbyy, Adobe Systems, Autodesk, Cisco Systems, Dell, Hewlett-Packard, ICANN, Intel Corporation, Lucent Technologies, OASIS, SDL International, Skype, Trend Micro, VMWare and XEROX.

To understand the idea of translation memory it is helpful to explain how a CAT software works. When performing a translation task, the translator is faced with sentences already translated by herself or by someone on her group. Therefore, a CAT tool stores in a database all performed translations. These translations are stored sentence by sentence (or sequence of words by sequence of words, since the reuse of translations is more effective with short sequences of words).

Therefore, a translation memory can, in a simplified way, be seen as a set of pairs that relate sequences of words in two different languages. This informal definition is near to the definition of parallel corpora. Note that for parallel corpora we are forcing an order, something that translation memories do not guarantee by themselves. Given that translation memories are stored in XML files an implicit order (the order of appearance) exists. This makes the TMX format relevant for storing parallel corpora.

Again, note that we are simplifying the structure of TMX removing non relevant elements. The macro-structure of a TMX file is defined as follows[9]:

$$\begin{aligned} tmx &\leftarrow header,\ body \\ header &\leftarrow @creationtool,\ @segtype,\ @srclang,\ @adminlang, \\ &\quad (note \mid prop)^\star \\ note &\leftarrow \#PCDATA \end{aligned}$$

[6] `http://www.tei-c.org/release/doc/tei-p5-doc/en/html/index.html`

[7] `http://wt.jrc.it/lt/Acquis/`

[8] `http://www.lisa.org/`

[9] Attributes are denoted with the @ symbol. Also, the *seg* element definition is simplified.

$$\begin{aligned}
prop &\leftarrow @type, \#PCDATA \\
body &\leftarrow tu^{\star} \\
tu &\leftarrow @srclang,\ @segtype,\ ((note \mid prop)^{\star}, tuv^{+}) \\
tuv &\leftarrow @xml:lang,\ ((note \mid prop)^{\star}, seg) \\
seg &\leftarrow \#PCDATA
\end{aligned}$$

A TMX file is a header with some meta-information and a body with a sequence of translation units (*tu*). A translation unit is a sequence of translation unit variants (*tuv*) with a segment (*seg*). Figure 2 presents a simple TMX file[10].

```
<?xml version="1.0"?>
<tmx version="1.4">
  <header creationtool="XYZTool" creationtoolversion="1.01-023"
          datatype="PlainText"   segtype="sentence"
          adminlang="en-us"      srclang="EN" />
  <body>
    <tu>
      <tuv xml:lang="en"><seg>hello</seg></tuv>
      <tuv xml:lang="it"><seg>ciao</seg></tuv>
      <tuv xml:lang="pt"><seg>olá</seg></tuv>
    </tu>
    <tu>
      <tuv xml:lang="en"><seg>world</seg></tuv>
      <tuv xml:lang="en"><seg>earth</seg></tuv>
      <tuv xml:lang="it"><seg>mondo</seg></tuv>
      <tuv xml:lang="pt"><seg>mundo</seg></tuv>
    </tu>
  </body>
</tmx>
```

Fig. 2. Example of a simple TMX file.

Meta-information can be added at different levels. As the *prop* and *note* elements are open content they can be used mostly for everything. Also, as they can be added at different levels (*header*, *tuv* or *tu*) they make it easy to annotate specific translation units or units variants. Unfortunately there is not a way to aggregate translation units in blocks. This is a problem if you wish to tag each translation unit with the source where the text came from. With TMX we have only two options: create a different TMX for each text source or to tag each translation unit with the text source.

[10] http://www.tecnick.com/public/code/cp_dpage.php?aiocp_dp=article_tmx#tmx

Regarding word annotation, TMX files support is very poor or inexistent and for that matter it will not be discussed in this paper.

As far as the TMX file format is concerned, it is also being used to make available parallel corpora. As an example, check the OPUS[11] project [7], that includes different types of corpora to download in TMX format.

2.3 XLIFF: XML Localisation Interchange File Format

Having a product available in a different language increases the potential market into which it can be sold. With the advent of globalization, industry needs to customize their products so that they can be available to the global market. This customization process is known as localisation. Although some may believe that localisation issues can be solved with translation the fact is that localisation not only includes translation but also adaptation of a product to a country's cultural and legal practices. By cultural practice we may include date/time formats, numeric or currency formats and symbols, e.g.. Also, the diversity of software platforms and technologies means that tools to support localisation are diverse and frequently incompatible with each other. To deal with this issues OASIS XLIFF [12] has emerged as a standard interchange file format for localisation-related data and metadata.

XLIFF is an XML-based format that enables translators to concentrate on the text to be translated. Likewise, since it is a standard, manipulating XLIFF files makes localisation engineering easier: once there is converters written for the source file formats, new tools can be written to deal with XLIFF and without concerns about the original file format. It also supports a full localisation process by providing tags and attributes for review comments, the translation status of individual strings, and metrics such as word counts of the source sentences.

XLIFF is the instantiation of XML localisation Interchange File Format. The XLIFF format grew out of a collaboration between a number of companies, including Sun Microsystems, but was soon brought under the management of an OASIS Technical Committee. In April 2002, the first Committee Specification for XLIFF was published. This is available at [13].

The XLIFF format aims to:

- Separate localizable text from formatting.
- Enable multiple tools to work on source strings and add to the data about the string.
- Store information that is helpful in supporting a localisation process.

In this article we will look specifically to the schema designed to encode parallel corpora.

[11] `http://opus.lingfil.uu.se/index.php`

[12] `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff`

[13] `http://www.oasis-open.org/committees/xliff/documents/xliff-specification.htm`

Again, please be aware that we are simplifying the structure of XLIFF by removing elements are not relevant for the purpose discussed here. A formal view of the macro-structure of a XLIFF alignment document follows:

$$\begin{aligned}
xliff &\leftarrow file^{+}\\
file &\leftarrow header,\ body\\
body &\leftarrow trans-unit^{+}\\
file &\leftarrow header,\ body,\ @source-language,\ @target-language\\
body &\leftarrow trans-unit^{+}\\
trans-unit &\leftarrow source,\ target,\ alt-trans^{*}\\
alt-trans &\leftarrow target\\
target &\leftarrow @xml:lang,\ ...text..\\
alt-trans &\leftarrow source?,\ target
\end{aligned}$$

That is, an alignment XLIFF file can be described as a collection of translation units. Each translation unit contains a sentence or paragraph that is extracted from the original document in an element called <source>, and the translator has to fill a <target> element with the appropriate language translation.

Legacy translations from previous projects can be added to a new translation unit using <alt-trans> elements. Since the header does not have relevant information to compare we will focus on the body section. In XLIFF, the body section contain the trans-unit (translation unit) elements, that are the main elements in an XLIFF file. The trans-unit elements store localizable text and its translations. These elements represent segments (usually sentences in the source file that can be translated reasonably independently). The trans-unit elements contain source, target, alt-trans. The alt-trans (alternative translation) elements represent translation alternatives for the source segment in the trans-unit element. An alternative translation is a translation found in a translation memory, a translation generated by a machine translation system, or a translation suggested by a translator or reviewer. These elements, the alt-trans elements, contain source and target elements. The target elements are the suggested translations of the trans-unit source. The source element represents the text that was matched against, from a TM(Translation Memory) system, for example.

The alt-trans element contains attributes. These provide information about the alternative translations, such as which tool produced them, or in the case of match-quality, a measure of the quality of the translation. The algorithm for generating the match-quality value in a given alt-trans element is specific to the tool that generated it. However, for a translation memory system, it is typically the percentage of words in the source element that match the source from its database.

Figure 3 presents a simple XLIFF file [14].

[14] Obtained at `http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html`

```
<xliff version='1.2'
       xmlns='urn:oasis:names:tc:xliff:document:1.2'>
<file original='hello.txt' source-language='en' target-language='fr'
       datatype='plaintext'>
<body>
<trans-unit id='hi'>
<source>Hello world</source>
<target>Bonjour le monde</target>
<alt-trans>
<target xml:lang='es'>Hola mundo</target>
</alt-trans>
</trans-unit>
</body>
</file>
</xliff>
```

Fig. 3. Example of a simple XLIFF file.

Due to the bilingual aspect of XLIFF it is recommended that the content within the <file> element is uniformly bilingual. In other words, for each <source> and <target> elements there is a child of <trans-unit> that is of the same language as the source-language and target-language attributes of the <file> element, respectively. The xml:lang attribute should not be used in those elements. The exception is that <source> and <target> elements that are children of <alt-trans> may contain an xml:lang attribute of a different language than that of the source-language and target-language attributes of the <file> element.

3 Comparing TEI, TMX and XLIFF

These three formats are different, and they were designed for different objectives. Table 1 compares some of the most relevant features of these formats. Note that we are comparing them with parallel corpora encoding in mind. So, documentation refers on how to use these formats to encode parallel corpora, and dedicated tools, the availability of tools to encode and manage parallel corpora using these formats.

A final decision on what encoding schema to use will highly depend on the objectives. Some examples and decisions you might take:

- Your parallel corpora will be used as a translation memory for machine translation software. In this case, it is clear that TMX format should be chosen;
- You are making available a multi-language corpora, in alignment pairs. It is easier to release each language as a separate XML file, and independent

Feature	**TEI**	**TMX**	**XLIFF**
Documentation	++	+	++
Schema simplicity to encode parallel corpora	−	++	++
Multi-language support	++	++	++
Sentence level alignment meta-data	−	++	++
Word level annotation	++	□	□
Dedicated tools	−	++	++
Availability of encoded corpora	+	++	-

Table 1. TEI, TMX and XLIFF comparison table (++ stands for very good, + for enough, and – for limited support. A □ is used when no support is present).

alignment files for each language pair. This way, the user can clearly choose what file to download.

The decision will also be highly dependent on what tools are available to manage your files. As it is described in the previous table, TMX is well served with tools to manipulate translation memories. From a wide range of computer assisted translation tools, to small GUI tools or even libraries, like `XML::TMX` [1]. TEI is quite served on tools when used as a schema to encode textual document. To manipulate parallel corpora there are just some few scripts developed by researchers that release their corpora in TEI format. Finally, XLIFF is an emerging standard that provides a structured interchange format for localisation data. In the particular part of encoding corpora one can say that XLIFF is better to other kind of functionalities.

4 Conclusions

In this article we gave a brief insight of the three major schemas available to encode parallel corpora. As the previous section showed, if we compare directly the features for each standard, we will end up selecting TEI as the best. It is not just well documented but it also includes in-depth discussion on the schema features. The biggest drawback is related to its embracing philosophy. As all kind of texts can be encoded in TEI it makes it quite difficult to develop robust tools that can handle the full schema.

The TMX format is in the other end of the continuum. It was developed for a specific purpose, it is very simple and fully functional for its main objectives. Being small, makes it quite easy to develop tools manipulating it[15]: all computer aided translation software have import/export facilities for this format.

XLIFF fits between the previous schemas. Is was designed to deal with a specific purpose, localisation, but it is generic enough to embrace a bigger set of documents related with that purpose.

[15] In fact only 90% of the schema is really used on most tools, but this subset includes the most relevant features.

Comparing with XLIFF, TMX is a format to exchange translation memory data from one tool to another. Its purpose is therefore different from XLIFF's, although both formats have a lot in common, especially regarding the inline markup elements. TMX uses only the encapsulation methods for inline codes (there native codes are enclosed within different elements), while XLIFF provides both the encapsulation method (using elements very similar to TMX's) and the placeholder method (where the native codes are removed to the skeleton file and replaced by a short element that refers to them.). TMX allows any number of languages in the same document. XLIFF is designed to work with one source and one target language.

How to chose one of them is a problem. But for sure, the authors do recommend the use of TMX over XLIFF to encode corpora parallel.

The main conclusion we can get from the analysis of these three standards is that both specifications, XLIFF and TMX, contain all the elements necessary to store source document formatting information in XML format. A good CAT (Computer Aided Translation) tool will allow a translator to reuse a sentence from an HTML page when translating a Rich Text Format (RTF) document, keeping text layout intact. This is the key reason why TMX format should be used as a complement to XLIFF. But also that TMX and XLIFF fail to on the word level notation, that only TEI can provide.

Acknowledgments

This work had the contribution of Prof. Alberto Simões.

References

1. José João Almeida and Alberto Simões. XML::TMX — processamento de memórias de tradução de grandes dimensões. In José Carlos Ramalho, João Correia Lopes, and Luís Carríço, editors, *XATA 2007 — 5ª Conferência Nacional em XML, Aplicações e Tecnologias Aplicadas*, pages 83–93, February 2007.
2. Sílvia Araújo, José João Almeida, Alberto Simões, and Idalete Dias. Apresentação do projecto Per-Fide: Paralelizando o português com seis outras línguas. *Linguamática*, 2(2):71–74, Junho 2010.
3. Nancy Ide and Keith Suderman. GrAF: A graph-based format for linguistic annotations. In *Proceedings of the Linguistic Annotation Workshop*, pages 1–8, Prague, Czech Republic, June 2007. Association for Computational Linguistics.
4. I. Dan Melamed. *Empirical Methods for Exploiting Parallel Texts.* MIT Press, 2001.
5. Alberto Manuel Brandão Simões. Parallel corpora word alignment and applications. Master's thesis, Escola de Engenharia - Universidade do Minho, 2004.
6. Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *5th International Conference on Language Resources and Evaluation (LREC'2006)*, Genoa, Italy, 24–26 May 2006.
7. Jörg Tiedemann and Lars Nygaard. The opus corpus - parallel & free. In *Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 26–28 2004.

8. Edward Vanhoutte. An introduction to the TEI and the TEI Consortium. *Lit Linguist Computing*, 19(1):9–16, April 2004.

A Modular MATLAB Compilation Infrastructure Targeting Embedded Systems

Ricardo Nobre

Faculdade de Engenharia da Universidade do Porto (FEUP)
Departamento de Engenharia Informática, Rua Dr. Roberto Frias s/n,
4200-465 Porto, Portugal
Email: ricardo.nobre@fe.up.pt

Abstract. Dynamic programming languages make extremely difficult for static compilation to achieve efficient results. MATLAB is one of the programming languages offering dynamic features and abstractions to easily perform matrix operations. These features, the vast portfolio of domain-specific libraries, and the tools to simulate MATLAB system specifications, make MATLAB a widespread programming language, ranging from embedded to scientific computing domains. However, MATLAB programs are usually seen as models and require aggressive specializations in order to achieve efficient embedded computing implementations. For example, the same MATLAB model may be implemented using double, single, and/or fixed-point precision data types. This kind of specialization needs analysis steps exploring different alternatives, measuring the impact on accuracy and precision errors, before deciding to a specific implementation. Furthermore, a selection of a particular implementation may have to be done according to the target embedded computing architecture. This paper presents a MATLAB compiler infrastructure able to generate different implementations from the same MATLAB input code. Specifically, our MATLAB compiler is modular and can be easily extended by using a strategic programming approach, and specific implementations can be derived by aspect-oriented information specified by the user. We also present the use of the current version of the compiler for translating MATLAB codes to C code implementations for embedded computing systems using different input type specifications, thus simulating different target scenarios.

Keywords: modular, MATLAB, C, compilation, specialization

1 Introduction

MATLAB [2] is a *de facto* standard high-level language and an interactive numerical computing environment for many areas of application, including embedded computing. It is widely used by engineers to quickly develop and evaluate their solutions [3]. By including extensive domain-specific and visualization libraries, MATLAB substantially enhances engineers productivity [3].

The flexibility of MATLAB, however, comes at the cost of interpretation (and JIT[1] compilation) and lack of type/shape information. In MATLAB the same identifier can be used to hold various data types and array shapes (i.e. the number of elements of each dimension of a multi-dimensional array) throughout the code execution. This makes it very handy for quick program prototyping but it is not ideal for static analysis, and in many cases precludes the application of advanced program transformations. The flexibility of MATLAB programs results in lower execution performance. This lack of performance is typically addressed by the development of an auxiliary reference implementation once the MATLAB code has been validated. This amplifies the maintenance costs as the programmer must now deal with multiple language versions of the application.

