Studies in Computational Intelligence 591

Kohei Arai
Supriya Kapoor
Rahul Bhatia *Editors*

Intelligent Systems in Science and Information 2014

Springer

Studies in Computational Intelligence

Volume 591

Series editor

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

About this Series

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

More information about this series at http://www.springer.com/series/7092

Kohei Arai · Supriya Kapoor · Rahul Bhatia
Editors

Intelligent Systems in Science and Information 2014

Extended and Selected Results from the Science and Information Conference 2014

Springer

Editors
Kohei Arai
Graduate School of Science
and Engineering
Saga University
Saga
Japan

Rahul Bhatia
The Science and Information (SAI)
Organization
London
UK

Supriya Kapoor
The Science and Information (SAI)
Organization
London
UK

ISSN 1860-949X ISSN 1860-9503 (electronic)
Studies in Computational Intelligence
ISBN 978-3-319-14653-9 ISBN 978-3-319-14654-6 (eBook)
DOI 10.1007/978-3-319-14654-6

Library of Congress Control Number: 2014960230

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

The Science and Information (SAI) Organization (www.thesai.org) is an international professional organization dedicated to promoting research, technology, and development by providing multiple platforms for collaboration of professionals and researchers to share existing and generate new knowledge.

Each year, SAI is responsible for organizing its annual conference, Science and Information Conference. The conference brings together some of the most devoted minds in the industry to share ideas and showcase achievements. These conferences are a place for researchers to promote and defend their work based on its merits. In addition to giving researchers a place to present their findings and data, these conferences are also used to help increase public engagement with science and community. The keynote talks are published on YouTube for the general populous to view in order to gain understanding of some of the advancements that are taking place.

SAI Conference 2014 was an overwhelming success, attracting 190+ delegates, speakers, and sponsors from 50 countries and provided great intellectual and social interaction for the participants. The inspiring keynote speeches and the state-of-the-art lectures have deeply motivated attendees and envisioned future research directions.

The conference was held in London from August 27–29 in 2014. In this book, we have invited 25 of the main contributors to this conference in the domain of Intelligent Systems and Artificial Intelligence to expand on their presentations and represent their work as a book chapter. Selected papers were invited to extend and submit them for a complete new review process for consideration. The final decision for the inclusion in the special issue has been strictly based on the outcome of the review process. The objective of the special issue is to make available recent results and report in-progress research in the field of computational intelligence. We hope the reader interested in the field will find this book stimulating, leading to even more contributions to the next conference.

We take this opportunity to thank authors, presenters, participants, keynote speakers, session chairs, organizing committee members, student volunteers, program committee members, steering committee members, and people in other various roles for their very valuable activity performed with great dedication and competence, adapting to several innovative procedures and reacting swiftly.

Kohei Arai
Supriya Kapoor
Rahul Bhatia

Contents

Computer Input by Human Eyes Only and It's Applications

Kohei Arai

Abstract Computer input by human eyes only is proposed for disabled and elderly persons, in particular, together with its applications to communication aid, having meal aid, information collection aid, watching TV aid, listening to radio aid, phoning aid, Electric Wheel Chair: EWC control, service robot control. Through experiments with disabled and normal persons including Alzheimer diseased persons, it is found that the proposed Eye Based Human Computer Interaction: EBHCI which includes computer input by human eyes only can be used for disabled and normal persons for a variety of applications.

Keywords Human computer interaction · Computer input by human eyes only

1 Introduction

The number of disabled persons is getting large. There are 3.7 million of disabled persons in Japan. 0.3 million of disabled persons out of 3.7 million of disabled persons are blind or having problem on their eye(s). Therefore, 3.4 million of disabled persons have difficulties on their body and can use their eye. Also, the number of elderly persons reached to 31 million persons in Japan and is increased remarkably. Such persons have difficulties on communications, having meals, information collections, watching TV, listening to radio, phoning, moving to somewhere with their Electric Wheel Chair: EWC, sightseeing, etc. The proposed Eye Based Human Computer Interaction: EBHCI is intended to provide such capabilities for disabled persons as well as elderly persons by using their eyes only.

K. Arai (✉)
Graduate School of Science and Engineering, Saga University,
1 Honjo, Saga 840-8502, Japan
e-mail: arai@is.saga-u.ac.jp

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_1

Computer input by human eyes only has been proposed and developed. It allows users to key-in operation and mouse operation by their eyes only and also allows head movements. Conventional computer input by human eyes only has the following problems,

- Key-in success rate is poor,
- Influence due to illumination conditions,
- It requires a time consumable calibrations,
- Influence due to head movements, etc.

The computer input by human eyes only uses visible to camera to acquire human face. Eye is extracted from the acquired human face image. Then extract cornea and sclera. Meantime, feature points (two ends of eyes, two ends of mouth, two ends of eyebrows, for instance) which allows estimation of head pose are extracted. Therefore, geometric relations among feature points extracted from human face images are important. In order to determine the geometric relations, calibration is required. During the calibration process, users have to look at the four corners and the center of the computer screen. It takes a couple of minutes to acquire human face images when they are looking at the five points on the screen. Sometime, the calibration has to be tried again when the geometric relations are changed a lot. Using the location of cornea center in the image and the geometric relations, line of sight vector (at where users are looking) is estimated. Sometime, the geometric relations cannot be estimated due to changes of illumination conditions. There are influences due to shadow, shade, feature points missing due to occlusion, blocking cornea by eye muscular, and so on. Near Infrared: NIR cameras with NIR light source (Light Emission Diode: LED) is effective to overcome the influences. On the other hand, cornea curvatures are different by human by human. Therefore, the calibration is required to adjust the geometric relation variations due to cornea curvature difference. Using double Purkinje images, cornea curvature can be estimated. Thus time consumable calibration is no longer needed when cornea curvature is estimated. More importantly, key-in success rate of the conventional computer input by human eyes only has to be improved. One of the reasons for poor key-in success rate of the conventional computer input by human eyes only is estimated gaze instability. Due to computational errors, the estimated location at which users are looking on the computer screen is varied during the time when users are key-in. Therefore, it is hard to determine the designated keys when the distance between adjacent keys is shorter than variance of the gaze estimation results in poor key-in success rate. If the standard keyboard of key layout is used for computer input by human eyes only, the distance between adjacent keys is around 1.5 cm on the 16 in. computer screen. It is not so easy to stabilize the estimated gaze location within 1.5 cm. Therefore, some breakthrough is highly required. The proposed computer input by human eyes only introduced moving keyboard. Key layout can be simplified because keyboard can be moved to arbitrary locations. Therefore, all the keys on the keyboard can be selected and

determined by using enlarged five keys, left/right/top/bottom/center. The keyboard which is displayed in the back of the five keys moves to the left direction when user is looking at left key. That is the same thing for the other four keys. Thus the entire keys on the keyboard can be selected and determined by using the five enlarged keys for control of the keyboard movement. Five of enlarged keys can be selected and determined easily because these are displayed on the full computer screen with enough size. Thus key-in success rate becomes 100 %. Every 0.4 s, moving keyboard can be moved. Therefore, it takes less than 10 s even for 24 keys has to be travelled from the left to the right ends of the moving keyboard (keyboard consists of 24 by 6 keys) 1–22.

Computer input by human eyes only can be applied to a variety of fields,

- Mouse operations
- Electric Wheel Chair control
- Having meal aid
- Communication aid
- Information collection aid
- Service robot control
- e-learning system, etc.

It is possible to manipulate mouse by human eyes only. Only thing users have to do is just looking at pointer displayed on to computer screen. All the mouse operations, click, drug, drop, etc. can be done with human eyes only. Electric Wheel Chair: EWC can be controlled by human eyes only. EWC turn right when user looks at right direction while EWC goes forward when user looks at the center. Meanwhile, EWC turn left when user looks at the left direction. Robot arm equipped beside the patient bed can retrieve a food from the meal tray through controlling robot arm motion (the tip of the robot arm) by patient's eyes. Using the proposed moving keyboard, sentences can be created by human eyes only. Then the sentences can be read by using software tool for type to talk. Thus the user can talk with other persons by human eyes only. Also, user can make phoning, search information through web sites, watching TV, listening to radio, search e-book, e-learning contents, etc. by human eyes only. Lecture can monitor the locations at which student is looking on the computer screen through lessons with e-learning contents by human eyes only. Students' gaze locations can be monitored by lecturers when the student uses computer input by human eyes only. Evaluation of students' concentration to the e-learning contents at which lecturers would like students look can be done with computer input by human eyes only. Service robot can be controlled by human eyes only. Using image acquired with camera mounted on the service robot, patient lay down on bed can control service robot movement. Also, patient can create sentences with computer input by human eyes only. Therefore, patient can enjoy conversation with the other peoples through service robot and also can ask something to the other peoples. The paper proposes the computer input system by human eyes only together with the aforementioned applications of the system.

2 Proposed Eye Based Human Computer Interaction: EBHCI

2.1 Eye Based Human Computer Interaction: EBHCI with Moving Keyboard

In order to improve key-in success rate of the conventional computer input by human eyes only, Eye Based Human Computer Interaction: EBHCI with moving keyboard is proposed. Figure 1 shows how accuracy is affected by gaze instability and key size.

There are the parameters which are affecting EBHCI in regard to accuracy as (1) instability of gaze, (2) size of keys and number of keys, and (3) position of keys ("zero position" has maximum accuracy because it seems ideal for the eye). The gaze targeting the key could hit the wrong spot if gaze instability is larger than key size. Gaze instability could be caused by poor camera image resolutions, noise, and environmental disturbances such as external light source. As a result, the method for detecting eye performance has become somewhat unreliable. In addition to the aforementioned factors, distance between eye and target display have also played a part. Accuracy decreases with distance of subject from key and increases when subject gets closer. This phenomenon is a result of the flicker of the camera influencing the calculation of eye gaze, and accuracy is proportional to distance from the key. The relationship between instability and distance is depicted in Fig. 2.