An alternative approach relies on a compilation tool to perform advanced analyses and to generate a reference C code directly from MATLAB, thus avoiding the lengthy user intervention. This automatic approach, however, is also fraught with obstacles as there are inherent limits to what information static analyses can extract. Rather than pushing the limits of static analysis - sometimes too conservative due to the lack of complementary information about the problem - we have adopted the pragmatic approach of allowing the user to control some of the aspects of the translation process at a very high-level. In our approach the compiler is aware of separate type and shape specifications provided by the user as a vehicle to convey information to the compiler regarding the types and array shapes of the used variables. The compiler uses the user-provided information and complements/checks its consistency against the information it can derive from its own analyses.

As to the compiler infrastructure, we rely on the concept of strategic programming[2] to accomplish a modular and flexible compiler framework. The infrastructure uses Tom[3] to achieve a built-in rewriting system for analysis, transformations, and code generation. The end result is a synergy between compiler analysis and the user that allows the compiler to generate very high quality code from MATLAB specifications and the possibility to generate different C code versions, important when targeting different embedded systems. In addition, the use of a well-known and mature strategic programming system as Tom allows our compiler infrastructure to be easier to extend in comparison to an approach using proprietary and specific data-structures and implementations.

This paper is organized as follows. Section 2 describes the compilation infrastructure. Section 3 presents some experimental results. Section 4 describes related work. Finally, Section 5 draws some conclusions and presents future work.

[1] Just-in-time compilation is a method that compiles code at run-time in order to improve the runtime performance of computer programs.

[2] Strategic programming is "a generic programming idiom for processing compound data such as terms or object structures", relying on the "separation of two concerns: basic data-processing computations vs. traversal schemes". Traversals are composed by passing the former as arguments to the latter [4].

[3] Tom is a language extension that provides pattern matching facilities for manipulating tree structures and XML documents [5,6].

2 Compilation Infrastructure

Our compiler infrastructure is organized according to the component diagram depicted in Fig. 1: a single front-end (*mat2tir*) and two back-ends (*tir2mat* and *tir2c*). The compiler is modular enough to allow back-ends for other output programming languages.

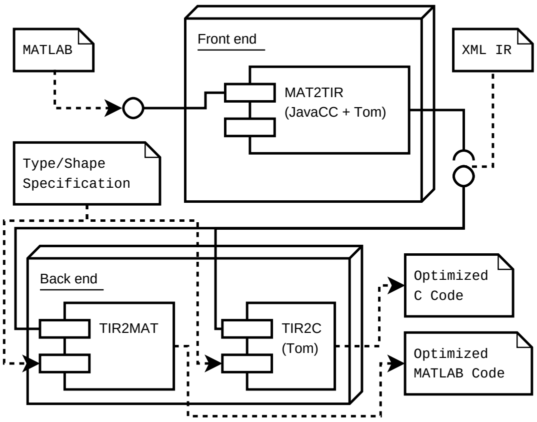


Fig. 1. Component diagram of MATLAB Compilation infrastructure.

The mat2tir component uses a parser generated from an LL MATLAB grammar [4] with JavaCC[5] to parse MATLAB M-files. The abstract syntax tree (AST) generated by the parser is then converted to a Tom [5,6] intermediate representation (IR) exported in XML format. This *mat2tir* component is internally structured as a traditional front-end parser and a Tom IR creator step.

The IR AST grammar is declared by Gom [5,6]. Gom provides a syntax to define ASTs. The Tom IR structure is built from the JavaCC AST by recursively visiting each node, respecting the specification declared using Gom syntax.

The Tom IR represents the input MATLAB code as expression trees directly obtained by the AST produced by the parsing front-end. The IR also represents the line numbers of MATLAB statements, code annotations, and information about types and shapes of the MATLAB expressions. Information about types and shapes is unknown until an inference engine is executed. Annotations are embedded in the MATLAB comments (they start with *%@*) and are used in our approach to identify specific MATLAB code locations (join points), or are inside a separated file loaded by the backend.

[4] LL grammars are a subset of contex-free grammars parsable from left to right, producing the leftmost derivation of the input.

[5] JavaCC is a parser generator for use with Java applications [8]

Table 1 depicts MATLAB code for a recursive factorial, part of the correspondent Tom IR and the generated C code specifying the input variable n as scalar and integer.

Table 1. (a) MATLAB code, (b) XML IR for the factorial function and C code generated from MATLAB with the following type and shape information input to the compiler: *n:int32:1x1.*

function x = factorial(n)
  if n == 1
    x = 1;
  else
     x = factorial(n−1);
     x = n*x;
 end
	<Start>
	<FunctionMFile>
	  <ConcIdentifier>
	   <Identifier>
	    "x" "1"
	    </Identifier>
	   ...
	   "1"
	 </FunctionMFile> "1"
	 </Start>
		#include "tensor.c"
		void factorial(int n, int* x) {
		  if(n==1) {
		    (*x) = 1;
		  } else {
		    factorial(e_minus(n, 1),
		        &(*x));
		    (*x) = e_times(n,(*x));
		  }
		}

a)	b)	c)

A *tir2x* form component then takes the Tom IR as input and, by combining it with the type/shape specifications written by the user, applies specific IR-based transformations or code generation steps. We have developed two backend applications, namely *tir2mat* and *tir2C*.

The MATLAB Generator backend (*tir2mat*) is used for validation purposes and generates MATLAB code from the IR. This MATLAB code is helpful as it gives the user the possibility to compare results obtained by using transformed and/or specialized code with the original code. This can be a way to evaluate and analyze accuracies in the case of type and word-length transformations, and to trace and debug in the case of inserting code for monitoring.

The C Generator backend (*tir2C*) allows us to generate C code from the MATLAB input code with the possibility to use the complementary information added by the user.

The transformations in the Tom IR use Tom [5,6] capabilities to manipulate tree structures. Pattern matching is used to find specific patterns of data-structures in the Tom IR where transformations are applied. The use of Tom allows us to achieve a flexible compiler infrastructure as transformations rules are described using Tom specifications.

Regarding the back-end, and when translating typeless/shapeless languages such as MATLAB to C, a key step is the definition of specific types and in particular array shapes. To this extent we have implemented a limited form of data-flow analysis [9] as well as an external user-provided mechanism for static determination of types and shapes of variables as described next. The idea is that just declaring information about the argument variables of a function, the external knowledge gained by the inference engine is sufficient to statically determine the shapes and types of all, or at least almost all variables.

2.1 Pipeline architecture and Strategies

The compiler is structured as a set of distinct engines attached to a single pipeline. An engine has both accesses to the compiler internal variables, like symbol tables, and the IR representing the input code at each step.

Fig. 2 depicts a diagram representing the main classes backend.

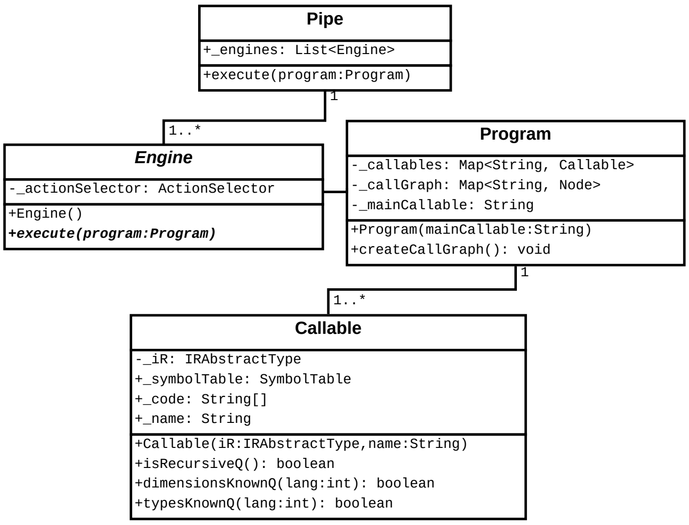

Pipe
+_engines: List<Engine>
+execute(program:Program)
1
1..*
Engine
-_actionSelector: ActionSelector
+Engine()
+execute(program:Program)
Program
-_callables: Map<String, Callable>
-_callGraph: Map<String, Node>
-_mainCallable: String
+Program(mainCallable:String)
+createCallGraph(): void
1
1..*
Callable
-_iR: IRAbstractType
+_symbolTable: SymbolTable
+_code: String[]
+_name: String
+Callable(iR:IRAbstractType,name:String)
+isRecursiveQ(): boolean
+dimensionsKnownQ(lang:int): boolean
+typesKnownQ(lang:int): boolean

Fig. 2. Main classes composing a backend instance.

The main class of the *tir2c* backend, *Tir2C*, instantiates a single pipeline (*Pipe* class) and attaches engines (extend *Engine*) in the order they should execute. Engines can manipulate the IRs representing the input MATLAB files, can change the internal state of the compiler (e.g. symbol tables, generated code), or both. For instance, the C generator engine, *CGeneratorEngine* class, is responsible for generating the C code from the imported IR, using the type and shape information about variables inferred by the inference engine (*InferenceEngineFilter* class) with information directly specified in external specification files, in case this files exist and contain type or shape declarations. The script inliner engine (*ScriptInlinerEngine*) is an example of an engine that inlines IRs for multiple MATLAB script files into a single IR.

The pipeline is executed after the call graph creation method, which returns an instance of the *Program* class holding information about all M-Files, each represented by an instance of the *Callable* class. A Callable object holds all compiler internal variables, symbol tables and IR structure representing an M-File. When executing (*execute()* method) the pipeline the *Program* object is passed as argument, and then each engine attached to the pipeline is executed (*execute()* method) with it passed as argument. Engines have access to all the internal variables (e.g. IRs and symbol tables) of the compiler.

2.2 Engine creation methodology

The execution of an engine is performed by passing each Callable from the *Program* object to a new instance of a class extending an abstract class called *ActionSelector*. The class extending the *ActionSelector* is always described using primitives from the Tom language extension in a *NAMEActionSelector.t* file, where *NAME* represents the name of the engine.

The logic to traverse the IR is encapsulated in *NAMEActionSelector.t* by a Tom strategy that for each Tom IR structure found in the IR calls a method from a class extending the *ActionSelection* class, which encapsulates the only methods that are allowed to change the compiler internal state for a given *Callable* passed as argument.

The *ActionSelector* of an engine acts as a selector for choosing the appropriate method to call from the *ActionSelection* component. Having the logic for the traversal of the IR completely separated from the actions to perform during the traversal of each IR node in separated files allows interesting software development techniques as the *ActionSelector* and the *ActionSelection* parts of each engine require different levels of understanding about the compiler architecture.

Extending the ActionSelector to create a new engine only requires knowledge about the IR and the Tom primitives used to implement strategies for traversing the IR. Extending the *ActionSelection* neither requires knowledge about how to specify Tom strategies nor how the IR is structured, as the *ActionSelection* part only deals with updating the symbol table and the IR and all the methods needed are automatically created when extending the abstract class. Transformations are only specified, using a strategic programming approach, in the class extending the *ActionSelector* abstract class. These classes are represented in 3.

2.3 C Generation

In terms of the implementation, the generated C code relies on a very flexible data structure called tensor. The tensor is used to represent, in C, all array variables. This structure has its dimensions and base type (integer, real and complex) dynamically allocated, thus relying on run-time tests to determine which operation to apply to the elements of the multi-dimensional array, only in case the precise dimensions are not known at compilation time (e.g., when such information is not determined by the static shape analysis stage). The number of elements of the whole structure is also stored for efficiency. We also developed a C tensor library with all the structures and functions to support the most common MATLAB matrix built-in functions.

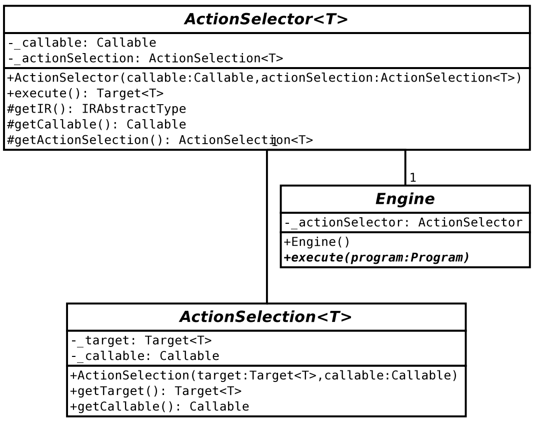

ActionSelector<T>
-_callable: Callable
-_actionSelection: ActionSelection<T>
+ActionSelector(callable:Callable,actionSelection:ActionSelection<T>)
+execute(): Target<T>
#getIR(): IRAbstractType
#getCallable(): Callable
#getActionSelection(): ActionSelection<T>
1
1
Engine
-_actionSelector: ActionSelector
+Engine()
+execute(program:Program)
ActionSelection<T>
-_target: Target<T>
-_callable: Callable
+ActionSelection(target:Target<T>,callable:Callable)
+getTarget(): Target<T>
+getCallable(): Callable

Fig. 3. Action Selection Framework abstract classes.

2.4 User-provided Type and Shape Information

To address the limitations of a static type/shape inference analyses and/or to force implementations according to user knowledge about the target implementations, our MATLAB compiler supports the specification of types and shapes of variables as a form of annotations in a separate file. This mechanism complements the capabilities of the type/shape-inference and allows the compiler to generate code using more accurate type information and/or specialized implementations.

The code generated by our compiler when considering the specification of the type and shape of the arguments of the top function results in a specialized code without type checks which gives an efficient C implementations, both in terms of performance and memory, whenever such specialization can be used in the final implementation. The generated code is clean when compared to human made code.

3 Experimental Results

We carried out a series of experiments to evaluate our MATLAB compiler and the impact of the type and shape information on the performance of the generated C code.

In our experimental evaluation we used a set of 11 code kernels originally written in MATLAB. We used our compiler to automatically derive C codes. These kernels are drawn from simple arithmetic operations in some cases using linear algebra operations such as dense matrix-vector multiplications, from digital signal processing applications, and from an industrial application.

One of the set of kernels used is an industrial code delivered by Honeywell for the REFLECT project [7,22]. The code is related to a Three-dimensional (3D)

Path Planning algorithm. This kernel plans a 3D path between the current position of an autonomous vehicle and a predetermined goal position. In this paper we consider the most computationally intensive task of the 3D path planning algorithm: the gridIterate function. This function consists of 4 nested for type loops and uses 3D arrays. In the experiments we consider a partial map of the environment of size $32 * 64 * 16$.

We then compared the execution time of each sample code using different type and shape declaration files. To compare the performance of the base types the programmer can specify, we carried out experiments where the different base types corresponding to distinct precision requirements were used. These include the specification of double (default), float, and fixed-point precision.

Preliminary tests comparing the number of cycles needed to run executables generated with Mathworks MCC tool[6] and the C code generated by our compilation infrastructure shown that, as expected, the compiled C code is much faster than Mathworks JIT approach (between 2 and 4 times faster). These tests were performed on an x86 microprocessor (in our case the Intel Core 2 Duo), not at all a processor for embedded systems, as it is one of the few platforms compatible with executables generated by MCC (execution requires run-time compiled libraries). Therefore MCC can not be considered as an alternative to our compiler for use with embedded systems.

3.1 Methodology

Experiments were conducted using the SimpleScalar/ARM simulator [10,11] to execute the object output by gcc[7] with the *-O3* flag from the C code generated by our compiler from MATLAB input files. The simulator models Intel StrongARM SA-11xx microprocessors, has been validated against a large collection of workloads, and is believed to be within 4% of real hardware for performance estimation [11]. The StrongARM family of microprocessors implements the ARM V4 instruction set architecture. This simulator reports the cycle count with high accuracy.

Note that we only declared the type and shape of the parameters of each function in the M-files, thus asserting that for all call-sites the same information holds. The information about all other variables is inferred at compilation time by evaluating the expressions where the input parameters are present. Furthermore, for these examples, we relied on type inference to determine the type and shape of all other local variables. This is clearly not the most flexible situation and we will continue to explore scenarios where type/shape specialization makes sense. In addition to the impact of shape information we also tested the performance improvement gained by using single precision floating point rather than the standard double precision.

[6] The MATLAB compiler “prepares MATLAB file(s) for deployment outside of the MATLAB environment, generates wrapper files in C or C++, optionally builds standalone binary files, and writes any resulting files into the current folder, by default.” [23]

[7] The GNU Compiler Collection [24]

3.2 Results and Discussion

We show in Fig. 4 the results obtained when using the SimpleScalar/ARM simulator.

The results consider the execution time improvements between float vs. double, fixed vs. double, and fixed vs. float implementations. The last group of columns (labeled "average") corresponds to the average speedup for our set of code kernels.

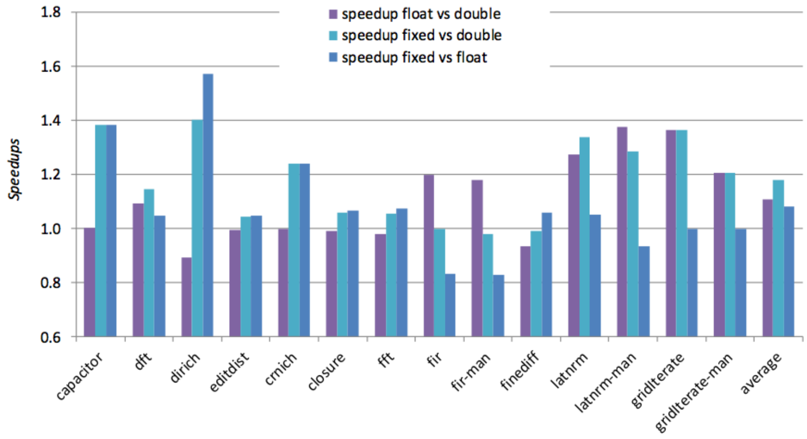


Fig. 4. Speedups for different implementations of the benchmarks.

As the performance results reveal there is on average an increase in performance of 1.16 times and 1.09 times when using the fixed-point arithmetic representation versus double and float precision, respectively. This is mainly due to the use of integer operations, that are faster than instructions operating on float and double data types for the specific ARM architecture used.

The fact that the performance relation between double and float is not consistent between benchmarks (mainly because of float-to-double and double-to-float conversions) is a proof that the specification of types and shapes can be valuable, even when the target architecture does not change. For instance, when considering fixed vs. float implementations, a speedup of 1.57 times is achieved for the dirich code and a slowdown is achieved for the fir code.

4 Related Work

Given the significance of MATLAB in early algorithm design and evaluation, many other research efforts focused on translating MATLAB code to languages that provide a more efficient execution environment (see, e.g., [12,13,14]). Due to the importance to execute directly MATLAB programs there has been research efforts to improve the execution of JIT MATLAB compilers. A recent example is the compiler presented in [15] that performs function specialization based on the runtime knowledge of the types of the arguments of the functions. DeRose and Padua developed the FALCON research project [12] that translates MATLAB to FORTRAN90 code with performance improvements that resemble the ones we report here for codes of similar complexity. They leverage an aggressive use of static and type inference for base types (double and complex) as well as shape (or rank) of the matrices.

Other researchers have also relied on simple type inference approaches such as ours. Banerjee et al [16,17] have used annotations to specify data types and shapes, simple type inference analysis and target VHDL code specifications for hardware synthesis onto Field-Programmable Gate Arrays (FPGAs). Later on, this work resulted on the commercial AccelFPGA compiler which supports, among others, annotations to define the shape of MATLAB variables.

As with the compiler to optimize Octave programs presented in [18] our compiler relies on a strategic programming approach. However, in our approach we extend the strategic programming approach (using Tom in our case instead of the Stratego approach in [18]) with user information regarding types and shapes. One of the closest related work is the embedded MATLAB (a subset of MATLAB) to C code translation existent in the MathWorks Real-Time Workshop [19] which allows the user to embed annotations with MATLAB code to achieve C code implementations for embedded systems [20].

The popularity of the MATLAB language is also reflected in the similar languages proposed. Examples of those languages are Scilab and Octave. Recently, a Scilab to C translator [21], named Sci2C, has been proposed. The Sci2C translator focuses entirely on embedded systems and it is completely dependent on annotations, to specify data sizes and precisions, embedded in the Scilab code. Our compiler distinguishes from Sci2C as it is capable to generate C code even without annotations and specialization of the generated C code can be achieved without polluting the original code (MATLAB, in our case). Furthermore, Sci2C requires that the size of arrays is fixed and statically known while our compiler also produces C code (including calls to realloc()) when those sizes are not statically known.

5 Conclusion

This paper presented a compiler infrastructure for MATLAB. The infrastructure relies on the concept of strategic programming, and especially on the Tom framework to analyse, transform and generate code from a Tom intermediate representation of the input MATLAB program.

The architecture of the compiler infrastructure is sufficiently flexible and modular to allow an easy integration of compilation stages by taking advantage of Tom rewriting capabilities, in the form of what we called the “Action Selection” framework.

One important stage of the compiler is the generation of C code. In order to achieve more efficient and/or specialized C code our approach allows users to add information about types and array shapes. This additional information is of paramount importance in most cases, especially in cases where advanced type/shape inference analysis needs to be conservative, and in the cases where one needs specialized implementations dependent on the target system and/or specific concerns. Additionally, the compiler infrastructure is sufficiently modular to allow the integration of other output languages generators, other than MATLAB and C.

Our ongoing work focuses on the use of an aspect-oriented language to specify more powerful type and shape information (possibly parametrically) as well as high-level code transformations as recently proposed. These code transformations will allow users to easily explore specific code optimizations at the MATLAB level.

The comparison of code generated by *tir2c* with code generated by Sci2C (performance, memory, and readability-wise) is work in progress.

Future plans include a static analysis (possibly further extended with profiling information) to identify automatically the variables in a MATLAB program whose types/shapes are more important for the type/shape inference engine.

Acknowledgment

This research has been partially funded by the Portuguese Science Foundation Fundao para a Ciência e Tecnologia (FCT) under research grant PTDC/EIA/70271/2006.

The author also acknowledges the support of the FP7 EU-funded project REFLECT for the access to MATLAB codes related to industrial applications.

The author acknowledges support from João M. P. Cardoso, affiliated with Faculdade de Engenharia da Universidade do Porto (FEUP) and Pedro Diniz, affiliated with Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento em Lisboa (INESC-ID.

References

1. Ricardo Nobre, Joo M. P. Cardoso, Pedro C. Diniz 'Leveraging Type Knowledge for Efficient MATLAB to C Translation', paper presented at CPC2010, Vienna University of Technology, Vienna, Austria, July 9, 2010.
2. MATLAB the Language of Technical Computing, `http://www.mathworks.com/products/matlab`.
3. Richard Goering, "Matlab edges closer to electronic design automation world" EE Times, 10/04/2004 `http://eetimes.com/electronics-news/4050334/Matlab-edges-closer-to-electronic-design-automation-world`
4. R. Lämmel, E. Visser, and J. Visser, Strategic programming meets adaptive programming, In Proc. 2nd Intl Conf. on Aspect-Oriented Software Development (AOSD'03), Boston, Mass., March 17 - 21, 2003. ACM, New York, NY, USA, pp. 168-177.

5. Jean-Christophe Bach et al.: Tom Manual - Version 2.7, http://tom.loria.fr
6. Emilie Balland, Paul Brauner, Radu Kopetz, Pierre-Etienne Moreau, and Antoine Reilles, Tom: Piggybacking rewriting on java, In 18th International Conference on Term Rewriting and Applications (RTA07), Paris, France, June 26-28, 2007, Springer LNCS 4533, pp. 36-47.
7. João M. P. Cardoso et al., REFLECT: Rendering FPGAs to Multi-Core Embedded Computing, Book Chapter, Reconfigurable Computing, Springer.
8. The JavaCC Home, https://javacc.dev.java.net/
9. A. Aho, J. Ullman, M. Lam and R. Sethi, Compilers: Principles, Techniques and Tools, Addison Wesley, 2006.
10. T. Austin, E. Larson, and D. Ernst, SimpleScalar: An infrastructure for computer system modeling, Computer, 35(2), 2002, pp. 5967.
11. SimpleScalar Version 4.0, Test Releases, http://www.simplescalar.com/v4test.html
12. De Rose, L. and Padua, D. Techniques for the Translation of MATLAB programs into Fortran 90. ACM Trans. Program. Lang. Syst. 21, 2 (Mar. 1999), pp. 28623.
13. Joisha, P. G. and Banerjee, P. 2007. A translator system for the MATLAB language: Research Articles. Softw. Pract. Exper. 37, 5 (Apr. 2007), pp. 535-578.
14. G. Almsi , D. Padua, MaJIC: Compiling MATLAB for Speed and Responsiveness, In Proc. of the ACM 2002 Conf. on Programming Language Design and Implementation (PLDI'02), June 17-19, 2002, Berlin, Germany.
15. Amina Aslam, and Laurie Hendren, McFLAT: A Profile-based Framework for Loop Analysis and Transformations, in the 23rd International Workshop on Languages and Compilers for Parallel Computing (LCPC2010), October 7 - 9, 2010, Rice University, Houston, Texas, USA.
16. A. Navak, M. Haldar, A. Choudhary and P. Banerjee, Parallelization of MATLAB Applications for a Multi-FPGA System, in Proc. of the 9th Annual IEEE Symp. on Field-Programmable Custom Computing Machines (FCCM'01), Rohnert Park, Calif., May, 2001.
17. P. Banerjee, D. Bagchi, M. Haldar, A. Nayak, V. Kim, R. Uribe, Automatic Conversion of Floating Point MATLAB Programs, in proc. of the 11th Annual IEEE Symp. on Field-Programmable Custom Computing Machines (FCCM03), Napa, Calif., 2003.
18. K. Olmos, E: Visser, Turning dynamic typing into static typing by program specialization in a compiler front-end for Octave, in Proceedings Third IEEE International Workshop on Source Code Analysis and Manipulation (SCAM03), 26-27 Sept. 2003, pp. 141- 150.
19. Real-Time Workshop: Generate C code from Simulink models and MATLAB code, http://www.mathworks.com/products/rtw/
20. Houman Zarrinkoub, and Grant Martin, Best Practices for a MATLAB to C Workflow Using Real-Time Workshop, MATLAB Digest - November 2009, http://www.mathworks.com/company/newsletters/digest/2009/nov/matlab-to-c.html
21. Scilab 2 C - Translate Scilab code into C code, http://forge.scilab.org/index.php/p/scilab2c/
22. REFLECT, FP7 EU Project: http://www.reflect-project.eu.
23. MATLAB R2011b Documentation, MATLAB Compiler. http://www.mathworks.com/help/toolbox/compiler/mcc.html
24. GCC, the GNU Compiler Collection. http://gcc.gnu.org/

Towards Using Automatic Development-time Debugging Techniques on Run-time Systems

Nuno Cardoso and Rui Abreu

Universidade do Porto, Faculdade de Engenharia
`nunopcardoso@gmail.com, rma@fe.up.pt`

Abstract. Despite the enhancement on software hardness induced by development-time automatic debugging tools, it is impossible, within the current state of the art, to create fault-free applications. In order to maximize systems availability, survivability, maintainability, and reliability with the lowest human intervention possible, the research on self-healing systems emerged. This position paper discusses some ideas on how to solve the issues that render the direct usage of development-time automatic debugging techniques onto run-time debugging impossible. The main focus is on Spectrum-based Fault Localization (SFL), a statistic-based technique that has proven itself very useful. We concluded that the process of rendering SFL usable in a run-time context is not as trivial as it might seem.

Keywords: Fault Localization, Automatic Debugging, Spectrum-based Fault Localization, Self-healing

1 Introduction

Society is increasingly dependent on technology. With the appearance of computers as a mainstream good, this dependency experienced an exponential growth. In parallel and as a result of this event, software also suffered from an increase in complexity leading to larger amounts of bugs in software and an increasingly harder process of fault detection, localization and correction.

Efforts have already been made in order to alleviate the debugging stress from humans by relaying part of the process to automated mechanisms. Several groups of tools exist and can be broadly categorized as shown in Fig. 1. The path depicted in green is the core subject addressed in this position paper.

Tools such as “Tarantula” [13], “AMPLE” [7] and “Zoltar” [12] already provide automated ways of targeting the probable cause or location of some error during development. In contrast to traditional debugging methods such as the use of tools such as “GDB” [20] or mere log analysis, automated fault localization tools improve the development cycle in several ways: (1) they reduce the time needed

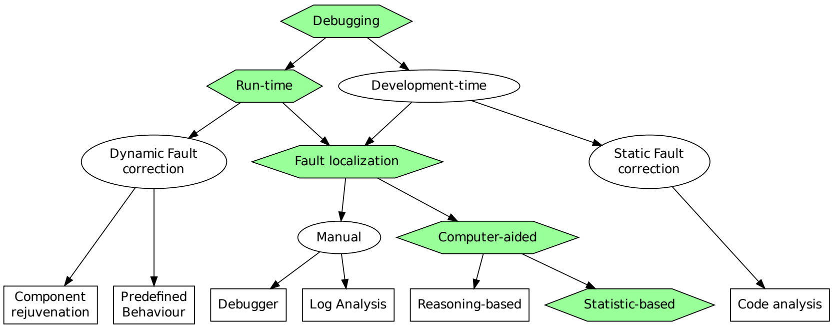


Fig. 1: Debugging tools classes

to find the source of the error, (2) they enforce a much more structured debugging process when compared to the ad-hoc, art-like traditional methods and (3) they enable regression testing.

Despite the great advance such tools represent, they largely rely on the test suite available, both in terms of its coverage and quality [1,3]. It is a well known fact that it is not always easy to create a comprehensive test suite for all of some software's components, either due to time, money or complexity restrictions. Given that, it is clear that such tools are not able to assure fault-free applications, arising the need to develop complimentary tools that are capable of autonomously "heal" failing applications during run-time.

Self-healing systems, have the objective of maximizing their availability, survivability, maintainability, and reliability reducing human intervention to the minimum by employing mechanisms that either prevent or solve eventual errors [14]. Most self-healing systems are composed of core components:

Monitor: Gathers status information from the system through sensors and preprocesses it for the next stage (analyzer).

Analyzer: Determines whether the received information represents a faulty or degraded state of the system and whether some action must be taken. Also, this component may be responsible for pinpointing the probable cause or set of causes of the system failure.

Planer: Creates a repair plan in order to return the system to an operational state. This component is the most challenging one as it is normally hard to accurately predict the impact of some plan through simulation. Also, it is of extreme importance to deliver a plan that (1) solves the failure and (2) solves it in a timely fashion. Due to the complexity of most systems, it is not always trivial to find a solution that meets both requirements.

Executor: Deploys the repair plan conceived in the previous stage.

Knowledge: This represents the knowledge base consumed and produced by all four previously mentioned tasks.

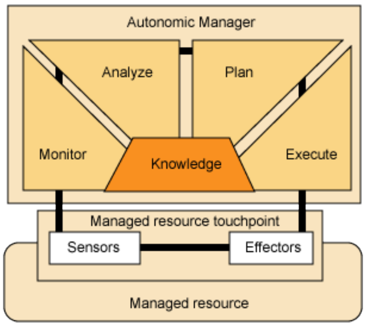


Fig. 2: MAPE-K control loop

This was called the autonomic control loop coined as MAPE or MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge) loop [11] (Fig. 2[1]).

When compared to development-time debugging, failures in run-time systems are not necessarily binary (pass/fail). Issues such as performance degradation, resource hogging among others are problems that may not represent a system failure within certain thresholds but, a close monitoring and analysis of such artifacts may enable the enhancement of the system behavior avoiding failures [16]. The generic state diagram of a run-time system can be seen in Fig. 3. It is clear that it is somewhat difficult to define the limits of some system's degraded state as it heavily depends on thresholds that are not normally agreed among all users (e.g. some threshold latency of response from some component may be acceptable for some and not acceptable for others). The correct definition of the degraded state boundaries is of great importance as, on the one hand small boundaries may cause a direct transition from a normal to a broken state and, on the other hand large boundaries may trigger a large amount of correction actions, further degrading the system's state.