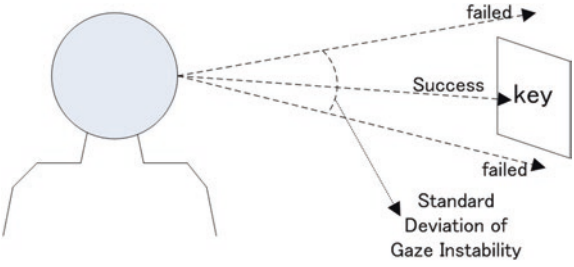


Fig. 1 Accuracy affected by gaze instability and key size

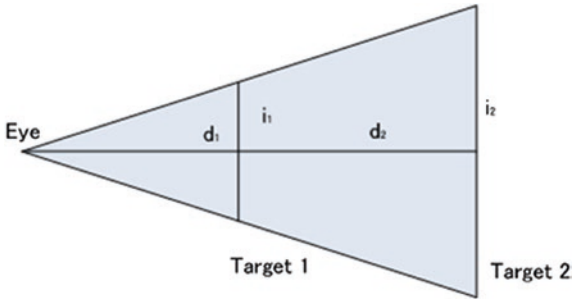


Fig. 2 Instability affected by distance of user-target

It shows that the instability is proportional with distance formatted as follow,

$$\frac{i_1}{i_2} = \frac{d_1}{d_2} \tag{1}$$

where i and d are instability and distance respectively. Although gaze instability could be minimized, making it become perfect is difficult due to external disturbances. If we assume that key size is S and the instability of gaze result is SD, the accuracy can be expressed as follows,

$$A \approx \frac{S}{SD} \times 100\,\%, \quad \text{if SD} \geq \text{S} \tag{2}$$

$$A \approx 100\,\%, \quad \text{if SD} < \text{S} \tag{3}$$

The two formulas above explain the condition when the instability is greater than or equal to key size. In such case, accuracy is proportional to key size and is inversely proportional to instability. If instability is lower than key size, accuracy could theoretically achieve 100 %. A real situation of such phenomenon is depicted in Fig. 3 which shows a condition when instability is greater than key size.

It shows the user selecting the "Q" character, yet the result has probability hit the area outside the "Q" range, hitting either "A" or "W" instead (Fig. 4).

The relation between the position key and distance between "zero position" and a key d can be formulated as inversely relation,

$$A \approx \frac{100\,\%}{d} \tag{4}$$

When the gaze direction is orthogonal with the target display, it means no eye movement has occurred, 100 % accuracy could be achieved if the camera could capture the best eye position. An ideal image of any deformed shape of eye

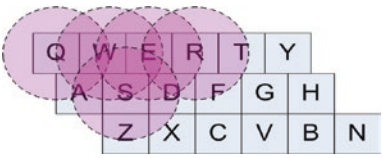


Fig. 3 The typing error caused by instability

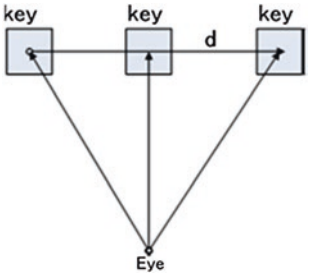


Fig. 4 Relation between eye and the keys aligned on the computer screen

could be captured. By using such eye image, the estimation of eye gaze and also blinking detection could be estimated with maximum accuracy. This happens because the components of the eye such as pupil, iris, two corners of the eye, sclera, and so on, clearly visible. Disturbance such as eyelashes covering half of pupil or an occluded eye caused by an eye moving to the side edge are not visible. On other hand, when the eye is moving to the edge of the eye, the accuracy becomes minimal because the disturbance affects both the gaze estimation and blinking detection. The accuracy decreases as result of the movement from eye to edge (the length from zero position until eye position).

In order to overcome such situation, an innovative keyboard for EBHCI by utilizing the moving keyboard is proposed. We propose the moving keyboard for improving the accuracy by maintaining relation between S, SD, and d (Size of Key, Standard Deviation of Instability, and key position) with ideal condition ($SD < S$ and $d = 0$). We improve the accuracy by increasing the key size without expanding of keyboard size. The 616 by 232 pixels keyboard size is used on 800 by 600 resolution of display, but with our innovation, we could increase the key size without changing of keyboard size.

The proposed keyboard consists of the moving keyboard (main layout) itself and five zoom-in transparent keys. The main layout is moveable, but the zoom-in transparent keys fixed. The main layout is moveable and controlled by the gaze. The Omni direction of main layout movement is controlled by user looking at the four zoom-in transparent keys surrounding the center point. If the user is looking at the key situated on left, then the main layout shifts to the right (opposite direction). The same thing also happens for right, up, and down. The function of transparent zoom-in keys is for helping users know the candidate key to be selected when controlling the keyboard. The transparent keys show the candidate of selected key.

2.2 Proposed Keyboard

In line with our explanations above, we improved accuracy by maintaining relation among S, SD, and d value into ideal condition requiring SD must be less than S and d equal to zero. It means that we could improve accuracy by increasing the key size (increase S) and keeping key position in the center (d equal to zero). A new keyboard is shown in Fig. 5. Basically, our keyboard is controlled by user looking at the five increased keys. When users want to select “A” character, users have to look at increased key on left.

As result of user looking at left, moving keyboard moves to right until users return looking at the center. Also, when users want to select “M” character, users have to look at increased key on right for three steps (assume the initial position is on G character), and continue to look at increased key on bottom for one step. As result of these, moving keyboard moves to left three steps and moves to upward one step.

The increasing key is only applied on the center key. We keep other keys for maintaining the keyboard size. The increased keys placed on center, left, right,

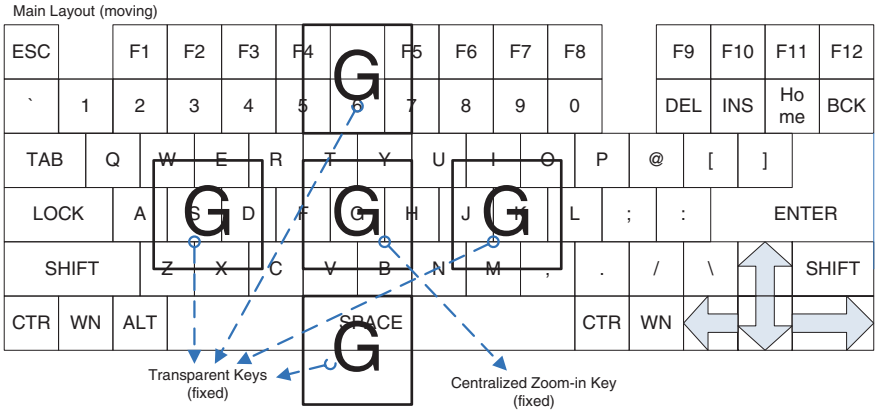


Fig. 5 The proposed keyboard

up, down, and center. These increased keys are used for helping users know the selected key while they are controlling the keyboard. The situation when users are controlling the keyboard, users only could look at one point (only could focus on one point). It means that when user is looking at the "A" character for instance, user could not see the other characters. In this case, when users want to select "A" character, firstly users find where the "A" position is. After users know the position of "A", users control the keyboard by looking at the one of increased key. For instance, if the position of "A" is on left, users look at increased key on left until "A" character moves to center. While "A" character moves to center, users still have to hold the gaze. In this situation (waiting the desired character moves to center and the holding gaze is required), users also have to know what the character on center is. To help users know the character on center while controlling the keyboard, the character on center is shown in four increased key on left, right, up, and down.

2.3 Implementation

The moving keyboard is realized on EBHCI by using single infrared camera (NetCowBow DC-NCR 131) and a Head Mounted Display (HMD) mounted on glasses. We use Optiplex 755 dell computer with Core 2 Quad 2.66 GHz CPU and 2G RAM. The infrared camera could maintain the lighting condition and we placed it on glasses for improving the resolution. The HMD mounted on glasses could maintain the distance between user and display. This hardware is shown in Fig. 6. The process for creating teu enlarged key is shown in Fig. 7.

Firstly, we crop the key on center and continue to increase the size. The increased key is pasted on the center replacing the previous key (normal size).

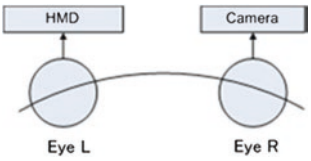


Fig. 6 The used hardware (Head Mount Display: HMD for *left eye* and NIR camera with NIR light source for *right eye*)

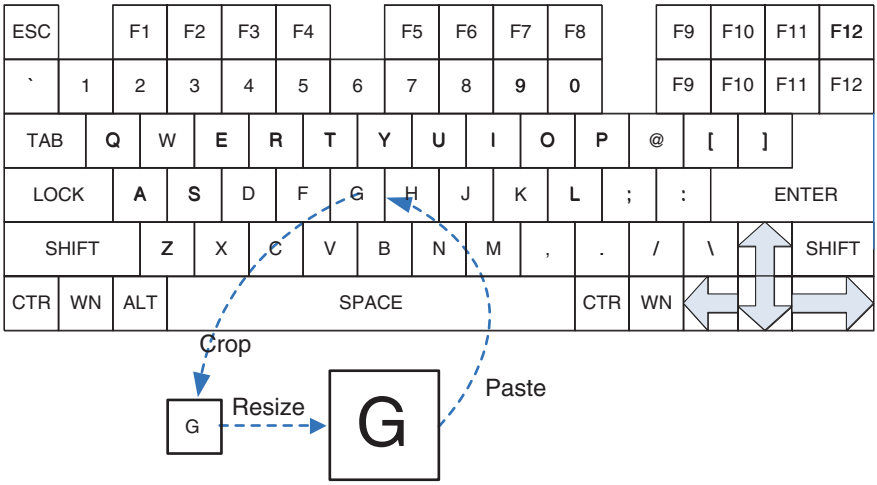


Fig. 7 The process of increasing key

To avoid covering main layout with new key size, we use the transparent paste method and show the both keys. Besides pasting on center, we also paste the new key on four places for controlling keyboard (left, right, up, and down).

All the software required for the proposed EBHCI with moving keyboard was developed by using C++ Visual Studio 2005 and OpenCV, an image processing library, downloaded free from the website. Figure 8a, b shows an example of screen shot image of the computer screen of the proposed EBHCI with fixed keyboard and that of moving keyboard, respectively.