The goal of this position paper is to discuss some issues and possible solutions for the application Spectrum-based Fault Localization (SFL) [5,1,2] technique to run-time fault detection.

The remainder of this paper is organized as follows. Section 2 will give an introduction on SFL applied to development-time debugging. In Section 3 ideas on how to face the issues that make impossible the usage of SFL in a run-time debugging scenario are discussed. In Section 4 we will draw some conclusions on the issues addressed in this paper.

2 Spectrum-based Fault Localization for development-time fault diagnosis

SFL is a lightweight statistic based fault localization technique that takes as its input a form of trace abstraction, called the program spectrum (Fig. 4b)

[1] http://www.ibm.com/developerworks/library/ac-telford/

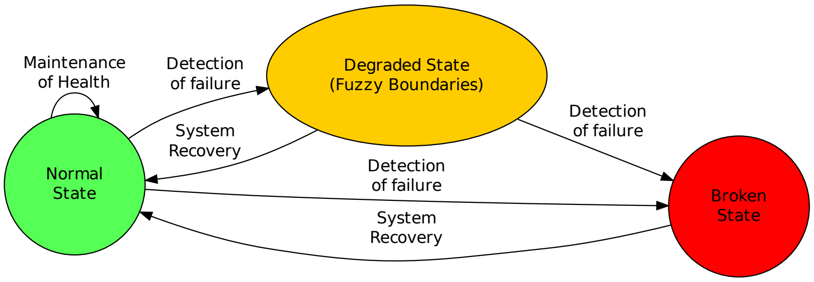
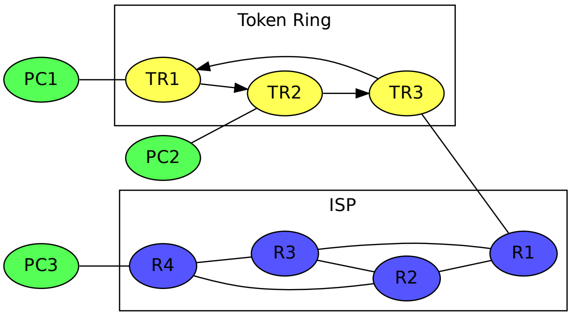


Fig. 3: Run-time system state diagram

and produces a list of likely fault candidates, with the associated probability of being the true fault explanation. The program spectrum consists of a matrix A, known as Program Activity Matrix [1] and a vector e containing the pass/fail information for each run/transaction. A consists of a $N \times M$ binary matrix being N the number of recorded runs an M the number of components in the system (the size of the component depends on the desired granularity, ranging from logical blocks to lines of code). For each row of matrix A, all the components involved in a specific "transaction" (see 3.1.5) are set to true and all the others to false.

With a program spectrum, the process of localizing the probable faulty component consists in finding which component column is the most similar to the error vector. This is done by calculating a similarity coefficient and further details on this issue can be consulted in [4]. Fig. 4 show an example of a network architecture (Fig. 4a) and a set of transactions with the correspondent Activity Matrix (Fig. 4b). The transactions in this illustrative example are UDP [17] messages sent from component x to component y ($x \rightarrow y$) and the pass/fail information is obtained from the checking the UDP header's checksum on the destination node. According to SFL and without much calculations, it is clear that components "PC1" and "TR1" are the most probable failing components as they are the only ones whose activity during all transactions matches the error pattern.

SFL has the advantage of not requiring detailed system models, using instead probabilistic techniques to infer candidates based on observations. Studies on SFL showed that it is an useful technique for reducing the scope of debugging. Even with low quality test suites, 80% of the initial code was considered not to be a potential fault location[4,6].



(a) Architecture: Computer network

	Component										
Transaction	PC1	PC2	PC3	TR1	TR2	TR3	R1	R2	R3	R4	Error
PC2→PC1	1	1	0	1	1	1	0	0	0	0	1
PC2→PC3	0	1	1	0	1	1	1	0	1	1	0
PC1→PC2	1	1	0	1	1	0	0	0	0	0	1
PC3→PC1	1	0	1	1	0	1	1	0	1	1	1

(b) Program Spectrum example

Fig. 4: SFL example

3 SFL for run-time fault diagnosis

While impressive diagnostic results for design-time testing and debugging have been achieved, little work has been performed in applying SFL to run-time fault detection and repair. A large number of issues exist that invalidate a direct application of this technique into run-time systems.

3.1 Issues on SFL for run-time fault diagnosis

In this section we will discuss some ideas on how to solve the issues that render impossible the direct usage of development-time SFL techniques in run-time fault localization. The proposed solutions are purely speculative and are yet to be thoroughly tested.

3.1.1 Fuzziness In run-time there is much less control, with much more variables environment than in development-time. As discussed earlier, there is a gray zone, the degraded state (Fig. 3), in which a binary pass/fail flag is too simplistic and may hide certain errors. For instance, a component that takes a long time to output the correct information could either be considered faultless or faulty depending on a timeout enforcement. Similar scenarios exist and some adaptation is needed in order to cope with them.

One possible solution for this issue is to add one extra vector to the program spectrum depicting the quality of the transaction. This vector enables the detection of soft-faults and a better incorporation with the self-healing state diagram presented earlier (Fig. 3).

Run-time fault diagnosis is majorly focused on finding dynamic solutions for dynamic problems. Mixing this quality of service information with additional per-component probing, it should be possible to better tailor remedies depending on the fault symptoms. Techniques such as software rejuvenation [15,21], in which components are restarted in order to improve the system's robustness, could be used much more wisely and also be restricted to cases in which this type of behavior makes sense.

3.1.2 State variability Another issue on applying SFL to run-time fault diagnosis is the precondition that same test suite should always deliver the same results. In run-time this preconditions doesn't hold anymore as the system isn't normally reset before each transaction and so, a faulty transaction may influence the next ones, possibly mangling the conclusions taken from the program spectrum.

An important open question related to this state variability consists in defining the number of transactions to keep track and how the distance from the diagnosis time target influences the diagnosis. Different systems present different dynamic behaviors and consequently the time span in which an observation is valid must be determined. This topic has an enormous complexity on its own and we predict that the accuracy of this parameter will strongly impact the quality of the framework.

Still on the matter of state variability, triggering a correction action poses a problem on the classic SFL technique. Two scenarios exist: the component subjected to a corrective action (1) maintains its column in matrix A or (2) a new column is created. For (1), new data may confuse the algorithm and produce erroneous predictions. As for (2), there is the advantage that the corrected component is treated as a new component and no adaptation is needed to the classic algorithm. Secondly, it is easier to create knowledge (the fifth phase of MAPE-K) on whether some corrective action actually works for a certain set of symptoms.

3.1.3 Dimension When applying SFL to a high dimension application, and considering the worst case scenario of instrumenting to the granularity of the statement, each transaction will have a significant impact in terms of resource utilization. For instance the Linux kernel version 3.2 has around 11.5 million lines of code. Each transaction record would occupy approximately 1.3 MB (in a dense binary matrix). An interesting option would be the creation of hierarchies with different levels of granularity. One challenge to this solution would be how to decide the inspection granularity of each components. This decision could be either human based or dynamic, based on previous gathered knowledge of the system. For instance, if a component only appears in successful and high quality transactions, it should be more or less safe to discard the high granularity details about its inner processes. On the other hand, components with a high probability of being faulty should be inspected with a higher degree of detail.

3.1.4 Test oracles In order to have the pass/fail information required by SFL, it is mandatory to have some kind of mechanism that evaluates the result of the transaction. In development-time debugging, this is done either by creating test cases manually or having some reference implementations that is known to be faultless. Some work ([3,8,9]) has already been done in evaluating the possibility using program invariants in order to provide pass/fail information at low cost. Program invariants are conditions that if broken the program is guaranteed to be in a faulty state. Such conditions may be user defined, naive or machine learned. User defined invariants may be defined in the code or in a model definition of the application and may manifest themselves in the form of exceptions or through specific framework calls. Naive invariants are those that are relatively easy to detect such as invalid pointer addresses or deadlocks. Machine learned invariants must be subjected to a prior training in order to perform adequately. Examples of machine learned invariants are array bounds or range detection. Once again, the effectiveness of machine learned invariants deeply relies on the quality and quantity of training done. Also, the use of machine learned invariants may introduce the existence of false negatives and, even worse, false positives.

3.1.5 Transactions Throughout this paper we used the term transaction to define a delimited set of operations. During this transaction we collect data to create the program spectrum. In development-time the transaction is the period in which a test from the test suite is active. When it comes to run-time systems, it is not trivial to define for every system and every component what a transaction is. On the one hand we have systems in which it is easy to define a transaction, for instance in a web server a transaction might be defined by the set of all operations done upon a request. On the other hand, we have systems in which it is not clear what a transaction represents: in a word processor, for instance, it is not easy to define what are the existent transactions. In order to solve this problem the framework must have some manner of defining boundaries in the application operation and enforce an oracle verification inside them.

3.1.6 Instrumentation In order to collect the data needed to create the program spectrum it is required that the target application or component is instrumented. Even though the performance impact derived from the instrumentation was reported to be low, averaging 6% loss [1], in run-time systems it can represent an issue. Further research must be done in order to optimize instrumentation either by applying a dynamic instrumentation or by optimizing the current methods.

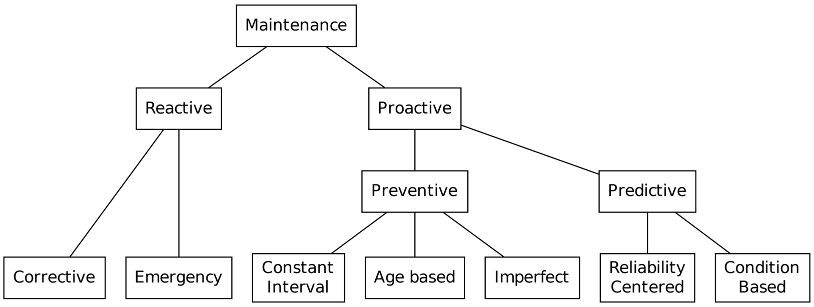


Fig. 5: Maintenance taxonomy

3.2 Further Research on SFL Improvement

In this section some ideas are presented on how to improve SFL functionality after it becomes adapted for run-time fault diagnosis ("Future Future-work").

3.2.1 Proactive maintenance Most of the time and as discussed earlier, there is a gradual transition from healthy state to a faulty state. With SFL we expect to have the possibility to acknowledge this transition and trigger mechanisms of proactive maintenance. Fig. 5 shows the taxonomy of software health maintenance as described in [16]. Reactive strategies are the ones to employ when the system is in a faulty state. For those cases SFL is used to target the faulty components. The more interesting and cost effective strategies are the proactive ones. Preventive strategies are based on heuristics such as age or mean time between failures of each component. Predictive strategies use knowledge from the system in order to perform maintenance tasks on the most troublesome components. The decision on which components maintenance is needed may be based either on their reliability score or on the observed condition. Using SFL we expect to have a reliable prediction of which components should be subjected to predictive maintenance thus increasing the overall system's survivability.

3.2.2 Improving system's self-awareness One important factor for good decision making in self-healing systems is self-awareness. In the scope of this paper we aim at proposing simple ways of improving some application self-awareness capabilities.

The first simple way of improving self-awareness would be to use a test suite that periodically probes the system's behavior in terms of correct response and performance. Such test suit should be first ran in an healthy system to create a reference. Afterward, during run-time, the same test should be re-ran either in predefined intervals or during idleness and the results compared to the reference to assess the system's health. This approach requires some human effort as, as far as we know, no automatic way of creating such test suites exists. However, with some adaptation, it should be possible to adapt unit tests, which are normally available for regression testing, to create the run-time test suite.

Another factor that might improve self-awareness would be to have means of measuring the success of some maintenance procedure. This measurement should enable the application to decide which action should be taken in order to self-adapt based on previous observations. This kind of reasoning can eliminate events such as recurrent triggering of corrective actions that do not solve the actual problem.

The discussed ideas are purely speculative and need further research in order to draw some conclusions on their effectiveness.

3.2.3 Visualizing System's Status From the experience of tools such as GZoltar [18,19] we have the opinion that a well designed visual interface improves the usability and effectiveness of fault localization tools. By having a model description of the system, based, for instance, in the ACME [10] specification, the status and the collected knowledge about the system could be made available. This way, users could more easily connect probabilist information on the source of the failures and the relationships between components.

Fig. 6 shows a very early prototype of the final interface. In this interface, we can see on the left the system's architecture, in which we have the different components of the system and their current status. By selecting them, we can plot their status over time on the right and this way we can detect possible interactions between components. In the picture we can see that the purple graph corresponds to "Node 3" and the yellowish graph to "Node 8". In this plot it is also possible to include the values of the different components probes in order to understand the reason of their malfunction. On the bottom we can see the overall health of the system. The colors range from green to red, being the first correspondent to a healthy state and the second to a broken state.

Future work on this interface would include the possibility to have different levels of detail, as discussed earlier for SFL, and zoom on some component to see its sub-components. This way the user has the ability to focus on the affected area of the system, not being overwhelmed by non important information for the issue being addressed. This behavior can be observed in the GZoltar "root-change" feature [19].

4 Conclusions

In this paper we have proposed some solutions to the issues that render impossible the usage of SFL on run-time fault localization.

At first sight, it appears trivial to apply the SFL technique to run-time fault localization although, in practice, several issues related to the environment differences between the two worlds

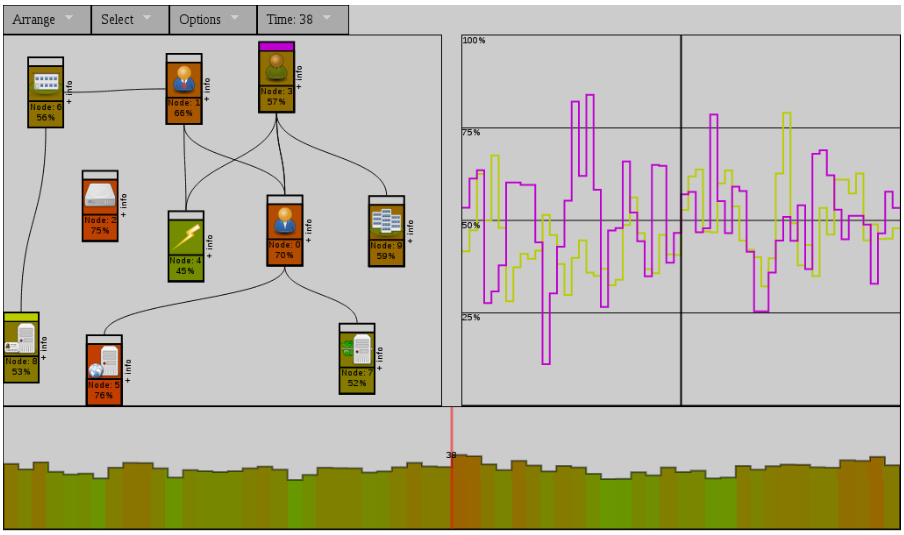


Fig. 6: User interface prototype

SFL has already shown its potential on detecting the possible fault sources in development-time debugging and, despite the great amount of adaptations needed, we expect similar results in an early stage of framework development.

Regarding the possible improvements on the basic SFL implementation for run-time fault localization, a vast universe of options exist. Development-time debugging is a much more advanced research area and we should try to import ideas and techniques that already proved themselves useful, as is the case of SFL.

References

1. Abreu, R.: Spectrum-based Fault Localization in Embedded Software. Ph.D. thesis, Delft University of Technology (November 2009) 2, 3, 4, 7
2. Abreu, R., van Gemund, A.J.C.: Diagnosing multiple intermittent failures using maximum likelihood estimation. Artif. Intell. 174(18), 1481–1497 (2010) 3
3. Abreu, R., González, A., Zoeteweij, P., van Gemund, A.J.C.: Automatic software fault localization using generic program invariants. In: Proceedings of the 2008 ACM symposium on Applied computing. pp. 712–717. SAC '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1363686.1363855 2, 7
4. Abreu, R., Zoeteweij, P., van Gemund, A.J.C.: On the accuracy of spectrum-based fault localization. In: Proceedings of the Testing: Academic and Industrial Conference Practice and Research Techniques - MUTATION. pp. 89–98. IEEE Computer Society, Washington, DC, USA (2007), http://dl.acm.org/citation.cfm?id=1308173.1308264 4
5. Abreu, R., Zoeteweij, P., van Gemund, A.J.C.: Spectrum-based multiple fault localization. In: ASE. pp. 88–99 (2009) 3

6. Abreu, R., Zoeteweij, P., Gemund, A.J.C.v.: An evaluation of similarity coefficients for software fault localization. In: Proceedings of the 12th Pacific Rim International Symposium on Dependable Computing. pp. 39–46. PRDC '06, IEEE Computer Society, Washington, DC, USA (2006), http://dx.doi.org/10.1109/PRDC.2006.18 4
7. Dallmeier, V., Lindig, C., Zeller, A.: Lightweight bug localization with ample. In: AADEBUG. pp. 99–104 (2005) 1
8. Ernst, M.D., Cockrell, J., Griswold, W.G., Notkin, D.: Dynamically discovering likely program invariants to support program evolution. In: Proceedings of the 21st international conference on Software engineering. pp. 213–224. ICSE '99, ACM, New York, NY, USA (1999), http://doi.acm.org/10.1145/302405.302467 7
9. Ernst, M.D., Perkins, J.H., Guo, P.J., McCamant, S., Pacheco, C., Tschantz, M.S., Xiao, C.: The daikon system for dynamic detection of likely invariants. Sci. Comput. Program. 69, 35–45 (December 2007), http://dl.acm.org/citation.cfm?id=1321774.1321800 7
10. Garlan, D., Monroe, R., Wile, D.: Acme: an architecture description interchange language. In: CASCON First Decade High Impact Papers. pp. 159–173. CASCON '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1925805.1925814 9
11. Ibm: An architectural blueprint for autonomic computing. Quality 36(June), 34 (2006), http://users.encs.concordia.ca/~ac/ac-resources/AC_Blueprint_White_Paper_4th.pdf 3
12. Janssen, T., Abreu, R., Gemund, A.J.C.v.: Zoltar: A toolset for automatic fault localization. In: Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering. pp. 662–664. ASE '09, IEEE Computer Society, Washington, DC, USA (2009), http://dx.doi.org/10.1109/ASE.2009.27 1
13. Jones, J.A., Harrold, M.J.: Empirical evaluation of the tarantula automatic fault-localization technique. In: Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering. pp. 273–282. ASE '05, ACM, New York, NY, USA (2005), http://doi.acm.org/10.1145/1101908.1101949 1
14. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer 36, 41–50 (January 2003), http://dx.doi.org/10.1109/MC.2003.1160055 2
15. Kolettis, N., Fulton, N.D.: Software rejuvenation: Analysis, module and applications. In: Proceedings of the Twenty-Fifth International Symposium on Fault-Tolerant Computing. pp. 381–. FTCS '95, IEEE Computer Society, Washington, DC, USA (1995), http://dl.acm.org/citation.cfm?id=874064.875631 6
16. Kothamasu, R., Huang, S.H., VerDuin, W.H.: System health monitoring and prognostics - a review of current paradigms and practices. The International Journal of Advanced Manufacturing Technology 28(9-10), 1012–1024 (2006), http://www.springerlink.com/index/10.1007/s00170-004-2131-6 3, 8
17. Postel, J.: User datagram protocol. RFC 768, Internet Engineering Task Force (August 1980), http://www.rfc-editor.org/rfc/rfc768.txt 4
18. Riboira, A., Abreu, R.: The gzoltar project: a graphical debugger interface. In: Proceedings of the 5th international academic and industrial conference on Testing - practice and research techniques. pp. 215–218. TAIC PART'10, Springer-Verlag, Berlin, Heidelberg (2010), http://dl.acm.org/citation.cfm?id=1885930.1885961 9
19. Riboira, A., Rodrigues, R., Abreu, R.: Interactive visualizations of automatic debugging reports (2011) 9

20. Stallman, R., Pesch, R., Shebs, S.: Debugging with GDB: the GNU source-level debugger. A GNU manual, GNU Press (2002), http://books.google.com.au/books?id=32s_AQAAIAAJ 1
21. Trivedi, K.S., Vaidyanathan, K.: Software aging and rejuvenation. In: Wiley Encyclopedia of Computer Science and Engineering (2008) 6

Session VIII

Affective Computing & Ubiquitous Computing

Chairman: Ricardo Nobre

Wireless mobile technology in physiological signal processing - Physiology as an asset for security
Pedro Ferreira and Eurico Carrapatoso

Satisfaction in Group Decision Support Systems based Meetings
João Carneiro

Wireless mobile technology in physiological signal processing

Physiology as an asset for security

Pedro Miguel Ferreira[1], Eurico Carrapatoso[2]

[1,2]Faculty of Engeneering of the University of Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto PORTUGAL,
[1]pro11021@fe.up.pt

Abstract. Any given human body generates electrical signals, known as physiological signals, which, by their nature, can be collected, transmitted, stored and interpreted in order to, among other purposes, assess state of emotion. This paper presents a system, tailored to validate wireless physiological signal transmission and processing feasibility, based on low budget wireless mobile equipment for a fight-or-flight response detection using artificial neural networks. The system was built on current theoretical knowledge on psychophysiology, regarding heart rate, in fight-or-flight emotional response. The system has not been tested on human subjects but still was found feasible with natural but undetermined constraints on power consumption related to wireless communication.

Keywords: Affective Computing, Physiological Signals, Artificial Neural Networks, Wearables, Real-Time Systems.

1 Introduction

Affective Computing is a relatively novel field of research that seeks to understand interaction of humans with computer systems. Some Affective Computer workgroups are committed to dim the line between computer systems and humans, thus making interaction as transparent as possible, whilst other groups are devoted to giving machines the ability to feel and reason. Understanding how emotions work, how they affect the human body and, in the end, if emotions are differentiable considering only physiological signals space is essential. For the moment there are a handful of theories and models for human emotion: somatic theories; neurobiological theories; cognitive theories; other theories. Nevertheless there is common ground between the theories.

Some Affective Computing investigators have been devoted to building wearables [23] which are usually an assortment of sensors and signal processing unit or units. These wearables have been put to use in many distinct scenarios such as collecting the user's physiological signals, to log or process them, in order to make changes in computer system interface or even allow multimedia emotional interactive systems. This study is based on a research [17] on wearable for driver stress assessment. The base line for the work described in this paper is that physiological changes occur, in heart rate, whenever fear arises as elicited emotion and that there is current market mobile equipment that may be used to build a wearable person based security system. Current handheld equipment allows wireless communication on Bluetooth technology and wireless internet/intranet access which are fundamental to connect to wireless sensors and to access internet/intranet based databases and services.

To test the effectiveness of the proposal a computer application was implemented, on J2ME mobile platform, which receives, processes, stores and performs automated evaluation of physiological cardiac signal in search for evidence of fight-or-flight emotional state.

2 Background and previous work

One of human basic needs, as stated by Maslow[1], is the need for security. Every healthy human being feels the need to maintain and preserve body integrity. This fundamental need is at risk on account of basic factors like: society's heterogeneity; individual moral evolution diversity; inequality of opportunities. And in times of economic crisis, such as the one currently settled, a social crisis usually follows, crime occurrences raise which, in turn, lowers society perception of security.

Existing security systems are mainly based on human intentional alarm activation. These systems fail on principle because intentional alarm activation, in situations such as the attempt to take a valuable object by force, or threat of force, usually puts the victim in state of fear. Self-preservation sense will be overwhelming, for untrained individuals, to the point that alarm activation will be the least of worries. The disabling condition, fight-or-flight response, may be key factor in developing personal tailored alarm systems which do not depend on will power for activation.

2.1 Emotion

Psychophysiology

The human body is endowed with a complex communication and control network, called Nervous System (SN), which allows proper interaction with both internal (body and its subsystems) and external environments (the world outside the body) [2]. The nervous system includes: sensors – to perceive events from the environment; integrators - to process and store data; components and engines that generate behavior and activate gland secretions. Organized by parts or layers, the nervous system is divided into two main components: the central nervous system (CNS) and Peripheral Nervous System (PNS) [2]. The CNS, comprising the brain and spinal cord, is responsible for, among other functions, cognition, memory and learning, reflex responses and behavioral responses. Learning and memory are forms of information processing that al-

[1] Maslow's hierarchy of needs was proposed by Abraham Maslow in his 1943: http://en.wikipedia.org/wiki/Maslow's_hierarchy_of_needs, visited 2-Jan-2012.

low changes in behavior or state, based on previous experiences, in order to cope with environment demands. Many of the activities of the body such as breathing, heartbeat and digestion take place without conscious intervention. One function of the autonomic nervous system (ANS) is maintaining the internal state of the body to achieve homeostasis. This sub-system of the nervous system participates in determining appropriate and coordinated response to environmental stimuli. The response to presence or perception of a threat is massive. Sympathetic nervous subsystem forces the secretion of the adrenal gland, increases heart rate and blood pressure, dilates the bronchioles, inhibits intestinal motility, increases the metabolism of glucose, dilates the pupils, promotes piloerection and forces dilation of blood vessels of skeletal muscles. This is the response, to intense fear or panic, known as fight-or-flight.

Emotion and psychophysiology

Studies have shown that emotions are related to increased activity in certain brain areas, more specifically in the areas of the limbic system [8]. This system is formed by the hypothalamus, the hippocampus, located in the temporal lobe, and the amygdala. It is well known, and accepted, that this system is responsible for characterization of emotional events labeling them as positive or negative [8].The limbic system, which contributes for flexibility and unpredictability in human behavior, has connections with the cerebral cortex. The brain is considered to be the result of two systems, limbic and cortex, which interact and function as a black box processing the inputs and generating the correspondent outputs [8]. For the neuroscientist Antonio Damasio [24], primary emotions are innate and quick reactions and correspond to emotions of fear, surprise and anger. This author also states that the essence of emotion can be seen as the set of changes in body state and that these changes have repercussions in various organs. This dedicated system responds to thoughts directed to a particular entity or event. Some body's status changes - for instance body posture and facial expression - are visible to outside observers. Other changes are only noticeable to the owner of the body. This neuroscientist also states that the emotion "is the combination of a mental evaluation process, simple or complex, with dispositional responses to that process, mostly directed at the body itself, resulting in an emotional state of the body, but also addresses the very brain (neurotransmitter nuclei in the brainstem), resulting in additional mental changes (...)"[24].

Any given organism will perceive an internal or external stimulus in his own way, eventually distinct from any other, and any given organism may perceive identical stimulus with different degrees of intensity in different days. Any given subject’s biological internal state affects His mental state and vice-versa. Every subject is the result of a tendency toward internal equilibrium (homeostasis) which depends on current internal state and current environmental stimuli.

In scientific literature there is no conclusive connection between any current theory of emotion and a set or sets of physiological signals. There is still a too high inter and intra subject variability. The system and algorithm depicted in this paper are, to some extent, empirical and based on the fact that physiological changes do happen. These changes may they be in response to environmental or internal stimuli, thus generating emotion, or may be in response to an emotion, thus somatizing it, generating corresponding physiological changes. In any of the theories it is common that in a fight-or-flight response physiological changes occur and, among those changes, heart rate usually rises in abrupt way like in an effort scenario.

2.2 Related work

One of the works in Affective Computing, related to this paper, was implemented by Jennifer Healey [17] in which its presented study and set of experiences to investigate the correlation between physiological signals and physical activities like walking, running or coughing.

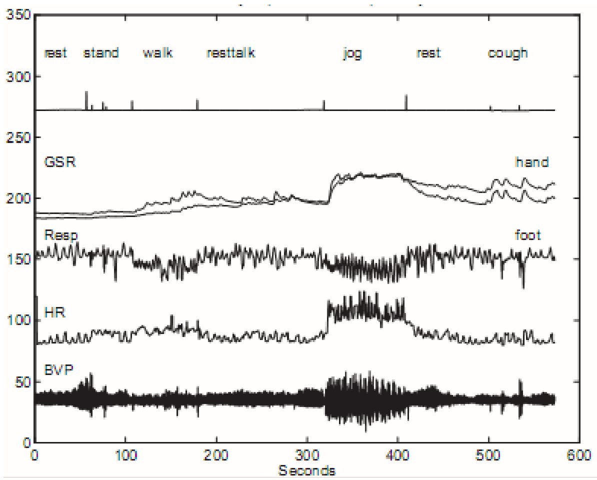
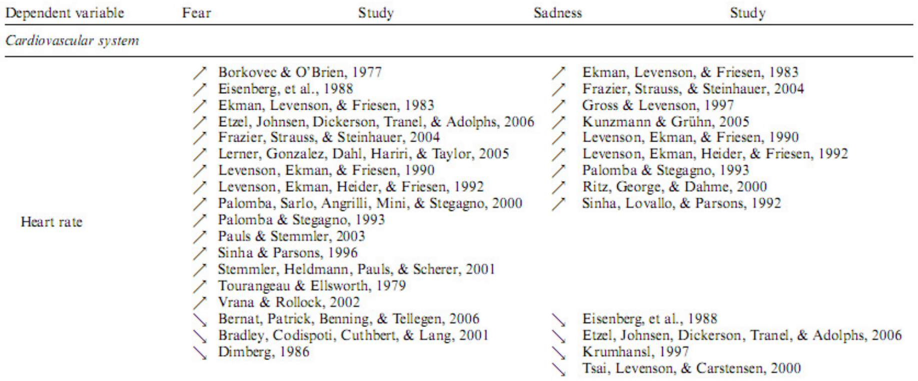


Fig. 1. Physiological signals during specific activities [17]

In previous figure, figure 1, it may be noted that the more reactive indicator is heart rate (HR). Heart rate follows activity changes, increasing or decreasing in excitement or relaxation. Figure 2, bellow, depicts a set of studies and their findings on how fear and sadness affect heart rate. Not all studies present the same effects.



Dependent variable	Fear	Study	Sadness	Study
Cardiovascular system				
Heart rate	↗	Borkovec & O'Brien, 1977	↗	Ekman, Levenson, & Friesen, 1983
	↗	Eisenberg, et al., 1988	↗	Frazier, Strauss, & Steinhauer, 2004
	↗	Ekman, Levenson, & Friesen, 1983	↗	Gross & Levenson, 1997
	↗	Etzel, Johnsen, Dickerson, Tranel, & Adolphs, 2006	↗	Kunzmann & Grühn, 2005
	↗	Frazier, Strauss, & Steinhauer, 2004	↗	Levenson, Ekman, & Friesen, 1990
	↗	Lerner, Gonzalez, Dahl, Hariri, & Taylor, 2005	↗	Levenson, Ekman, Heider, & Friesen, 1992
	↗	Levenson, Ekman, & Friesen, 1990	↗	Palomba & Stegagno, 1993
	↗	Levenson, Ekman, Heider, & Friesen, 1992	↗	Ritz, George, & Dahme, 2000
	↗	Palomba, Sarlo, Angrilli, Mini, & Stegagno, 2000	↗	Sinha, Lovallo, & Parsons, 1992
	↗	Palomba & Stegagno, 1993		
	↗	Pauls & Stemmler, 2003		
	↗	Sinha & Parsons, 1996		
	↗	Stemmler, Heldmann, Pauls, & Scherer, 2001		
	↗	Tourangeau & Ellsworth, 1979		
	↗	Vrana & Rollock, 2002		
	↘	Bernat, Patrick, Benning, & Tellegen, 2006	↘	Eisenberg, et al., 1988
	↘	Bradley, Codispoti, Cuthbert, & Lang, 2001	↘	Etzel, Johnsen, Dickerson, Tranel, & Adolphs, 2006
	↘	Dimberg, 1986	↘	Krumhansl, 1997
			↘	Tsai, Levenson, & Carstensen, 2000

Fig. 2. Effect of fear and sadness on heart rate [18]

Another related work is “The Mobile Patient and the Mobile Physician Data Access and Transmission” [3]. That paper describes a system composed of a wearable device - VivoMetrics LifeShirt®, a Personal Digital Assistant (PDA) and internet connection to a base station (server). In this project the wearable records physiological signals with submission done later through PDA or desktop computer. The wearable has the ability to send data, via Bluetooth, but does not process data.