At the bottom left of Fig. 8a, acquired user's face image is shown. The extracted corneas are marked with white circles in the face image. The enlarged eye image is also shown at just above the face image. He is trying to type "sagauniversity" using his eyes only. He initiated program of EBHCI program. Then he opens his window of word processing which is shown in near the center and typing the characters with fixed keyboard which is shown at near the top of the image by his eyes only. On the other hand, the extracted eye image is shown at the bottom right of Fig. 8b while the extracted ellipsoidal shape of cornea is shown on the left of the extracted eye image. Five enlarged keys (left/right/top/bottom/center) are shown in the center of the Fig. 8b. At this moment, user is looking at the key of "G". Therefore, the five

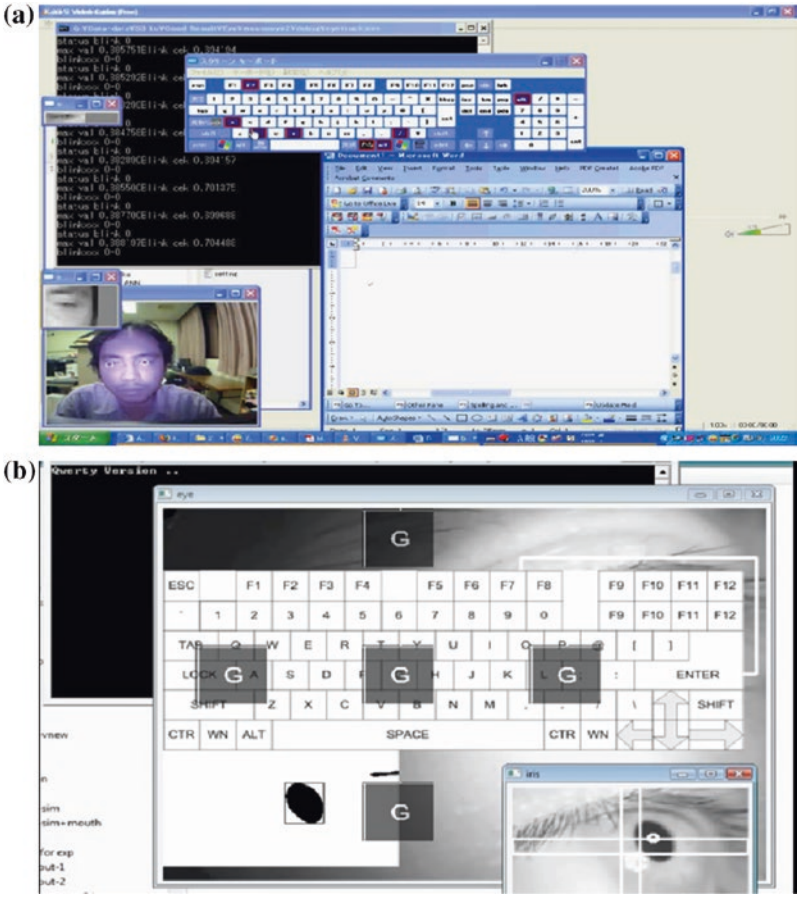


Fig. 8 Example of screen shot image of the computer screen of the proposed EBHCI. **a** Fixed keyboard. **b** Moving keyboard

keys are indicating "G". "H" is appears when user looks at right direction of "G" 0.4 s after. That is the same thing for the other keys. "G" is selected and determined when user looks at the center key, "G" for 0.7 s. This is the principle of the EBHCI with moving keyboard.

2.4 Experimental Results

The experimental discussed below tested the performance of the proposed keyboard. We measure accuracy, time required for typing of the proposed method, and compare the accuracy between our proposed keyboard and an existing one (fixed keyboard type).

Table 1 Result of accuracy

User	Expertise	Moving keyboard (%)	Fixed keyboard (%)
1	Expert	100.00	92.86
2	Expert	100.00	76.19
3	Beginner	82.14	71.43
4	Expert	100.00	85.71
5	Beginner	71.43	78.57
6	Expert	100.00	66.67
Average		92.26	78.57

Table 2 Required times for typing a word

User	Expertise	Moving keyboard(s)	Fixed keyboard(s)
1	Expert	117.50	154.00
2	Expert	138.67	195.33
3	Beginner	180.50	275.00
4	Expert	101.00	197.33
5	Beginner	161.50	213.00
6	Expert	109.00	227.00
Average		134.69	210.28

The experiment was conducted by user types “sagauniversity” using EBHCI with our proposed keyboard. After calibration step, user selected character on keyboard by controlling the keyboard using their eye. The experiment was conducted for four times. The key-in success rate and the time required is shown in Tables 1 and 2, respectively.

As shown in Tables 1 and 2, the key-in success rate is improved through experiences. Also, it is found that the proposed EBHCI with moving keyboard is superior to the EBHCI with fixed keyboard for both success rate and the time required for key-in.

Using EEG sensor, frequency component analysis is made for the proposed EBHCI with moving keyboard and that with fixed keyboard (Fig. 9 and Table 3).

The results shows that alpha frequency component of EEG signals is dominant when the users use the proposed EBHCI with moving keyboard while beta frequency component is dominant when the users use that with fixed keyboard. Therefore, it may say that users’ stress when users use the proposed EBHCI with moving keyboard is smaller than that with fixed keyboard because alpha frequency is dominant when the users are in relax status. Psychological status changed from sympathetic to parasympathetic when the users use the proposed EBHCI with moving window. Therefore, users’ tiredness when users use the proposed EBHCI with moving window is much less than that for the fixed keyboard.

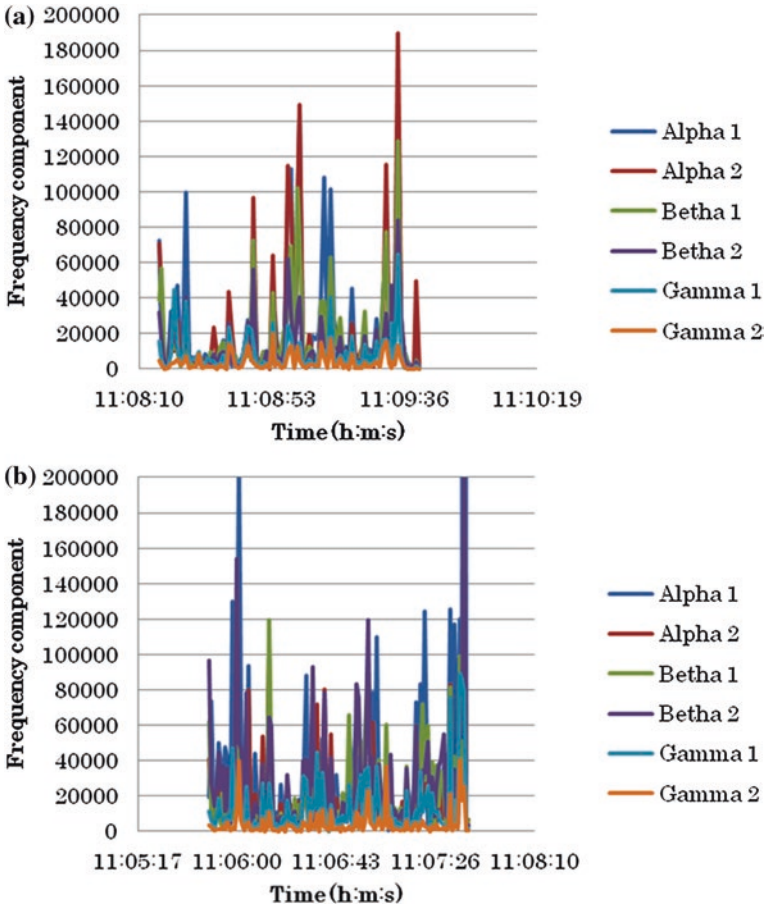


Fig. 9 Results from frequency component analysis of EEG signals when users use EBHCI. **a** Moving keyboard. **b** Fixed keyboard

Table 3 Peak alpha frequency component of EEG signals when users use EBHCI

No.	Nationality	Moving	Fixed	Comparison
1	Indonesian	4.93E+05	3.55E+05	27.99
2	Sri Lankan	1.91E+06	6.26E+05	67.23
3	Vietnamese	9.13E+05	3.05E+05	66.59
4	Thailand	1.95E+06	5.01E+05	74.31
5	Thailand	5.74E+05	3.23E+05	43.76
6	Japanese	1.52E+05	1.13E+05	25.46

3 Applications of Eye Based Human Computer Interaction: EBHCI with Moving Keyboard

3.1 Mouse Operations

Mouse operation can be done with the proposed EBHCI. Figure 10 shows example of the screen shot image of mouse operations. Cursor can be manipulated by human eyes only. Clicking icons can also be possible by human eyes only. Drag and drop operations can be done by human eyes only. Even for Ctl-Alt-Del operation is capable by human eyes only because gaze location can be detected by EBHCI.

3.2 Electric Wheel Chair: EWC Control

EWC control can be done with the proposed EBHCI. Figure 11 shows outlook of the EWC control by human eyes only. User wears HMD and NIR camera with NIR light source.

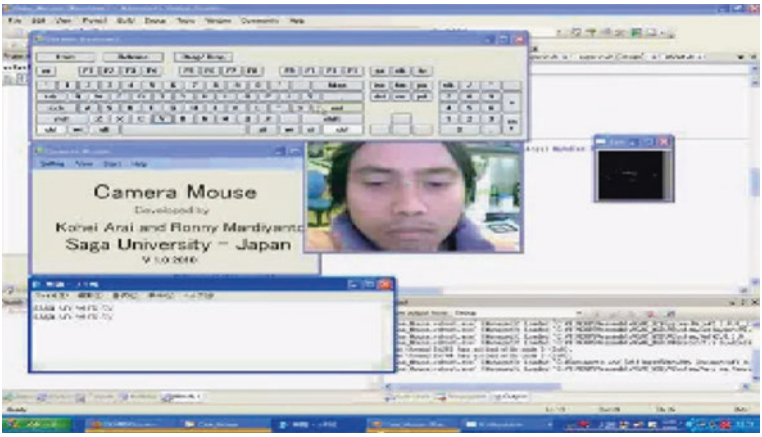


Fig. 10 Camera mouse

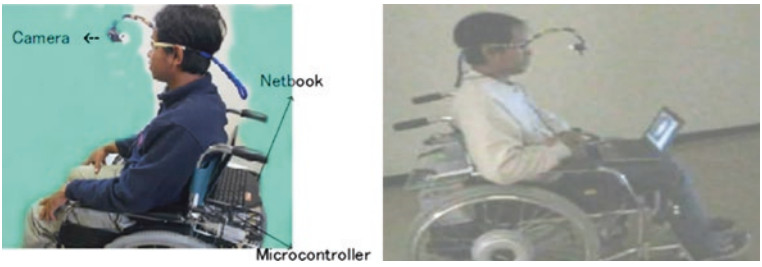


Fig. 11 EWC control by human eyes only

The EBHCI with tree keys (turn left, go straight ahead, turn right) is installed in the netbook PC. Information of selected and determined keys is transferred to microprocessor which allows control the EWC. Thus user sitting on the EWC can move forward, turn left and right by human eyes only. EWC is stopped when user looks the top and bottom column, other than middle column of left/right/center for safety reason. Users look down when they find obstacles while users look up when they are surprised. Therefore, EWC would better to stop in such cases. Move backward is obviously dangerous.

3.3 Having Meal Aid

Patients who lie down on the bed can retrieve their desired food on the meal tray by human eyes only. Figure 12a shows schematic view of the proposed having meal aid system with robot arm equipped beside their bed. Magic hand is attached

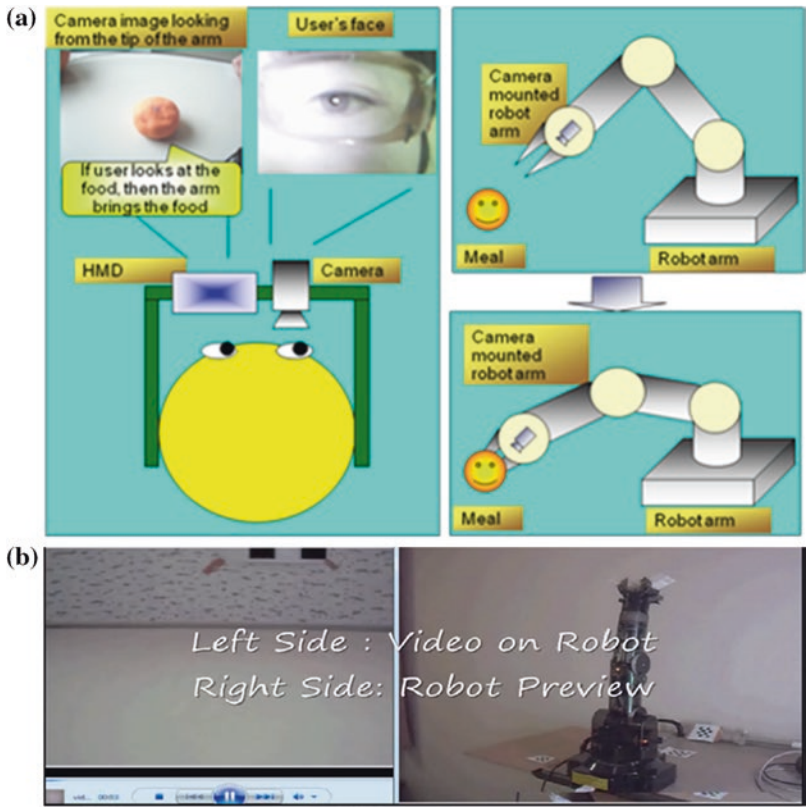


Fig. 12 The robot arm for having meal aid. **a** Schematic view. **b** Outlook of the robot arm for having meal aid

to the tip of the robot arm together with NIR camera and NIR light source. Figure 12b shows outlook of the robot arm for having meal aid.

Acquired image by the NIR camera mounted on the tip of the robot arm can be seen by user. User can control robot arm using the acquired image and navigate the tip of robot arm to the desired food. Thus user retrieve any food on the meal tray by magic hand.

3.4 Communication Aid

The proposed EBHCI allows communication aid. Figure 13 shows schematic view of the communication aid with the proposed EBHCI.

Using computer input by human eyes only, sentences can be created with the simplified keys shown in Fig. 13. This is just an example of Japanese language communication aid system. These 9 keys are enough to create any sentences. Under the 9 characters, there are 5 character candidates. Therefore, 45 of fundamental Japanese character sounds can be selected and determined. After creation of sentences, type to talk software tool allows pronunciation.

3.5 Information Collection Aid

Information collection can be done with the proposed EBHCI with the main menu shown in Fig. 14. There are six menus. When user select web browser, then web

Fig. 13 Schematic view of the communication aid with the proposed EBHCI

Fig. 14 Main menu of the proposed information correction aid system

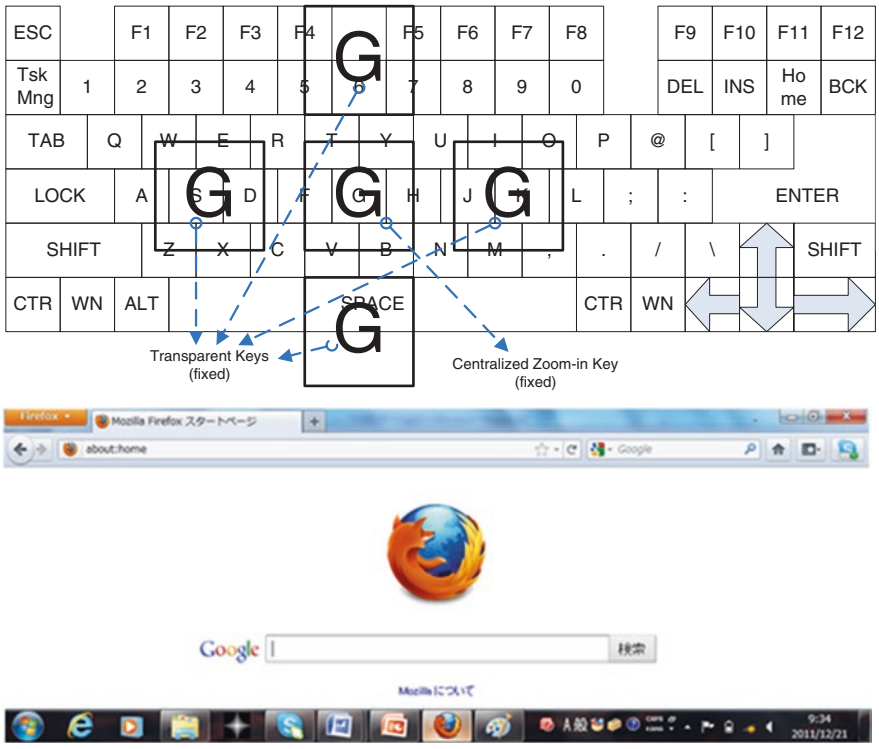


Fig. 15 Web browser and EBHCI with moving keyboard for keyword input

browser and EBHCI with moving keyboard for keyword input is appeared as shown in Fig. 15.

Watching TV, listening radio, phoning, reading e-book, and learning with the accessed e-learning contents are also available for users as shown in Fig. 16.

During watching TV, screen shot can be done and the required information for purchasing something (price, name of the product, phone number, address, etc.) can be extract when user select extract key by human eyes only. Radio station can be selected with human eyes only. Therefore, user can turn desired radio station with desired volume. Phoning is available by the proposed EBHCI with the menu shown in Fig. 16c. In the middle column, there are 10 keys from 0 to 9. Using left and right key, user can select and determine the desired phone number. After that user can call someone when user select call key situated the top center. Reading e-book can also be available. After download the e-book contents through the search engine, contents can be displayed together with forward/backward of the page can also be done with human eyes only. Figure 16e shows outlook of the information collection aid system. In Fig. 16e, user is reading e-comic content.

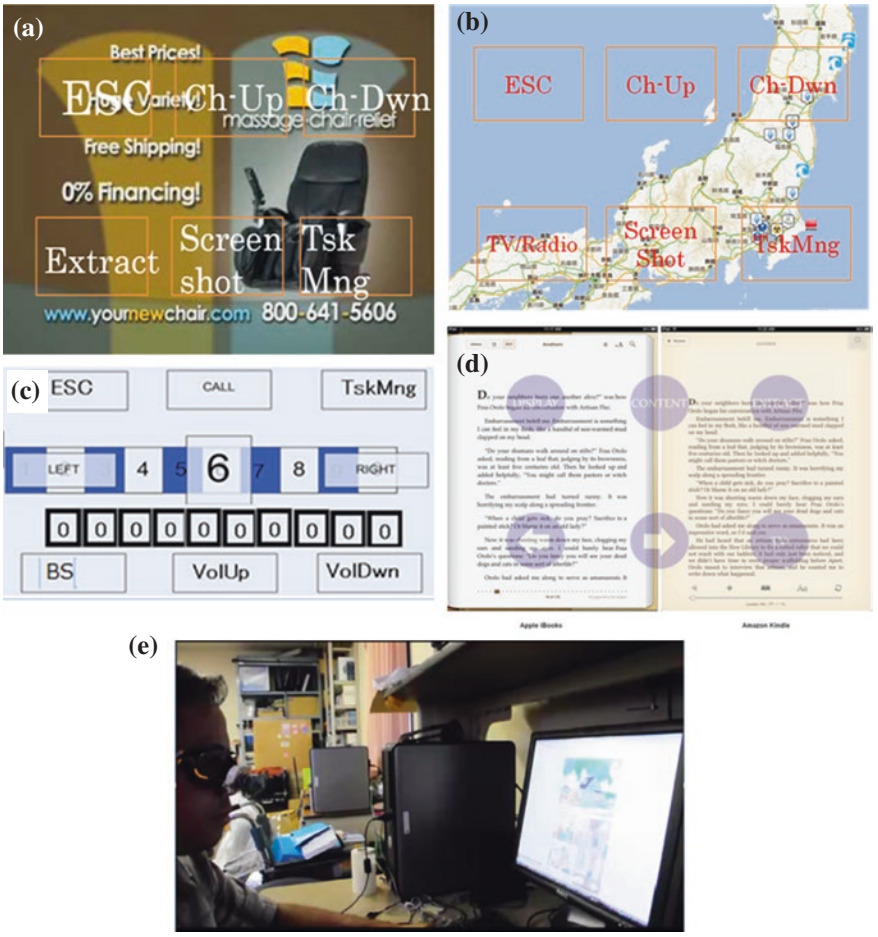


Fig. 16 Information collection aid with the proposed EBHCI. **a** Watching TV. **b** Listening radio. **c** Phoning. **d** Reading e-book. **e** Outlook of the information collection aid system (reading e-comic)

3.6 Service Robot Control

Figure 17 shows schematic diagram of the proposed service robot. Also, Fig. 18a shows outlook of the service robot while Fig. 18b shows acquired image by the service robot.

Using the proposed EBHCI, user can look at the scenery which acquired by the NIR camera which is mounted at the tip of the service robot. User's eye image is acquired with another NIR camera mounted on the user's glass. Therefore, EBHCI is fully functional. Service robot can be controlled by user's eye. Service robot turns left when user looks left direction while the robot turns right when user

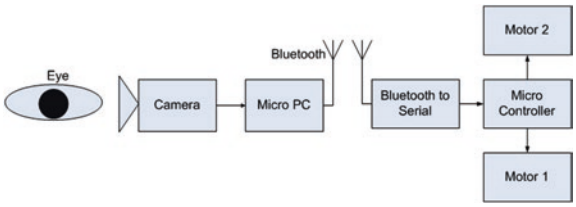


Fig. 17 Schematic diagram of the proposed service robot

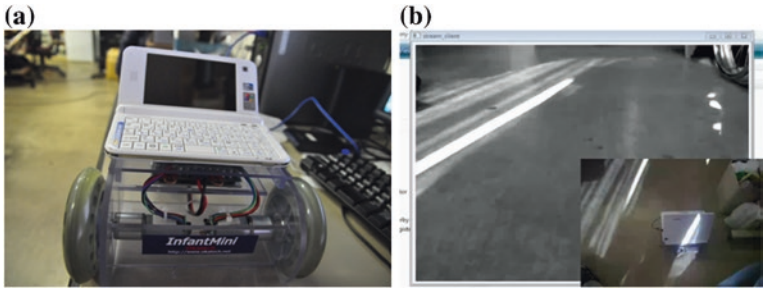


Fig. 18 Service robot. **a** Outlook. **b** Acquired image by service robot

looks right direction. Meanwhile, the robot goes straight ahead when user looks ahead (middle center). Communication aid is also available for the service robot. Therefore, user can enjoy conversations with the other persons in front of the service robot, even if the other persons are far from the user. Figure 18b shows the acquired image by service robot. At the bottom right, acquired image by me is also shown in Fig. 18b.