2.3 Automated emotion recognition

Processing physiological signals has been one of the approaches, with varying degrees of success, in mapping signals space to emotional state space [8]. If there is a correlation between physiological signals and emotional state, of any given organism, we can say it is a privileged way to recognize the relationship between man and the environment. These signals can become, progressively, elements of characterization, classification and prediction of an individual's emotional state.

The most effective methods and, therefore, more used methods in automated emotion recognition are support vector machines (SVM) and artificial neural networks (ANN). For this work ANN with preprocessing of data was chosen for the classification.

Artificial neural networks (ANN).

An artificial neural network (ANN) is a paradigm of data processing based on the structure and organizational model of the human brain. An ANN is a set of artificial neurons, with weights associated with each connection (input), and depending on the combination of activation function of the neuron with the weighted sum of inputs it produces an output. The ANN can perform learning by modifying its parameters including the weight and value of activation. The ANN is specialized in a particular subject and often used in activities of classification and pattern recognition. ANN presents favorable results and good tolerance to lack or failure of input values. The concept of artificial neuron was created by Warren McCulloch and Walter Pitts [8]. The weights (W) associated with each input connection (X) are determined in learning stage. The bias, not mandatory, is the activation limit for the neuron. An activation function is applied to the computation of the weighted sum of inputs.

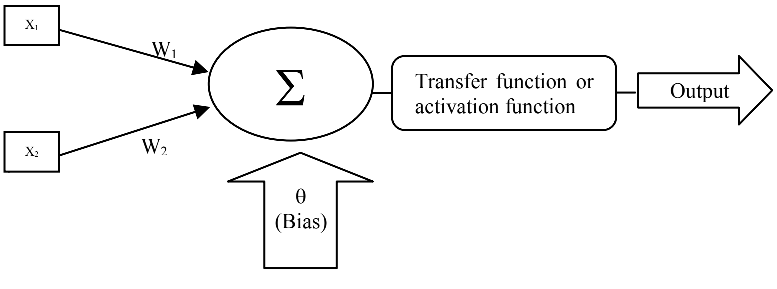


Fig. 3. Generic neuron in ANN

3 Proposed framework and method

The developed system consists of three distinct modules: the PDA - collects and processes physiological signals (client); module for processing requests for service and storage of physiological signals in a database (local server); virtual emulated ECG sensor module.

3.1 ECG Data

To study physiological signals and correlations, some sets of records, found at Physionet, MIT (physionet.org), were used. This service provides several online databases of physiological trustworthy signals extracted from multiple individuals. Among them are the data sets made available by Jennifer Healey [17] during an investigation on stress in drivers. These data sets are multiparameter records (ECG, GSR, Respiratory rate and BVP), of healthy volunteers, obtained in regular office activities and during driving in the city of Boston.

3.2 Framework

Building a wireless and mobile personal security system, able to perform physiological signal transmission and processing, based on wireless low budget technology, requires equipment such as: handheld Smartphone or PDA; wireless ECG sensor; computer acting as server for stored data (database). To assess feasibility of this current proposal we chose to emulate these equipments. The compromise solution was built on Sun's Java Wireless toolkit which allows emulation of smartphone with *Bluetooth* and *wifi* capabilities. The ECG sensor was also virtualized using the same toolkit. The virtual ECG sensor was built on the same principles of current marketed solutions and displays/sends integer values as current subject's heart rate. On account of data to be transferred, between the ECG sensor and the handheld equipment, being simple integer values makes this virtualization as plausible as any real setting environment. No problems were expected regarding data transfers, in what comes to transmission delays, which could impair the emulated systems approach.

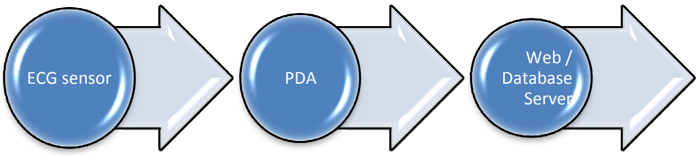
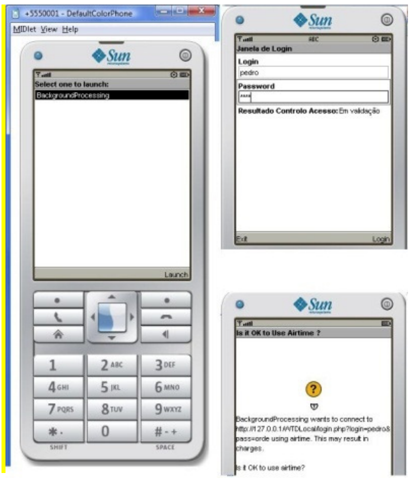


Fig. 4. Proposed Framework

In this projected framework the ECG sensor is responsible for sending, every 2 seconds, a numerical integer value (heart rate). Values will be received by the PDA/handheld equipment. The PDA will perform necessary preprocessing of data collected and feed the results to the ANN. Preprocessing simplifies data and eases ANN workload in performing the classification of emotion state according to previous learning. Any generated "stress" alarms will also be recorded in a database table. Figures 5 and 6 depict aspects of the Sun's Java Wireless toolkit emulated equipment. Reference heart rate can be raised or lowered by pressing keys 2 or 1, respectively.


+5550001 - DefaultColorPhone

Select one to launch:
BackgroundProcessing

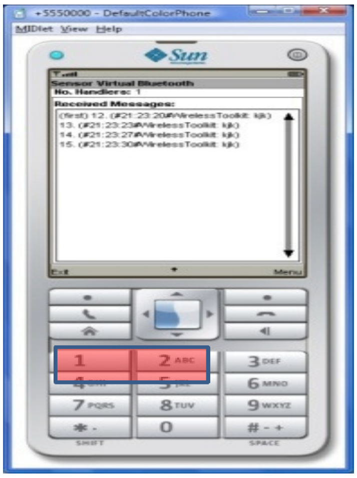
Janela de Login
Login
pedro
Password
Resultado Controlo Acesso: Em validação

Is it OK to Use Airtime ?

BackgroundProcessing wants to connect to http://127.0.0.1/WTDLocal/login.php?login=pedro&pass=orde using airtime. This may result in charges.

Is it OK to use airtime?

Fig. 5. Emulated handheld/PDA


+5550000 - DefaultColorPhone

Sensor Virtual Bluetooth
No. Handlers: 1
Received Messages:
(first) 12. (#21:23:20#WirelessToolkit: kjk)
13. (#21:23:23#WirelessToolkit: kjk)
14. (#21:23:27#WirelessToolkit: kjk)
15. (#21:23:30#WirelessToolkit: kjk)

Fig. 6. Emulated ECG sensor

3.3 Algorithm

The algorithm that pre-processes heart rate data employs the concept of heart rate effort [15]. This effort was determined by dividing the instantaneous heart rate by the estimated maximum upper limit (see table 1). At the Portuguese Foundation of Cardiology (PFC) web page can see that rates above 75% of maximum heart rate frequency can be considered excessive for untrained subjects and that the reference heart rate, during exercise, is between 60% and 75% of maximum heart rate. Maximum frequency depends on age and can be defined, though not the most accurate formula, as the constant 220 minus age in years (see table 1).

Age	Reasonable effort heart rate Limits	Maximum heart rate
20	120-150/min	200/min
25	117-146/min	195/min
30	114-142/min	190/min
35	111-138/min	185/min
40	108-135/min	180/min
45	105-131/min	175/min
50	102-127/min	170/min
55	99-123/min	165/min
60	96-120/min	160/min
65	93-116/min	155/min
70	90-113/min	150/min

Table 1. Reference heart rate by age [15]

The solution found to analyze ECG was implemented on artificial neural network with a preprocessing algorithm for cardiac signal. The ANN has 10 neurons in the first layer, 7 in the hidden layer and 1 in the output layer. The activation function is the standard sigmoid (logistic). The handheld system and software must load all artificial neural network parameters, previously defined, stored in a database. It also records, throughout the day, cardiac signal values. This behavior keeps the system up to date in what comes to the natural and evolving state of the subject which can vary throughout life. The implemented algorithm relies on a sliding window. This approach helps to overcome variability in very close readings. Still it is considered as real time architecture despite physiological signal is read in a constant discrete-time. Any computer system is necessarily discrete. The data received from the virtual ECG sensor, which simulates the physiological signal sensor, running on Sun's Java wireless toolkit emulator, is integrated in a sliding window of size 10. The window contains the last 10 numerical values of heart rate. The determination of the emotional state of fear is realized by artificial neural network starting from the suppositions and knowledge on psychophysiological reaction fight-or-flight already stated.

Some numerical constants are found in the algorithm. They are an empirical approach and should be subject of further study. The use of sliding windows dims the probability of occurrence for false positives because singularities, errors or short duration startles, shall have diluted effect.

The preprocessing algorithm acts under the following specifications:

- When a user logs on successfully 10 units of heart rate will fill the "sliding window" of attention;
  - While the window is being filled will calculate:
    - Average (current in window values and previous homologous work period average);
    - Standard deviation (in window values);
  - The first value available in the “sliding window” is the homologous period average stored in the database.
- After filling the window the assessments for emotional state can be held calculating:
  - Window Average;
  - Window Standard deviation (Std);
  - Std Jitter (numerical difference between current calculated Std and the previous Std);
  - Window average jitter
    - numerical difference between current calculated window average and the previous;
  - Instantaneous Effort rate (Rest)
    - instantaneous value obtained from the sensor divided by the average of recording at rest condition;
  - Instantaneous Effort rate (working)
    - instantaneous value obtained from the sensor and divided by the average of recording at working condition;
  - numerical difference between the instantaneous value and the average recorded for the same period (dif1);

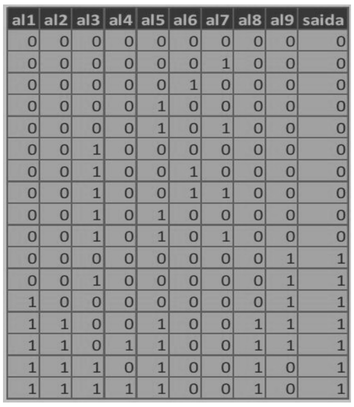
- Variation of the previous value (Dif1) to Std of the same work period (Dif1/desvpPerHomol) -> (VAR1);
- Assess the rate of stress to the average at rest
  - ○ If equal or greater than 1.4 times the average value obtained at rest then it creates an alarm (AL1 = 1);
- Assess the rate of stress compared to the average at work
  - ○ If equal or greater than 1.25 times the average value obtained at work then it creates an alarm (AL2 = 1);
- Evaluate the difference between instantaneous value (from the sensor) and the result of multiplying 2.8 by the lowest of values (standard deviation recorded for the same period or the numerical constant 17)
  - ○ If greater than 0 then create alarm (AL4=1)
- Verify if windows average has risen
  - ○ If affirmative create alarm (AL5=1);
- Assess whether the instantaneous value is lower than the average recorded at rest subtracted from the corresponding standard deviation at rest
  - ○ If true then create alarm (AL6=1);
  - ○ No one should have a working heart rate lower than the resting heart rate subtracted of the Std at rest.

The previous protocol allows the creation of different alarms (AL1 to AL6) which will be integrated into an alarm sliding window with 10 slots. This is the “stress” window. Alarms 7 and 8 will be determined according to following rules:

— Determine the average window of alarm AL6
  - Higher than 0.5 indicates low heart rate or disconnection without logout (AL7=1);

— Determine the average window alarms [AL1 to AL5]
  - Higher than 0.5 indicates shortage of data or actual stress tension (AL8=1);

All values AL1 to AL9 are fed to the ANN. The following table, table 2, presents the all conditions under which a “stress alarm” will be generated (saida=1).



al1	al2	al3	al4	al5	al6	al7	al8	al9	saida
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0
0	0	1	0	0	1	1	0	0	0
0	0	1	0	1	0	0	0	0	0
0	0	1	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	1	1
0	0	1	0	0	0	0	0	1	1
1	0	0	0	0	0	0	0	1	1
1	1	0	0	1	0	0	1	1	1
1	1	0	1	1	0	0	1	1	1
1	1	1	0	1	0	0	1	0	1
1	1	1	1	1	0	0	1	0	1

Table 2. ANN input/output matrix. Inputs:(AL1 – AL9); output (saida)

4 Results

Regarding the outcome of the created software, claims can be made that the preprocessing algorithm performs the appropriate processing, in accordance with the programmed rules. The artificial neural network performs calculations, without error, in accordance with assigned values for each input connections and for each neuron. The activation function, equal for all active artificial neurons, worked as expected and results are traceable. The preprocessing algorithm reduces all integers or real values to Boolean which, in turn are fed to the ANN. Alarms are generated and recorded in the database through requests made to the local Web server. The virtual sensor allows variation of the value placed on the output through emulator's virtual keyboard.

Coherent classification, of fight-or-flight response, was observed in PDA software. With undetermined efficiency, the PDA software algorithm and ANN classification method mapped ECG values to the corresponding and expected emotion. Whenever ECG value was consistently high, with a heart rate effort equal to or greater than 60% of maximum heart rate - recorded in the database for each employee - a stress alarm is generated. Similarly if the heart rate remains consistently below the resting ECG value, subtracted of the standard deviation, a "disconnected alarm" is generated. For this purpose a very low value of 10 beats per second was used.

5 Conclusions

The proposal of using off-the-shelf technology to enhance security of people was found feasible. There is available and mature enough technology to transmit, process and record physiological signals as described in above sections. Moreover, the tests and technical grounds used, within this approach, suggest that using minimally or non-invasive wireless sensors in conjunction with mobile devices to collect, analyze and record physiological data, in regular activity in the workplace, is both possible and desirable. The wireless solution eliminates mobility constraints and will be worthy for scenarios where one or more employees are exposed, to some extent, to security threats. Such scenarios include Taxi drivers, money transport trucks, flight stewardesses and any public access offices like jewellery shops, gas stations and others.

The presented algorithm has the advantage of analyzing values, read by sensor, in relating them to values recorded for each subject. It presents, in theory, high interpersonal stability. The preprocessing algorithm presents fair to good user undependability. Each subject, after being submitted to ECG studying at rest and at regular work, has a recorded base line (mean and standard deviation for ECG).

Of the examples, presented in background and previous work section, it is credible that all the necessary technologies are mature enough to produce a technological solution for personal security. Nevertheless false positives, in fight-or-flight classification, may occur out of classification error, ANN training deficiencies, or unexpected but not threatening scenarios.

6 Future work

Although the algorithm looks promising in dimming false positive occurrence, on account of sliding window, testing it with real equipment is essential to validate the system. Experimenting in controlled and uncontrolled environments to gather data and cross reference will be required.

Validation of empirical values used is also necessary. Exploitation of new equipment, like VitalJacket[2], may prove worthy despite it is a medical graded equipment and, therefore, more expensive. Sliding window size is also empirical and, therefore, should be submitted to further study.

The algorithm should be tested with SVM, method referred as high in efficiency and accuracy, with and without pre-processing.

Integration of more physiological signals, galvanic skin response, respiration rate and volume, could prove worthy for classification accuracy purposes. Current open source Android devices and Arduino, an open source hardware platform, will enable multiple sensor integration without mobility loss.