3.7 e-Learning System

During lessons with e-learning content, the gaze location at which students is looking is important. Leaning efficiency is improved when students is looking at an appropriate location. Also, leaning efficiency is improved when students concentrate to the e-learning content. By using the proposed EBHCI, gaze location and students' concentration can be monitored by lecturer in synchronized learning mode. Lecturer can notify students when they are not concentrated and they are looking at inappropriate portion of the e-learning content. Figure 19 shows the difference between the gaze location and the location at which lecturer would like students to look, blinking, and Peak Alpha Frequency: PAF amplitude (alpha frequency component at peak frequency of EEG signals). PAF amplitude goes down when the difference is increased while students blink when the difference is

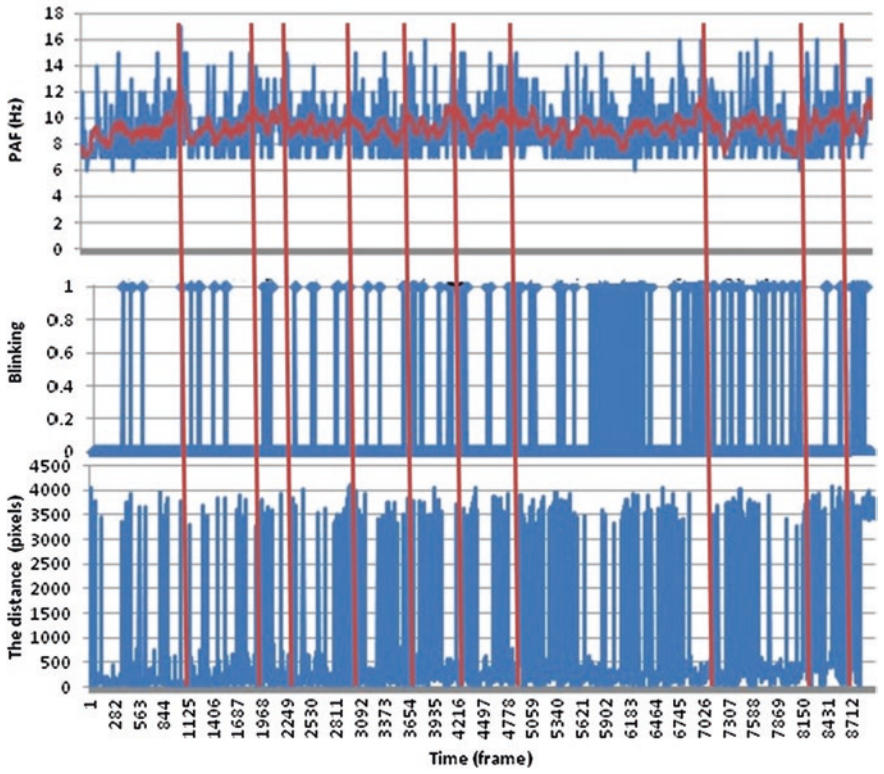


Fig. 19 Relation among PAF amplitude, blink timing and the difference between the gaze location and the location at which lecturer would like students are looking

getting large. The experiment is carried out when Japanese students have a lesson with e-learning content of English reading which is shown in Fig. 20. In the text, black circle shows the location at which lecturer would like students to look.

Another example of the application of the proposed EBHCI to e-learning lessons is shown in Fig. 21. It is just a screen shot of the e-learning content of "Kabuki Lesson". This content can be divided into four parts, (1) Lecturer's face, (2) Moving picture, (3) Keywords and descriptions, (4) others. Just after the lesson, students have to have an achievement test. The score of the achievement test and correlation between the location at which students are looking and their score are shown in Table 4.

The result is obvious. The score of the students who are looking at (1) portion mostly is not so good while the score of the students who read the keywords and descriptions, (2) mostly is pretty good. Meanwhile, the score of the students who look at (3) or (4) mostly is relatively bad.

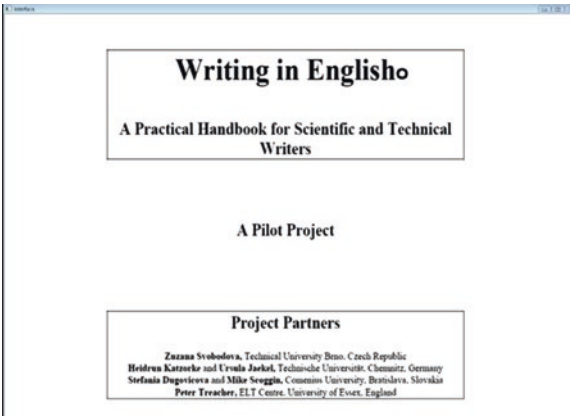


Fig. 20 e-learning content of English reading lesson used for the experiment

Fig. 21 Screen shot of the e-learning content for "Kabuki Lesson"

3.8 Wearable Computer

Configuration of the proposed wearable computing system is shown in Fig. 22.

Users wear the Head Mount Display: HMD mounted glass. The glass also carries backward looking camera. The backward looking camera looks users' eye. Users' line of sight can be estimated with the acquired images of the backward looking camera.

The required processes for estimation of the position at which users is looking on the display screen of the HMD through line of sight estimation are done in the Tablet PC. Also gathered data of the user's location and accelerometer data

Table 4 Score of the achievement test and correlation between score and the location at which students are looking

Score	1	2	3	4
11	4	10	3	2
10	4	9	5	1
10	5	9	3	2
8	5	8	4	2
8	6	7	3	3
8	5	7	6	1
7	6	6	5	2
6	6	5	4	4
6	4	6	5	4
4	6	3	4	6
Correlation	−0.57844	0.980581	−0.28022	−0.81132

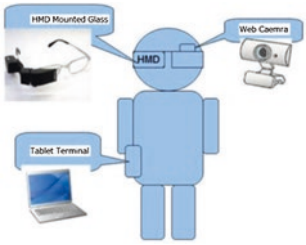


Fig. 22 Configuration of the proposed wearable computing system

are handled in the mobile terminals or Tablet PC. Therefore, user's location can be known with the GPS receiver data. Even if the user is situated indoor, WiFi capability allows estimation of the user's location by using signal lengths from the nearest three WiFi access points.

Wearable computing with Input-Output devices Base on Eye-Based Human Computer Interaction: EBHCI which allows web services, location/attitude/health condition monitoring is proposed. Through implementation of the proposed wearable computing system, all the functionality is confirmed. It is also found that the system does work well. It can be used easily and also is not expensive. Experimental results for EBHCI show excellent performance in terms of key-in accuracy as well as input speed. It is accessible to internet, obviously, and has search engine capability.

4 Conclusion

Computer input by human eyes only is proposed for disabled and elderly persons, in particular, together with its applications to communication aid, having meal aid, information collection aid, watching TV aid, listening to radio aid, phoning aid, Electric Wheel Chair: EWC control, service robot control. Through experiments

with disabled and normal persons including Alzheimer diseased persons, it is found that the proposed Eye Based Human Computer Interaction: EBHCI which includes computer input by human eyes only can be used for disabled and normal persons for a variety of applications.

Acknowledgments The author would like to thank all the patients who are contributed to the experiments conducted. The author also would like to thank Professor Dr. Takao Hotokebuchi, President of Saga University for his support this research works.

References

1. Arai, K., Uwataki, H.: Computer input by human eyes only with cornea center detection which allows users' movements. J. Electr. Eng. Soc. Jpn. Trans. (C) **127**(7), 1107–1114 (2007)
2. Arai, K., Yamaura, M.: Blink detection accuracy improvement with morphologic filters for computer input by human eyes only. J. Image Electron. Eng. Soc. Jpn. **37**(5), 601–609 (2008)
3. Arai, K., Yajima, K.: Communication aid system by means of comouter input by human eyes only. J. Electr. Eng. Soc. Jpn. Trans. (C) **128**(11), 1679–1686 (2008)
4. Arai, K., Mardiyanto, R.: Computer input with blink detection based on Gabor filter. J. Vis. Soc. Jpn. **29**(Suppl. 2), 87–90 (2009)
5. Arai, K., Mardiyanto, R.: Real time blinking detection based on Gabor filter. Int. J. Hum. Comput. Interact. **1**(3), 33–45 (2010)
6. Arai, K., Yamaura, M.: Computer input with human eyes only using two Purkinje images which works in a real time basis without calibration. Int. J. Hum. Comput. Interac. **1**(3), 71–82 (2010)
7. Arai, K., Mardiyanto, R.: Camera mouse and keyboard for handicap person with trouble shooting capability, recovery and complete mouse events. Int. J. Hum. Comput. Interac. **1**(3), 46–56 (2010)
8. Arai, K., Mardiyanto, R.: A prototype of electric wheel chair control by eye only for paralyzed user. J. Robot Mechatron. **23**(1), 66–75 (2010)
9. Arai, K., Mardiyanto, R.: Autonomous control of eye based electric wheel chair with obstacle avoidance and shortest path finding based on Dijkstra algorithm. Int. J. Adv. Comput. Sci. Appl. **2**(12), 19–25 (2011)
10. Arai, K., Mardiyanto, R.: Eye-based human-computer interaction allowing phoning, reading e-book/e-comic/e-learning, Internet browsing and TV information extraction. Int. J. Adv. Comput. Sci. Appl. **2**(12), 26–32 (2011)
11. Arai, K., Mardiyanto, R.: Eye based electric wheel chair control system-I(eye) can control EWC. Int. J. Adv. Comput. Sci. Appl. **2**(12), 98–105 (2011)
12. Arai, K., Mardiyanto, R.: Evaluation of users' impact for using the proposed eye based HCI with moving and fixed keyboard by using eeg signals. Int. J. Res. Rev. Comput. Sci. **2**(6), 1228–1234 (2011)
13. Arai, K., Mardiyanto, R.: Electric wheel chair controlled by human eyes only with obstacle avoidance. Int. J. Res. Rev. Comput. Sci. **2**(6), 1235–1242 (2011)
14. Arai, K., Mardiyanto, R.: Evaluation of users' impact for using the proposed eye based HCI with moving and fixed keyboard by using eeg signals. Int. J. Res. Rev. Comput. Sci. **2**(6), 1228–1234 (2012)
15. Arai, K., Mardiyanto, R.: Electric wheel chair controlled by human eyes only with obstacle avoidance. Int. J. Res. Rev. Comput. Sci. **2**(6), 1235–1242 (2012)
16. Arai, K., Mardiyanto, R.: Camera mouse including "Ctl-Alt-Del" key operation using gaze blink and mouth shape. Int. J. Adv. Comput. Sci. Appl. **4**(3), 83–91 (2013)

17. Arai, K., Mardiaynto, R.: Method for psychological status estimation by gaze location monitoring using eye-based human-computer interaction. Int. J. Adv. Comput. Sci. Appl. **4**(3), 199–206 (2013)
18. Arai, K., Mardiyanto, R.: Speed and vibration performance as well as obstacle avoidance performance of electric wheel chair controlled by human eyes only. Int. J. Adv. Res. Artif. Intell. **3**(1), 8–15 (2014)
19. Arai, K.: Wearable healthy monitoring sensor network and its application to evacuation and rescue information server system for disabled and elderly person. Int. J. Res. Rev. Comput. Sci. **3**(3), 1633–1639 (2012)
20. Arai, K.: Wearable computing system with input output devices based on eye-based Human Computer Interaction: HCI allowing location based web services. Int. J. Adv. Res. Artif. Intell. **2**(8), 34–39 (2013)
21. Arai, K.: Vital sign and location/attitude monitoring with sensor networks for the proposed rescue system for disabled and elderly persons who need a help in evacuation from disaster areas. Int. J. Adv. Res. Artif. Intell. **3**(1), 24–33 (2014)
22. Arai, K.: Method and system for human action detection with acceleration sensors for the proposed rescue system for disabled and elderly persons who need a help in evacuation from disaster areas. Int. J. Adv. Res. Artif. Intell. **3**(1), 34–40 (2014)

New Ideas for Brain Modelling 2

Kieran Greer

Abstract This paper describes a relatively simple way of allowing a brain model to self-organise its concept patterns through nested structures. For a simulation, time reduction is helpful and it would be able to show how patterns may form and then fire in sequence, as part of a search or thought process. It uses a very simple equation to show how the inhibitors in particular, can switch off certain areas, to allow other areas to become the prominent ones and thereby define the current brain state. This allows for a small amount of control over what appears to be a chaotic structure inside of the brain. It is attractive because it is still mostly mechanical and therefore can be added as an automatic process, or the modelling of that. The paper also describes how the nested pattern structure can be used as a basic counting mechanism. Another mathematical conclusion provides a basis for maintaining memory or concept patterns. The self-organisation can space itself through automatic processes. This might allow new neurons to be added in a more even manner and could help to maintain the concept integrity. The process might also help with finding memory structures afterwards. This extended version integrates further with the existing cognitive model and provides some new conclusions.

Keywords Neural modelling · Self-organise · Connection strengths · Mathematical process

This is an extended version of the paper accepted by The Science and Information Conference (SAI'14), London, 27–29 August, 2014.

K. Greer (✉)
Distributed Computing Systems, Belfast, UK
e-mail: Kgreer@distributedcomputingsystems.co.uk
URL: http://distributedcomputingsystems.co.uk

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_2

1 Introduction

This paper describes a relatively simple way of allowing a brain model to self-organise its concept patterns through nested structures. For a simulation, time reduction is helpful and it would be able to show how patterns may form and then fire in sequence, as part of a search or thought process. The equation that is tested is possibly a simplified version of existing ones ([13] Eq. 1, for example) which would also consider the synaptic connections in detail. This model is very much generalised and considers the pattern firing properties only. As such, it would be a quicker algorithm to use and so it may allow for more economic test runs that are concerned with this aspect in particular. The algorithm is attractive because it is still mostly mechanical and therefore can be added as an automatic process to any simulator. The process can also be used as a basic scheduling or counting mechanism and so it might be possible to add more mathematical operations to a brain-like model, without changing the basic components too much.

The pattern model that is suggested is based on one of the most commonly occurring structures in the real world and so it is clearly understood and could be added to a computer program relatively easily. It is however a generalised high-level idea, without exact cell workings or synapse connections, for example. After the model and the reasons for suggesting it are described, some tests based on a relatively simple equation will be presented, to show the correctness of the idea. It will then be shown how the ensemble patterns can fit in with the current cognitive model and some other new ideas will be presented, along the lines of self-organisation and regulation.

The rest of this paper is organised as follows: Sect. 2 describes some related work. Section 3 describes the main ideas of the new model. Section 4 describes some tests and results that confirm the main idea. Section 5 integrates the ensembles with the current cognitive model, while Sect. 6 gives some conclusions on the work.

2 Related Work

This section is more biologically-oriented, where the author is not particularly expert, but the papers might make some relevant points. The aim is to show that the proposed structure is at least practical. The paper [13] is quite closely related and includes a number of important statements. It gives one example of an equation for the firing rate that includes the whole range of inputs, including external sensory and other neurons' excitatory/inhibitory input. It states that for the firing to be sustained, that is, to be relevant, requires sufficient feedback from the firing neurons, to maintain the level of excitation. Once the process starts however, this can then excite and bring in other neurons, when inhibitory inputs also need to fire, to stop the process from saturating. A weighted equation is given to describe how the process can self-stabilise if 'enough' inhibitory inputs fire and

a comparison with the equation is given in Sect. 4. The paper [10] also studies the real biological brain and in particular, the chemospecific steering and aligning process for synaptic connections. It notes that there are different types of neuron, synaptic growth and also theories about the processes. While current theory suggests that growth is driven by the neuron itself, that paper would require it to be driven almost completely by the charged 'signal'. Current theory also suggests that the neuron is required first, before the synapses can grow to it. However, they do note a pairwise chemospecific signalling process, as opposed to something that is just random and they also note that their result is consistent with the known preferences of different types of 'interneurons' to form synapses on specific domains of nearby neurons.

The paper [14] also describes how neurons can change states and start firing at different rates. The paper [11] describes that there are both positive and negative regulators. The positive regulators can give rise to the growth of new synaptic connections and this can also form memories. There are also memory suppressors, to ensure that only salient features are learned. Long-term memory endures by virtue of the growth of new synaptic connections, a structural change that parallels the duration of the behavioural memory. As the memory fades, the connections retract over time. So, there appears to be constructive synaptic processes and these can form memory structures. The paper [12] is more computer-based and describes tests that show how varying the refractory (neuron dynamics) time with relation to link time delays (signal) can vary the transition states. They note that it is required to only change the properties of a small number of driver nodes, which have more input connections than others. These nodes can control synchronization locally and they note that depending on the time scale of the nodes, some links are dynamically pruned, leading to a new effective topology with altered synchronization patterns. The structures tested are larger control loops, but it is interesting that the tests use very definite circular pattern shapes. The work of Santiago Ramón y Cajal[1] has been suggested as relevant, in particular, with relation to pacemaker cells. This is definitely interesting and will be discussed in later sections. However, while Cajal appears to classify neurons, based on location defining their function, this paper does not consider different neuron types. It is only interested in location for allowing them to operate as part of a thinking process.

The author has proposed neural network or cognitive models previously [4–7] and it is hoped that this paper does not contradict that work. The aim has been to build a computer model that copies the brain processes as closely as possible, so as to realise a better or more realistic AI model. In particular, this paper might be considered an extension of [4]. It is still a computer program however and a close inspection of how the biological components work has only tentatively been introduced. The goal is to try and make the underlying processes as mechanical, or automatic, as possible, so that the minimum amount of additional intelligence is required for them to work.

[1] http://www.scholarpedia.org/article/Santiago_Ramón_y_Cajal.

Earlier themes included dynamic or more chaotic linking, time-based events, pattern formation with state changes, clustering and even hierarchical structures with terminating states or nodes. This paper is associated with some of the earlier work, including the more chaotic neural network structures [4, 5], or the pattern forming levels of the cognitive model [6, 7].

3 Reasons for the Firing Patterns Model

It is important to remember that an energy supply is required to cause the neurons to fire. It is probably correct to think that the brain must receive a constant supply of energy to work. If a neuron fires, this would necessarily use up some of the energy, which is why the supply must be refreshed. If thinking about the single neuron, it is thought that ion channels cause the neuron activation, where pressure or force is not the main mechanism.[2] A neuron itself does not have the intelligence to fire, in the sense that it is reactive and not proactive. The fact that inhibitors are used to suppress the firing rate shows that the neurons cannot decide this for themselves. They also need an automatic mechanism to switch off. The activation might be traced back to the external stimulus, which is a continuous energy source, although pressure would be another one [15]. Note also that the brain would be expected to give feedback, which in turn might change the input, and so on. Therefore, if considering the energy used by the system, it would make sense to nest sub-concepts, based on the idea of distance alone.

3.1 Sub-concepts

When thinking about brain firing patterns, it appears to be very random and complex. Pictures or scans of activity however usually show distinct brain areas that are active, where this in itself is interesting. If the firing activity was completely random, then specific areas should possibly not be present, as synapses would travel in any direction to other neurons. So there is an indication here that the firing activity is contained. This then means that it could be inwards, or inside the originally activated area. This can sometimes be almost the whole brain, however. A simple example might also illustrate something. Thinking about a coffee cup, the cup itself can be imagined, sitting by itself. To expand the scenario, possibly a kettle is imagined, filling the cup with water. A specific action would also be invoked here. But is it the case that when the kettle is imagined, the whole kitchen scenario is retrieved? Even

[2] Pressure is not very relevant for this paper, but was used as part of an earlier argument 4 to help the synaptic structures to grow and re-balance.

if this is done sub-consciously, it would put the scenario into a familiar context, when the kettle and subsequent actions are then easy. This is therefore interesting, because it suggests that the larger activity areas could be these larger concepts that then contain lots of smaller ones inside of them. The kitchen can contain a kettle and a cup, for example. If firing is restricted to the kitchen scenario, it is easy to imagine that it might activate the other smaller kitchen-related concepts. This is also a part of how we try to model the real world, mathematically or formally, in our processes or diagrams, for example.

Further, a single concept can be imagined by itself and even without a background. But the addition of context, invoked by an action or other object, forces the relevant background, even if it is relatively weak. So is the coffee cup and kettle driving the activation and triggering the kitchen that lives somewhere else, or is the span or area of activation now wide enough to activate part of the parent kitchen concept? If the firing was always inwards then activating a larger concept would be difficult, so at least some lateral or outwards positive activation is required. But then again, a coffee cup or kettle might be terminal states that are accessed directly (see possibly the θ value of Eq. 1 in [13]). As separate pattern groups also need to link, lateral signals could excite a general area between them as well. An action might even originate in a different brain region, bringing all of the connected areas into play. Figure 1 is a schematic of the general idea. There is a larger pattern with nested ones and some excitatory and some inhibitory signals. Traversing the larger area would bring in more of the background patterns or images.

So the currently firing pattern is what defines the brain state. If there is no other way of controlling this, the ability to switch off the other areas in an automatic manner is required. If the parent provider encapsulates the new or most active next state, then this activity could be through a relatively simple and easy to understand process. The inhibitors will naturally send more negative feedback to their neighbouring environment, thereby weakening the parent signal compared to the new firing pattern. If new areas inside of them then become active, the process can repeat again. The most obvious catch is the fact that lateral linking and activation is always required and also from other brain regions that perform other functions. It is however, still a natural way to self-organise.