References

1. Barbosa, F.: Psicofisiologia. Retrieved 07 16, 2009, from Psicofisiologia: http://www.fpce.up.pt/psicof/aulas_praticas/modulo_2.pdf
2. Berne, M. R., & Levy, N. M.: Physiology - Third Edition. St. Louis - Missouri: Mosby (1993)
3. Cárdenas, A. F., Pon, R. K., Cameron, R. B., & A.Coyle, M.: The Mobile Patient and the Mobile Physician Data Access and Transmission. University of California at Los Angeles (2005)
4. Castagnetto, J., Rawat, H., Schumann, S., Scollo, C., & Veliath, D.: Professional Php Programming. Wrox Press Ltd. (1999)
5. Cortez, P., & Neves, J.: Redes Neuronais Artificiais. Braga (2000)
6. Davison, A. (2005, Setembro 2). Chapter B1 Echo Bluetooth. Retrieved Março 12, 2009, from Killer Game Programming in Java: http://fivedots.coe.psu.ac.th/~ad/index.html
7. Deitel, H., & Deitel, P.: Java Como Programar, 4.ª edição. Bookman (2003).
8. Dias, A. J.: Computação Afectiva - implementação de um wearable multimédia. Porto: Faculdade de Engenharia da Universidade do Porto (2005).
9. DONALD OKEN, HELEN A. HEATH.: The Law of Initial Values: Some Further Considerations. PSYCHOSOMATIC MEDICINE (2005).
10. Enderle, J. D., Blanchard, S. M., & Bronzino, J. D.: INTRODUCTION TO BIOMEDICAL ENGINEERING. Elsevier Academic Press (2005).
11. Ferreira Marques, F.A.; Ribeiro, D.M.D.; Colunas, M.F.M.; Silva Cunha, J.P.: "A real time, wearable ECG and blood pressure monitoring system", 6th Iberian Conference on Information Systems and Technologies (CISTI) (2011)
12. Fonseca, M. T.: Aquisição Automática de Dados em Unidades de Cuidados Intensivos. Porto: Faculdade de Engenharia da Universidade do Porto (2006)
13. Fonseca, T., Ribeiro, C., & Granja, C.: Vital Signs in Intensive Care: Automatic Acquisition and Consolidation into Electronic. *Journal of Medical Systems , 33* (1) (2009).
14. Fultz, P., & Hollinger, R.: Medical Astronaut Monitoring System (MAMS), (2001)

[2] Information obtained from Biodevices website, http://www.biodevices.pt, visited 02-01-2012

15. Fundação Portuguesa de Cardiologia. (n.d.). *Exercício físico e saúde cardiovascula*. (Fundação Portuguesa de Cardiologia) Retrieved 07 16, 2009, from Fundação Portuguesa de Cardiologia: http://cardiologia.browser.pt/PrimeiraPagina.aspx?ID_Conteudo=52
16. Gonçalves, H. M.: Caracterização da Variabilidade da Frequência Cardíaca através de Métodos Lineares e não Lineares. Porto (2004)
17. Healey, J. A., & Picard, R.: Wearable and Automotive Systrems for Affect Recognition from Physiology (2000)
18. KREIBIG, S. D., WILHELM, F. H., ROTH, W. T., & GROSS, J. J.: Psychophysiology. (44) (2007)
19. Krishnamoorthy, C., & Rajeev, S.: Artificial Intelligence and Expert Systems for Engineers. CRC Press, CRC Press LLC (1996)
20. Montgomery, K., Mundt, C., Thonier, G., Tellier, A., Barker, U. U., Ricks, R., et al.: Lifeguard- A Personal Physiological Monitor For Extreme Environments. Stanford University (2004)
21. MUCHOW, J. W.: *CoreJ2ME: Tecnologia e MIDP*. São Paulo: Makron Books (2004)
22. Li, S., & Knudsen, J.: Beginning J2ME: From Novice to Professional, Third Edition. Berkeley, CA: Apress (2005)
23. Lessa, J.: Fato para Computação Afectiva: protótipo de interface revelador de uma identidade. Faculdade de Engenharia da Universidade do Porto (2005)
24. Damásio, A. R.: O ERRO DE DESCARTES - Emoção, razão e o cérebro humano. EDITORA SCHWARCZ LTDA (1998).

Satisfaction in Meetings based on Group Decision Support Systems

João Carneiro[a]

[a]MAP-i – MAP Doctoral Program in Computer Science

joao.carneiro@fe.up.pt

Abstract. Group Decision Support Systems (GDSS) have been strongly developed and studied. More data have been increasingly added to these systems in order to be more assertive. It is considered that with certain data it is possible to the system to assess automatically the participants' satisfaction regarding the decisions made. Here, is proposed a methodology that allows a GDSS to measure the participant's satisfaction with the decision, considering aspects such as personality, emotions and expectations. The developed methodology was implemented in a GDSS with a multi-agent architecture consisting of agents that reflect the real participants and are modeled according to their personality. In this paper we show that the proposed model assesses correctly the participants' satisfaction.

Keywords: Decision Satisfaction, Group Decision Support Systems, Outcomes, Affective Computing

1 Introduction

One of the main factors to the success of an organization is the quality of the decisions that are made. However, when someone is questioned about the quality of a decision, the answer does not reflect only the outcomes evaluation, but also, even unconsciously; it includes the process evaluation necessary to reach the decision. This insight concerns how the goodness of a decision depends not only on its relation to ends or outcomes but also on whether the means used to make it were suitable [1][2]. Thus, one should be given prominence to the process, when drawing conclusions about the results.

Nowadays the decisions made by managers and executives are mostly performed in groups. Thereby, group decision-making is a process in which a group of people, called participants, act collectively analyzing a set of variables, considering and evaluating the available alternatives in order to select one or more solutions. The number of participants involved in the process is variable and all of them may be either at the same place at the same time or geographically dispersed at different times [3]. It is a known fact that the amount of hours a decision-maker spends in a meeting is not mostly used to make decisions. The time spent on things like social issues, are responsible for consuming the majority of the time of a process [4][5][6].

Aiming to satisfy all these requirements, Group Decision Support Systems (GDSS) adapts and develops over the time, incorporating new features, and modifying their architectures. Due to the costs to create conditions that allow participants to meet in the same place at the same time (time, travel, etc.) the ubiquitous GDSS appeared allowing decision-makers to contribute with their ideas to the decision process anywhere, anytime [7]. In addition to these available capacities, the architectures of GDSS have also suffered big changes, particularly in the use of architectures for multi-agent systems [8][9][10].

The affective computing is a recent research area which is included in the Artificial Intelligence (AI). Liu Jianming and Wang Jincai defined it as “the computing related to, which comes from or deliberately influences emotions” [14]. The affective computing allows extending the existing interaction between the human being and the computer through the inclusion of emotional communication, and of the necessary means to understand that information which derives from affectivity [15]. The most recent studies indicate that emotions play an essential role in the decision-making, perception, learning and in a diversity of other cognitive functions.

There are several studies about satisfaction, the majority of these studies are related to the participant satisfaction using GDSS or other systems and technologies [11][12][13]. The analysis of the satisfaction of a meeting participant on the decision-making process needs a rather more complex analysis, on which there is not much literature.

The goal of this paper is to analyze the participants’ satisfaction with the decision, which is mostly done through the system with minimal time-spending cost to the participant.

2 Decision Satisfaction

The satisfaction with a decision resulting from a decision process is something that needs a complex analysis that involves multiple variables. Obviously the satisfaction is related to what we think a good decision is. But what is a good decision? The classic answer to what makes a decision good concerns outcomes. A good decision has high outcome benefits (it is worthwhile) and low outcome costs (it is worth it) [16]. Higgins says that psychologically, then, a decision is perceived as good when its expected value or utility of outcomes is judged to be more beneficial than the alternatives. The benefits include the social benefits of a decision, such as those received from a “politically correct” or ingratiating decision. The costs of attaining the outcomes can also influence whether a decision is perceived as good. The outcome benefits have to be weighed against the costs of attaining the outcomes. The costs include not only the goods or services one must give in exchange for receiving the benefits but also the costs of the decision-making process itself. The decision-making process that would optimize outcomes might not be used because the costs in cognitive effort or time are too high [17][18].

Therefore it is clear that there is much more than knowing whether the chosen alternative was participants’ favorite in order to evaluate their satisfaction with the decision. It was suggested that a purely cognitive approach may be inadequate in modeling satisfaction ratings, so it is particularly important to include emotional variables [19][20][21]. The research that has been made in the field of satisfaction has recognized that there is a need to incorporate the emotional and affective components in regulating consumer's satisfaction [22].

Therefore, it is not only the final results or the decisions made that determine the quality and the satisfaction of the decision. This insight concerns how the goodness of a decision depends not only on its relation to ends or outcomes but also on whether the means used to make it were suitable [23][24]. Suitability here refers only to what is morally proper. By considering the more general meaning of suitable as "fit", a new perspective on what makes decisions good is possible [16].

3 Proposed Model

Bailey and Pearson agreed that satisfaction in a given situation is the sum of one's feelings or attitudes toward a variety of factors affecting that situation [25]. Our goal is to manipulate certain data, which at the end allows the system itself to evaluate the status of the participants satisfaction with the decision. Therefore to analyze the participants satisfaction with the decision it is important to consider the chosen alternative, the expectation related to the decision and to the process, the personality as well as the emotional changes.

3.1 Point 1 – Satisfaction concerning the chosen alternative

At this point the quality of the chosen alternative is considered from the perspective of the participant, and for that it is important to know what is considered by the participants as a good alternative. As said before, according to the literature the perception of the decisions' quality is related to the advantages that the participant identifies in that alternative, comparing it to the others. Thus, whereas the preferred alternative is the best in the participants' perspective, the distance between the preferred alternative and the chosen one means a loss of participants' satisfaction regarding the decision. The loss of satisfaction comprises the difference in the assessment made by the participant for each of the alternatives, as well as what the participant did not achieve with the final decision. The participants' assessment of each alternative varies over a period of [0, 1], where 0 means "I do not like nothing" and 1 means "I like very much". There are five different scenarios that may occur in a meeting affecting the satisfaction differently (see Table 1).

Table 1. Different Scenarios in a Meeting that Affect the Satisfaction

Scenario	Initial Preferred Alternative	Preferred Alternative Changed	Chosen Alternative
1	A	-	A
2	A	-	B
3	A	B	A
4	A	B	B
5	A	B	C

Thus, the result of a Point 1 is the reference of the satisfaction, while others will have an impact (positive or negative) on this point. The impact that each point has on Point 1 leads to a value affected by the occurrences of the other points, hoping that this becomes an approximation of the real value.

In order to understand the satisfaction taking into account these different scenarios, we suggest the following formula:

$$D_{Satisfaction} = \begin{cases} \frac{2Alt_F - Alt_P}{2},\ N = 0 \\ \frac{2Alt_F - Alt_P + \frac{\sum_{i=1}^{N}(Alt_{F_{i+1}} - Alt_{F_i})}{N}}{2}, N \neq 0 \end{cases} \tag{1}$$

Where:

- Alt_F = Appreciation of the final alternative, alternative chosen in the meeting;
- Alt_P = Appreciation of the participants' preferred alternative;
- N = Number of changes in alternatives' appreciation made by the participant;
- $Alt_{F_{i+1}}$ = The participants' appreciation of the chosen alternative at the stage $i + 1$;
- Alt_{F_i} = The participants' appreciation of the chosen alternative at the stage i.

3.2 Point 2 – Participants' expectations according to the decision and process

Consciously or not, people create expectations on (almost) everything. The relationship between expectations and the satisfaction is rather obvious. According to assimilation theory [26], consumers experience a psychological conflict if they perceive a discrepancy between their expectations and their perception of the consumption experience [27].

We conclude that it is important to know the participants' expectations according to some issues, in order to have a more accurate perception of the satisfaction. We think it is important to study the participants′ expectations on the following topics:

1. Complexity of the meeting: The participant should be questioned about how he thinks the meeting will be held, in order to reflect on whether he thinks it will have many conflicts and if the understanding among the participants will be problematic. And so the following question can be asked: Will this meeting be problematic?
2. Probability of the participants' preferred alternative to be chosen: Understanding the expectations regarding the probability of the participant's preferred alternative to be chosen.

There are three different types of satisfaction's impact for each suggested topic:

- Positive Impact: When the final results exceed the expectations.
- Negative Impact: When the expectations are higher than the results achieved.
- Without Impact: When the expectations are achieved.

In order for the participant to classify the expectations, it is suggested to use the Visual Analogue Scale (VAS). A Visual Analogue Scale is a measurement instrument that tries to measure a characteristic or attitude that is believed to range across a continuum of values and cannot easily be directly measured. For example, the amount of pain that a patient feels ranges across a continuum from none to an extreme amount of pain. The expected value will be between [0, 1]. The way expectations are taken into account in the analysis will be explained onwards.

As stated in the model the impact of expectations regarding the preferred alternative operates on the result of Point 1. To calculate this impact it was applied one formula for the case of positive impact and other one for the negative impact.

Positive Impact

This type of impact occurs when the chosen alternative is the preferred of the participant agent, immediately if exists or not, the impact of the expectation will be positive. The following formula is used to calculate the impact:

$$impact = (1 - E) * AP \tag{2}$$

Where:

- E = Expectation that the participant has about his preferred alternative to win;
- AP = Appreciation made by the participant of his preferred alternative.

After the impact calculus the result of Point 1 can be recalculate given the expectation by the following formula:

$$SatAltDecCExp = P1 * (1 - |P1|) * impact \tag{3}$$

Where:

- $SatAltDecCExp$ = satisfaction with the alternative chosen by the group, taking into account the expectation created;
- $P1$ (Point 1) = result of the satisfaction with the alternative chosen by the group;
- $impact$ = result of the calculation of the positive impact.

With this last formula is obtained then the satisfaction of the participant for the chosen option, but considering its expectation regarding this alternative be the chosen one.

Negative Impact

This type of impact occurs when the alternative is not the preferred choice of the participant, so if exists or not, the impact that the expectation will cause will be a negative impact. The following formula is used to calculate the impact.

$$impact = (AP - AE) * E \tag{4}$$

Where:

- AP = Appreciation made by the participant of his preferred alternative.
- AE = Appreciation made by the participant of the alternative chosen.
- E = Expectation that the participant has about his preferred alternative to win;

The calculus of the negative impact is different from the one on the positive impact. This is due to the fact that this case is related to the impact the participant has a certain expectation that has not been met and that symbolizes a loss. This loss is in turn related to the difference between the evaluation of its preferred alternative and the one elected by the group.

After the impact calculus, the result of Point 1 can be recalculated given the expectation by the following formula:

$$SatAltDecCExp = P1 * ((1 - |P1|) * (-1)) * impact \quad (5)$$

Where:

- $SatAltDecCExp$ = satisfaction with the alternative chosen by the group, taking into account the expectation created;
- $P1$ (Point 1) = result of the satisfaction with the alternative chosen by the group;
- $impact$ = result of the calculation of the negative impact;

For the case of negative impact it means that the preferred alternative of the participant was not the chosen one, the formula to calculate Point 1 taking into account the expectation its a bit different. In this case the component that has a negative impact is the difference between the expectation and zero. This difference reflects the belief that the participant had of his preferred alternative victory that didn't happen and would have a negative impact on the satisfaction.

3.3 Point 3 – Factor concerning the Personality

The personality is a concept that can’t be briefly defined, because it has a different meaning according to some psychologists who study it. Although most of them would agree that the field of personality is the study of how individuals differ from each other, psychologists would differ about the best way to conceptualize these types of differences [28]. The fact that people differ in their ideas and attitudes, makes them react differently to the factors they are exposed to. Lately, the satisfaction with most of the scenarios of a persons’ personality is being studied. For instance, Shiammack et al. [29] conducted a study on two factors of The Big Five: the Neuroticism and the Extraversion that contribute to life satisfaction. Another was conducted by Timothy et al. [30], where they tried to establish a correlation between the values of each type of personality of The Big Five and Job satisfaction.

Knowing that the personality of each one of us influences satisfaction, we think it is relevant to take into account the personality on this analytical model of satisfaction. At this point, we can’t do any kind of considerations on how each personality type lives the satisfaction in this context. Anyway, this point remains open because we find it relevant.

3.4 Point 4 – Emotional Changes

Knowing the importance of the decision-making process and to make conclusions about the participants’ satisfaction regarding decision-making, it is necessary to understand what happens during the process. As mentioned before, it is important to include in the analysis of satisfaction affective and emotional components [19][20][21][22].

Having said this, we want to include, at this point, the analysis of emotions generated and to know how they can change the participants’ mood. There are two important points to be studied:

- The sum of emotional spaces that exceed positively or negatively the normal state: it is thus possible to measure the emotional cost that the meeting had to the participant;
- Participant mood at the end of the meeting.

Emotional changes.