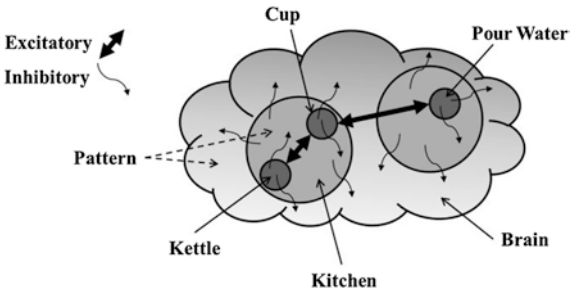


Fig. 1 Example of nested concepts in a brain area

3.2 Mathematical Processes

Another interesting use of the nested patterns is not to retrieve sub-concepts, but to implement a basic timer, counter or even battery, that could be part of more mathematical processes. The idea of a battery, counter or timer here, refers to controlling the energy supply to a particular group of neurons. A more general supply is converted into one that can schedule something, or run for a pre-determined amount of time. It might be part of a whole cycle of pattern activations as follows: Some pattern activates another pattern that is the on-switch to a timer or battery. The on-switch activates the outer-most pattern of the nested group that makes up the new structure. This cycles inwards as described, until the inner-most pattern is activated. This might be 3 nested patterns, for example. Each nested pattern, when activated, might send a signal somewhere, but the inner-most one also sends a signal to the off-switch pattern that is beside the on-switch. The off-switch sends inhibitors to the on-switch, asking it to turn off. This then removes the signal inducer to the new structure and the whole cycle can stop.

As this is only an idea, an alternative and possibly better mechanism would be to slowly increase both the excitatory and inhibitory signals. The first activation phase from the outer-most pattern to the first nested one might not activate all neurons one level in. This also means however that they would not all send inhibitory signals back. So the outer-most pattern might be able to send several phases of signal before it receives an overwhelming inhibitory signal. The same situation can occur between any of the nested pattern sets. Continual activation signals can switch on more neurons the next level in, but then they also send more inhibitory signals back. If the excitatory signal is mostly inwards and the inhibitory one mostly outwards, this should result in the whole region eventually switching itself off.

Slightly more doubtful: if the inhibitory signal only affects active neurons, then they can possibly fire in any direction, because the inner patterns will receive less than the outer ones, based on time events and so the outer ones will switch off first. Signal strength might also be a factor, if the excitatory signal is stronger than the inhibitory one. In that case, the inhibitory signal sent forwards to the nested patterns might not be strong enough to deter their activation; but the inhibitory signal sent back to the parent would then need to be from multiple nested or enclosed patterns, not just the immediate one. If signal strength is a factor, this would actually make the equation of Sect. 4.1 even better and might also work better within a hierarchy setting, where signal is sent back to any ‘upper’ level, not just an ‘enclosing’ level. Also implicit then is the fact that there is a difference in how the excitatory and the inhibitory signals are created, where the excitatory ones have more direction.

So there would still be the desired and gradual build-up of signal and shut-down afterwards, even if the inter-relations between the basic components are slightly different. Note that these cases are started by a constant, external energy supply, which then gets shut-down or ignored. It would also be helpful if inhibitors could change a neuron state without switching it off completely and ideas of localised firing already exist [12]. The schematic of Fig. 2 tries to describe the most general

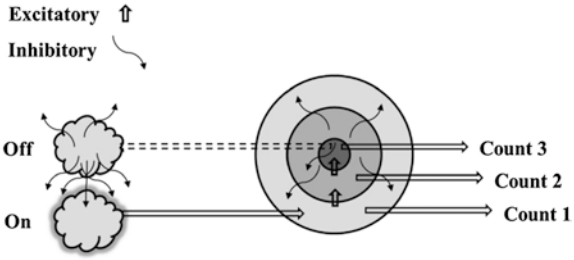


Fig. 2 Example of a timer, counter or battery

case. Some area of the brain excites and starts the outer-most pattern firing. This is the 'on' switch with a signal to the outer-most circle. The pattern cycles through to the inner-most one that can then ask for the provider to switch off. This is the 'off' switch. Each nested pattern can also send a signal somewhere else, which would implement the counting mechanism. The paper [13] notes another model that already exists called Synfire chains. Synfire chains fire in a definite outwards direction and offer some degree of control through firing stages at different levels. This then leads to problems of explosion from a sustained input and requires noise or other to control the firing rates. So the main question for the model of this paper is whether it can actually occur naturally, as other formations appear to be outward facing. It is worth remembering that clear pattern boundaries get created however and an inward firing process might be replaced by a hierarchical one, without changing anything else. See Sect. 5.2.

As well as deciding to fire, this paper would require the neuron to intelligently control direction. Why would the neurons prefer to fire inwards instead of any direction? The theory of this paper however allows that intelligence to be replaced with an economic reason, based on the conservation of energy. If thinking about stigmergic systems, [2] for example, the ant colony selects the most economic path unintentionally and neurons equally influence each other. The idea of grouping more closely, neurons that fire at the same time, is also the well-known doctrine of Hebb [9]. The search process would also conceivably converge on terminal states [4], where Fig. 3 could help to describe the economic argument. The idea of 'neurons that fire together wire together' requires a link between the two or more groups involved. If search occurs from a broad group to a smaller terminal state; then if that search is outwards, as in Fig. 3a, the distance between the terminal states and the nodes in general is greater than if it is inwards, as in Fig. 3b. Note in particular the case where the terminal

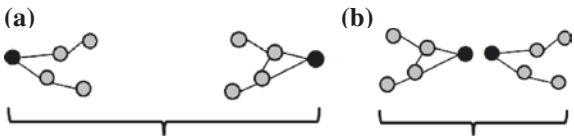


Fig. 3 The two search process that ends with closer terminal states *b* is more economic

states join to complete a circuit. Also, exactly as with stigmergy, if both pattern sets receive the same amount of energy, Fig. 3b will reinforce more, because the signal can take a shorter route. That might just provide a reason why it is easier for the inward pointing search to then connect with another related search area, than the outward pointing one. Therefore, even by chance, a random process might prefer the inward facing groups.

4 Testing and Results

Testing of the theory can be carried out by implementing some basic reinforcement algorithms and updating specific node values, to simulate the different timings of the node pattern activations. The traditional increment/decrement reinforcement algorithm worked well enough to give the desired result. With that, the node value is incremented with excitatory input and decremented with inhibitory input. The decrement value can be weighted to be only a fraction of the increment. Some assumptions are made with regards to the neurons, which helps to simplify the problem further:

- Each neuron has only one excitatory output and one inhibitory output.
- The excitatory output goes only to the other neurons firing in the same pattern.
- The inhibitory output goes only to any other neuron that is in any parent pattern.
- Neurons are in the same pattern if they fire at the same time. This is measured by time increments $t_1 \ldots t_n$.

4.1 Test Conditions

Equation 1 for firing rate networks in [13] is probably a complete version of the equation that might be used. These tests only consider the excitatory/inhibitory part, to measure how the patterns will switch on and off through their interactions. The firing interactions are further restricted by the aforementioned assumptions. The resulting test equation for this paper could therefore be written as follows:

$$X_{it} = \sum_{p=1}^{Pi} Ept - \left(\sum_{k=Pj}^{l} \sum_{y=1}^{m} \sum_{j=1}^{n} (Hjy * \delta) \right)$$

where $y \neq t$ and $i \in P_i$ and *not* $j \in P_i$ and

X_{it} = total input signal for neuron i at time t.
E_p = total excitatory input signal for neuron p in pattern P.
H_{jy} = total inhibitory input signal for neuron j at time y.
δ = weights inhibitory signal.

t = time interval for firing neuron.
y = time interval for any other neuron.
n = total number of neurons.
m = total number of time intervals.
l = total number of active patterns.
P_i = pattern for neuron i.
P = total number of patterns.

In words, the tests measure how the total signal input to each neuron pattern changes. All neurons in the same pattern fire at the same time and send each other their positive signal. Any active neuron also receives a negative signal from any other nested pattern neuron. If two nested patterns are active, for example, the inner-most sends inhibitory signals to both the outer-most pattern and the first nested one. The first nested pattern sends inhibitory signals to just the outer-most one. Over time, neurons continue to fire based on—total pattern firing strength minus total inhibitory firing strength from all other nested patterns.

Table 1 Relative pattern strengths after firing sequences

Neurons	t = 3	t = 4	t = 5
1	7.5	5.0	0.0
2	7.5	5.0	0.0
3	7.5	5.0	0.0
4	7.5	5.0	0.0
5	7.5	5.0	0.0
6	7.5	7.5	5.0
7	7.5	7.5	5.0
8	7.5	7.5	5.0
9	7.5	7.5	5.0
10	7.5	7.5	5.0
11	5.0	7.5	7.5
12	5.0	7.5	7.5
13	5.0	7.5	7.5
14	5.0	7.5	7.5
15	5.0	7.5	7.5
16	0.0	5.0	7.5
17	0.0	5.0	7.5
18	0.0	5.0	7.5
19	0.0	5.0	7.5
20	0.0	5.0	7.5
21	0.0	0.0	5.0
22	0.0	0.0	5.0
23	0.0	0.0	5.0
24	0.0	0.0	5.0
25	0.0	0.0	5.0

4.2 Test Results

The test results are quite straightforward and show the desired set of relative counts or signal strengths. Just the traditional increment/decrement algorithm is shown in Table 1. There are 25 neurons in total and 5 in each nested pattern. The inhibitory signal is set to be half that of an excitatory one, but if a pattern only contains 5 neurons, that leaves a possible 20 other neurons that might send inhibitory signals. Each firing cycle activates a new nested pattern, until all patterns are active. After that, each firing cycle would update signals from all patterns. The inhibitory signal is sent from the inner pattern to its outer ones only, so the inner-most one does not receive inhibitory signals. When all patterns are active, the inhibitory signal builds up to overwhelm the excitatory signal. This of course, depends on the pre-set relative strengths and numbers of excitatory and inhibitory signals.

Neurons 1–5 are the outer-most pattern. Neurons 6–10 are the first nested pattern and so on, until neurons 21–25 are the inner-most nested pattern. At time t1, the first pattern only fires (neurons 1–5). At time t2 pattern 1, then pattern 2 fires. At time t3, pattern 1, then pattern 2, then pattern 3 fire, and so on. The outer patterns have more excitatory input to start with, but as the other patterns switch on and send negative feedback, eventually they will switch off the outer patterns. This would then actually starve the inner patterns of input, until they switch off as well.

5 Cognitive Model

It turns out that the nested ensembles can fit into the current cognitive model, described in [5–7] and most recently in [4]. All of the elements mentioned in earlier papers are still relevant and so the model can be refined further. While it may not be 100 % correct biologically, it is becoming quite detailed and still consistent over the main ideas.

5.1 Hierarchical Nesting

The first thing to think about again is the regions, or nested regions, of pattern ensembles. As the individual elements are likely to be located randomly, duplication can help. For example, if the concepts in question are duplicated they can occur in different locations and collections, but the permanent ensemble will probably be formed where they are located closest to each other. It is noted that duplication also occurs because different parts of the brain store the same concept for different reasons. The most economic group might complete the connections first, satisfy the input requirements and reinforce the most, as in stigmergy. Therefore, duplication makes it more likely that any ensemble can form, or if the neuron is missing, does it just get created? It is also noted that neurons are created from some sort of chemical

reaction and are not required to grow at the end of a synapse, or anything like that. So the stimulus itself can create new neurons as needed.