Let $IntEmoPGlogal$ be the intensity of positive emotions:

$$IntEmoPGlogal = \frac{\sqrt{(P)^2+(A)^2+(D)^2}}{\sqrt{3}} \tag{6}$$

Let $IntEmoNGlogal$ be the intensity of positive emotions:

$$IntEmoNGlogal = \frac{\sqrt{(P)^2+(A)^2+(D)^2}}{\sqrt{3}} \tag{7}$$

After calculating the intensities, it will be calculate d the emotional cost of the meeting:

If IntEmoPGlogal = IntEmoNGlogal Then Cost=neutral
If IntEmoPGloga > IntEmoNGlogal Then Cost=positive
If IntEmoPGlogal < IntEmoNGlogal Then Cost=negative

Participant mood. The mood intensity in each moment is calculated according to the formulas presented in last section.

Is also relevant to calculate the impact of the expectative that the participant had, for the meeting goes well (not conflicting). To calculate that impact we follow an approach similar to the one described in next section.

Impact of expectations in mood.

Positive Impact: The positive impact happens when the meeting has a positive Cost.

$$impact = (1 - E) * H \tag{8}$$

Where:

- E = expectation of the process goes well;
- H = difference of intensity between initial and final mood.

The value of the impact is to update the value of mood based on initial expectations.

$$HumFinCExp = H + (1 - |H|) * impact \tag{9}$$

Where:

- $HumFinCExp$ = mood value after considering initial expectations;
- H = difference of intensity between initial and final mood.

Negative impact: The positive impact happens when the meeting has a positive Cost.

$$impact = H * E \tag{10}$$

Where:

- E = expectation of the process goes well;
- H = difference of intensity between initial and final mood.

The value of the impact is to update the value of mood based on initial expectations.

$$HumFinCExp = (H + (1 - H) * impact) * -1 \tag{11}$$

Where:

- $HumFinCExp$ = mood value after considering initial expectations;
- H = difference of intensity between initial and final mood.

Neutral impact: The neutral impact happens when the intensities of initial and final mood are equal.

Final satisfaction calculation. After getting the value of satisfaction with the alternative chosen by the group (based on participant preferred alternative) and humor given the expectation of the process goes is necessary to reconcile them. So, it is now necessary to calculate the impact that humor has on satisfaction that the participant has on the alternative chosen, considering their initial expectations.

$$SatFinal = SatAltDecCExp + (1 - |SatAltDecCExp|) * HumFinCExp \tag{12}$$

Where:

- $SatAltDecCExp$ = satisfaction with the alternative chosen by the group, taking into account the expectation created;
- $HumFinCExp$ = mood value after considering initial expectations.
- $SatFinal$ = reflects the participant agent satisfaction with the group decision.

4 Measuring the Result of the Satisfaction Using the Model

To measure the output of each of the points of the model we must define how they are related. It is considered that the first issue of Point 2 (complexity of the meeting) is strongly related to Point 4 (emotional disorders), while the second issue of Point 2 (probability of the participant's preferred alternative is chosen) is strongly related to Point 1 (satisfaction with the alternative chosen by the group). So the Point 2 (expectations) will not work isolated, but it will influence the results of the other two points.

The expectations will change the values for Point 1 and Point 4 through a particular impact. The impact causes an expectation that is obviously not always the same. Even knowing the impact that causes the expectation is positive, negative or neutral, it is necessary to quantify that impact.

Beyond expectations, Points 3 and 4 will also have an impact on Point 1. This is because it is considered that the satisfaction about something always gets related to the evaluation that is made (the choice of the service, product, etc.). After this evaluation there are other factors such as expectations and the process to change satisfaction. Thus, the Point 1 will be in this case the analysis that is performed by a human being, while the other points according to the context will affect or not (positively or negatively) the satisfaction. The end result of satisfaction will belong to the interval [-1, 1].

The following scale of satisfaction, developed and inspired from the work of Babin and Griffin [31], reflects the satisfaction of the final result obtained by the formula here introduced (see Table 2).

Table 2. Scale of Satisfaction

Designation	Interval
Extremely Dissatisfied	[-1 ; -0,75]
Very Much Dissatisfied	]-0,75 ; 0,5]
Dissatisfied	]-0,5 ; 0,25]
Some Dissatisfaction	]-0,25 ; 0[
Some Satisfaction	[0 ; 0,25[
Satisfaction	[0,25 ; 0,5[
Very Much Satisfaction	[0,5 ; 0,75[
Extremely Satisfied	[0,75 ; 1]

5 Experiments and Discussion

In order to validate the presented model, 6 experiments were conducted; each one of them consists of a large number of simulations. Due to the big number of experiments and simulations, in this topic the experiments are presented, followed by the interpreting of the results. To validate the model was conducted its implementation in the prototype developed in the ArgEmotionsAgents project and the case study presented in [28] was used. The case study described here has the goal of inferring the satisfaction regarding the alternative chosen by the participants of the meeting. In the illustrated problem of the case study, the participants had to choose the best renewable source of energy to be installed in a certain place.

Table 3 presents the data modeled by the participants in their agent regarding the taste for the preferred alternative and for the chosen alternative. This Table also presents the type of personality of each participant which is obtained by the participants' answer in the questionnaire of the big five inventory and the participants' expectations regarding their preferred alternative to be chosen by the group and the process of the meeting occurring without conflicts.

Table 3. Modeling of the attributes

Agent	Personality	Favorite Alt.	Chosen Alt.	Alternative Exp.	Process Exp. Goes well
AgP1	Negotiator	0.67	0.2	0.05	0.7
AgP2	Avoider	0.65	0.65	0.2	0.4
AgP3	Submissive	0.91	0.85	0.7	0.6
AgP4	Aggressor	0.81	0.76	0.65	0.42

The first experiment was conducted with the goal of understanding, in a general way, the satisfaction results obtained, if they made sense and if they respected the theories found in the literature. In Table 4 are presented the data related to the emotions generated by the participant throughout the meeting process, as well as the gain or loss of mood at the end of meeting, comparing to the initial one. We can see that, for instance, AgP1 and AgP4 had a positive emotional cost, which lead to a gain of mood; AgP2 had a neutral emotional cost, not changing its initial mood, and AgP3 had a negative emotional cost which led to a loss of mood. In Table 4 are presented the results of the satisfaction obtained by each participant at the end of the meeting process.

As we can see, the results are related to the appreciation made by the participants to the alternative. But if we analyze with attention we can see that, for example, AgP3 had considerably less satisfaction than AgP4, when the difference of the appreciation made by the 2 participants for the chosen and preferred alternatives are similar, like the classification of the expectations. This is due to the big difference of the existing emotional cost between the two participants. As AgP3 had a negative meeting according to the emotional context, AgP4 had a positive meeting, which means that final satisfactions are different.

Table 4. Participant's Emotions and Mood

Agent	Emotions	Gain/Loss of Humor	Satisfaction	Designation
AgP1	Joy, Distress, Distress, Joy, Joy, Joy	0.1624	-0.066	Some Dissatisfaction
AgP2	-	0	0.684	Very Much Satisfaction
AgP3	Distress, Distress	-0.0152	0.373	Satisfaction
AgP4	Joy, Distress, Joy	0.31143	0.534	Very Much Satisfaction

A set of more specific experiments was conducted in order to understand the model's behavior in a big number of scenarios. The next experiment is related to the appreciation made to the preferred alternative with the expectations. This experiment only considers the case in which the participant's preferred alternative is chosen by the group. Figure 1 shows the results obtained. In the y axis is the final satisfaction, in the x axis is the expectation and on the right the result obtained from the appreciation made to the preferred alternative, used in point 1 of the model.

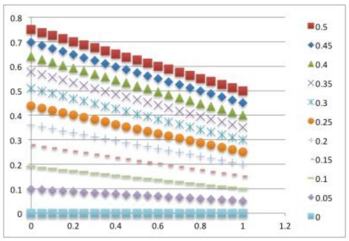


Fig. 1. The impact made by alternative appreciation and expectations in final satisfaction

The 231 simulations conducted in this experiment allow us to clearly understand that the more meaning the alternative has for the participant and the less is the expectation for this to be chosen by the group, in case it is chosen the bigger the impact, leading to a bigger satisfaction. With this experiment we can see that the ideologies found in the literature, related to the appreciation, expectation and satisfaction are proved.

In Figure 2 (left) are presented the results for the satisfaction obtained when the participant changes the classification of his/her favorite alternative and when it is chosen by the group. The y axis represents the participant's satisfaction at the end of the meeting; the x axis represents the classification the participant started to give to the alternative and on the right are the original values of the classification of the favorite alternative before the change. As one can see in this experiment, 110 simulations were conducted. One can see that the bigger the assessment, the bigger the satisfaction of the participant at the end of meeting; however it can be seen that as the assessment is higher, the distance between the satisfaction results of the several points from the initial assessments lowers, approaching the peak. This shows that the bigger the change in the assessment made, the bigger the positive impact if the alternative is chosen by the group.

In Figure 2 (right) are presented the results obtained in the experiment and we want to understand the impact of the mood in the final satisfaction, considering that the expectation on the process goes well. In the y axis there is the final value of the satisfaction; in the x axis there are the simulated final values of the mood, and on the right there are the values which represent the expectation on the process to go well from the participant's point of view.

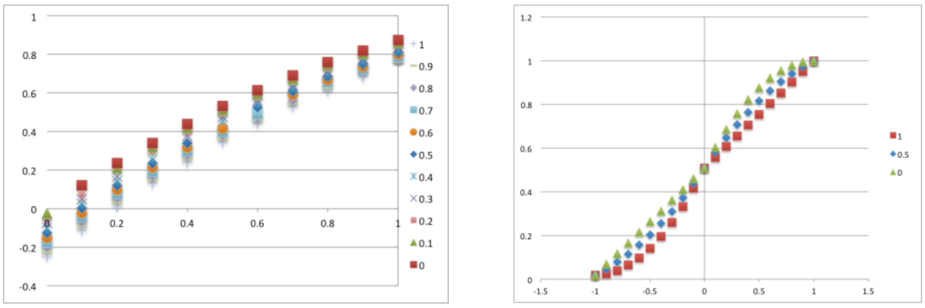


Fig. 2. Changing the alternative appreciation (left) and the impact of mood in final satisfaction (right)

63 simulations were conducted, which allow seeing that when the expectation for the process to go well is 0, and the mood had a gain, until the maximum satisfaction is reached, the impact of mood in satisfaction is always the bigger, on the other hand, when the mood had a loss, its impact is lower. The exact opposite happens when the expectation for the process to go well is 1. This experiment validates correctly the humor's influence and the impact of the expectation of that humor in the satisfaction.

6 Conclusions and Future Work

Several concepts of affective computing and decision making were presented in this paper. The concepts of satisfaction and the existing models which assess the satisfaction were also presented. Furthermore, it was proposed in this paper a new model which allows assessing automatically the satisfaction of participants of a meeting, which is supported by a group decision support system. We believe that the proposed model allows obtaining a large amount of useful and valuable information. A big number of experiments and simulations were conducted, which allowed to validate correctly every assumption created after reading the literature of different areas (psychology, computer science, economy and sociology).

For future work, we intend to conduct experiments in partnership with psychologists, in which we will conduct a case study with real people. With that work, we also intend to make the model more assertive, through the changes that might be done after studying the data obtained.

To the best of our knowledge, this is a first attempt to implement a model of satisfaction analysis completely automatic, which considers every point found as relevant in the literature.

References

[1] L. R. Beach, Image Theory: Decision Making in Personal and Organizational Contexts, New York: Wiley, 1990

[2] J.G. March, A Primer on Decision Making: How Decision Happen, New York: Free Press, 1994

[3] F. Luthans, Organizational Behavior. In 10th ed. McGraw Hill Irwin, Boston, 2005

[4] C. Argyris and D. Schon, Theory in Practice, Jossey-Bass, San Francisco, 1974

[5] L. R. Hoffman, Applying Experimental Research on Group Problem Solving to Organizations, Journal of Applied Behavioral Sciences, 1979, pp. 375-391

[6] H. Mintzberg, The Nature of Managerial Work, Harper and Row, New York, 1973

[7] J. Grudin, Group Dynamics and Ubiquitous Computing, 2002

[8] G. Marreiros, C. Ramos, and J. Neves, Modeling Group Decision Meeting Participants with an Agent-Based Approach, 2007

[9] T. Ito and T. Shintani, Persuasion Among Agents: An Approach to Implementing a Group Decision Support System Based on Multi-Agent Negotiation, 1997

[10] H. Zhang, H. Leung, K. Raikundalia, AOCD: A Multi-Agent Based Open Architecture for Decision Support Systems, 2005

[11] O. Briggs, G. Vreede and B. Reinig, A Theory and Measurement of Meeting Satisfaction, 2002

[12] X. Tian, W. Hou, K. Yuan, A Study on the Method of Satisfaction Measurement Based on Emotion Space, 2008

[13] P. Souren, S. Priya, K. Ramamurthy, User Satisfaction with System, Decision Process, and Outcome in GDSS Based Meeting: An Experimental Investigation, 2004

[14] L. Jianming and W. Jincai, The distance education system based on affective computing. China medical education technology, vol. 22, no. 1, 2008, pp. 67-69

[15] R. W. Picard, Affective Computing for HCI, 1999

[16] E. T. Higgins, Making a Good Decision: Value From Fit, 2000

[17] H. A. Simon, A Behavioral Model of Rational Choice, Quarterly Journal of Economics, 69, 1955, pp. 99-118

[18] H. A. Simon, Motivational and Emotional Controls of Cognition, Psychological Review, 74, 1967, pp. 29-39

[19] V. Liljander and T. Strandvik, Emotions in Service Satisfaction, International Journal of Service Industry Management, 8(2), 1997, pp. 148-169

[20] R. L. Oliver, R. T. Rust, S. Varki, Customer Delight: Foudations Findings, and Managerial Insight, Journal of Retailing, 73(3), 1997, pp. 311-336

[21] J. Wirtz and J. E. G. Bateson, Consumer Satisfaction with Services: Integrating the Environment Perspective in Services Marketing into the Traditional Disconfirmation Paradigm, Journal of Business Research, 44, 1999, pp. 55-66

[22] J. Wirtz, A. S. Mattila and R. L. P. Tan, The Moderating Role of Target-Arousal on the Impact of Affect on Satisfaction – An Examination in the Context of Service Experiences, Journal of Retailing, 76(3), 2000, pp. 347-365

[23] L. R. Beach, Image Theory: Decision Making in Personal and Organizational Contexts, New York: Wiley, 1990

[24] J. G. March, A Primer on Decision Making: How Decisions Happen, New York: Free Press, 1994

[25] J. E. Bailey, S. W. Pearson, Development of a Tool for Measuring and Analysing Computer User Satisfaction, Management Science (29:5), 1983, pp. 530-545

[26] M. Sherif, C. I. Hovland, Social Judgment: Assimilation and Contrast Effects in Communication and Attitude Change, New Have, CT: Yale University Express, 1961

[27] I. A. Bosque, S. M. Collado, The Role of Expectations in the Consumer Satisfaction Formation Process: Empirical Evidence in the Travel Agency Sector

[28] R. Santos, G. Marreiros, C. Ramos, J. Neves and J. Bulas-Cruz, Personality, Emotion, and Mood in Agent-Based Group Decision Making, Intelligent Systems, Vol. 26(6), 2011, pp. 58-66

[29] U. Shimmack, S. Oishi, R. M. Furr and D. C. Funder, Personality and Life Satisfaction: A Facet-Level Analysis, 2004

[30] A. Timothy, D. Heller, M. K. Mount, Five-Factor Model of Personality and Job Satisfaction, Journal of Applied Psychology, Vol. 87(3), 2002, pp. 530-541

[31] B. J. Babin and M. Griffin, The Nature of Satisfaction: An Updated Examination and Analysis, 1998

Author Index

PROCEEDINGS OF THE 7TH DOCTORAL SYMPOSIUM IN INFORMATICS ENGINEERING

ISBN 978-972-752-141-8

9 789727 521418

FEUP EDIÇÕES

www.fe.up.pt / **dsie12**