If looking at the neural network model of [7] again—trying to justify everything is silly, but a similar situation that favours a unique set of closer grouped entities might be relevant. It was also shown in the neural network that noisy input could be filtered out more easily, which might also be a helpful feature for the nested ensembles. The noise would be filtered out more easily because it might not be consistently the same in each group, whereas each specific concept would be. Keeping the individual groups separate does not allow random noise to form into more common clusters. The ensembles are then connected through the hierarchy. It has also been suggested that the physical space and the logical space are different. Neurons occupying the same regions can still belong to different hierarchy levels, for example. For a comparison with the neural network, the hidden layer(s) is a combination of the nodes in the level below. For a nested ensemble, this is simply the parent or enclosing region of the group in question. This can continue up to the outer-most region. That would be the largest region in size, but would represent the most global and general concept as well. So the hierarchy is from the smaller nested regions to the larger enclosing one. The idea of a trigger, as shown in the earlier figures [4, 6], is also appropriate and is even represented in Fig. 1 of this paper. It could be the lateral connecting synapse between the two inner circles that might be used to link-up different types of concept into logical sequences.

Also, can Fig. 3 of this paper be compared to the concept trees of other papers [4], Fig. 6, for example? It is shown again here in Fig. 4. Static knowledge also needs to be learned and base nodes at the bottom of trees might provide activation paths to the

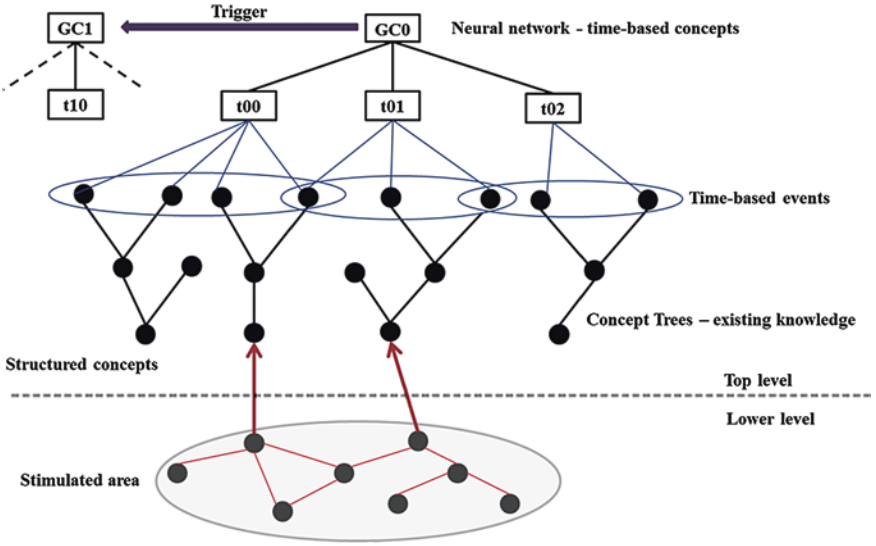


Fig. 4 Integration of elements into the cognitive model, also from 4

groups or concepts at the leaves that then get arranged further through time, at the time-based events layer, for example. The dynamic time-based layer is maybe where tree groups are initially connected-up to form ensembles, but it is better to have 3 concept groups there instead of 1 and have them link-up in another level if required. Maybe that way, the ensemble can still be a more abstract arrangement, while the structured process that builds the concept trees remains as well. So the ensembles are also somewhere along this first time-based line and then up through the whole hierarchy. Figure 3 of this paper maybe has the node structures 180° the wrong way round, where the broader regions should try to connect with each other, if they were part of the time-based events layer of Fig. 4. Figure 3 is just illustrating the point however, where it is easier to connect single nodes than multiple ones. The idea of a construction process in one direction and a search in the opposite direction is also again consistent. We can guess that the construction process for the ensemble hierarchy is again from the static concept trees to the dynamic global concepts. If that is the case, we would in fact see small details first and then aggregate them into larger entities. This is good for another reason. There is then a direct path to these smaller concepts that gets added during their formation and would allow them to be accessed directly, as in memory retrieval. We then perform a search in the opposite direction, from our general impression to the finer details. We find the general region first and then search 'inside' of that for specific information. Or maybe if the search is unclear, a larger area must be activated first, as in browsing. If these end with concept tree-like structures, then the small ensembles can even trace down to the single terminating base nodes that may allow a common connection with other brain regions.

Looking at the actual human brain, the search probably starts in the neocortex [8] which is the thin upper layer that envelops most of the rest of the brain. So that is OK and is a top-down cognitive search. This means that the initial learning process must be more bottom-up and possibly carried out more through observation than prediction or interpretation. But that is probably OK as well. The hippocampus is supposed to be where memories are stored, or at least, is critical to their formation. It is a separate structure to the neocortex, as the concept trees are to the ensemble hierarchy. Synfire chains [13] demonstrate a cascading activation process over a hierarchical structure that could represent a search process over the ensemble hierarchy. This has been pointed out earlier in Sect. 3.2 as opposite to the inward firing process assumed by the nested ensembles, but it may in fact be the same. The inward firing of this paper is from the outer region to the nested regions, which is the same as from the upper hierarchical level to the multiple lower levels. It is just the physical representation that can be confusing. One difference might be that while the outer-most region represents one basic concept, it might contain more neurons numerically. These then excite more concepts in the next level but they are each represented by smaller numbers of neurons. This is interesting in itself, if a neuron must represent something specific, or can it be purely functional? Are some neurons used simply to activate the next level upon request? It might be worth mentioning the pyramidal neurons found by Cajal, for Sect. 5.2 in particular. These have multiple inputs at the base and an output axon with synapses that span any area and so can flow in any direction. They would also be ideal as the base neurons of (concept tree) memory structures.

5.2 Circuit Reinforcement and Balancing

It has been shown previously [2, 3] how ant or termite colonies can collectively determine an optimal route through a basic reinforcement mechanism, without any prior or global knowledge about the route. Each insect leaves a trace that is read by the other insects and it is only that process which determines the optimal route. Figure 3 of his paper shows how a similar process will encourage the closer sets of neurons to form together first, as the overall distances are less. This is also based on local information only and with no prior knowledge. As these insects are believed to work through a nervous system and not a brain as humans have, it is reasonable to apply a similar type of process for the pattern constructions. Research has already shown how firing neuron activity can saturate ([1, 3], for example), but this does not mean a free energy supply. It must also be considered that if a signal is sent from a neuron to more than one place, then it can only be at a fraction of the single signal strength and so it might require a faster firing rate to maintain the same strength to multiple places. Inhibitors probably help to self-regulate the firing rate. The more that feeder neurons activate an area, the more it will send out inhibitors that in turn might slow down the feeder activation signals, until some happy medium is met.

Distance is also important along a single route and must be considered along with the energy consumption and the neuron threshold value. More energy would be required to force a signal along a longer path, where repeated firing by the feeder neurons would probably be required to maintain the signal flow. The paper [4] includes very basic equations that consider disruption of the signal over some distance. There does not have to be a forceful disruption for this aspect, only the natural impedances, but similar types of equation can be applied. These will be described in quite abstract and general terms, so it is the idea of them and not exact values that is important. For a signal to be maintained therefore, we need to consider how much energy might be lost over a particular distance. For example:

- Let T_m be the threshold for the neuron that is to be activated.
- Let I_s be the input signal from the feeder neuron. Consider a single line or path, with just one feeder neuron to the next neuron.
- Let d be the distance from the feeder neuron to the activation neuron.
- Let α be the amount of signal that is lost per unit distance.

Then, even if a neuron can eject the same amount of output that is received as input, the output signal required by the feeder neuron might be:

$$I_{sn} + (d \times \alpha).$$

Or an excess of $(d \times \alpha)$ is required to cover the distance of the signal to the next neuron.

As a neuron can act as a capacitor, this can mean that multiple signals are sent and stored, until the cumulative result fires the activation neuron. If the activation neuron is then required to activate another neuron further along, it faces the same

problem. Thinking again in very abstract terms, the additional required signal becomes a multiple of the requirements of each individual neuron along the path. For example, if neuron 1 needs to fire twice to send enough signal to activate neuron 2 and neuron 2 needs to fire twice to send enough signal to activate neuron 3; then neuron 1 needs to fire 4 times to allow neuron 3 to fire. As far as balancing the neurons' organisation inside of any ensemble is concerned, this is actually a good result. Consider a line of these neurons that span a particular region, where the distances between them varies as follows:

N1 5 **N2** 2 **N3** 2 **N4** 2 **N5** 10 **N6** 10 **N7**

When clustering, it can be a bit localised, where typically the closest distance between any two points is measured. If the neurons N1–N7 represent the points along a line, then neuron N3 would typically be considered to be at the centre of a cluster. If a region spans the whole area from N1 to N7 however, we would prefer any new neurons that get added to be evenly spaced and not to amass around neuron N3, being related with the closest local distances. If we use a cumulative multiplication of the required signal amount across the whole line, then the centre is determined by the distance over the whole region only and not between individual neurons. In the example, if each numerical value is the additional required signal to reach the next neuron, then a value of 2 is required to go from N3 to N4, for example. But N4 then also needs 2 to go to N5, and so on. We are also trying to minimise the amount of energy that is required and also allow synapses to travel in both directions from a single neuron output. With a cumulative multiplicative count, a cluster centre like N3 would travel to N1 in one direction and N7 in the other, requires $(2 \times 5) + (2 \times 2 \times 10 \times 10) = 410$ additional firings. Neuron N5 looks like it is at the edge of a cluster, but to span the same distance, it requires $(5 \times 2 \times 2 \times 2) + (10 \times 10) = 140$ additional firings. The larger distances are prohibitive for neurons not located in the centre, distance-wise. So if this region was excited from the centre, which appears to be the most economic, it would prefer neuron N5 over neuron N3. This might also help to space the creation of new neurons better, because the energy or stimulus is always located in the centre and not necessarily in the densest region. There is also a stability or integrity reason why a central activation is better. If the activation path to the ensemble was at the edge and not at the centre, then it might be easier to change the ensemble concept by adding neurons to the other side of the edge. If the activation path is to the centre, then even if something new gets added, the original concept can maintain its original meaning, where the change is an addition rather than a radical shift to something else.

5.3 One-Way System?

With clear input and output sets to neurons and the requirements to complete circuits, it may be thought that most processes are one-way systems that are cyclic in nature. Although, recent brain models show fibrous or tree-like branching

properties, at some level of granularity. Cyclic or circular completions are very easy to understand, but assume that the whole process is 'as one', where it might then be further assumed that the signals that flow need to be very similar. While they match with the known neuron functionality, there may be other problems. For one thing, there are definite regions in the brain and so signals would need to cross boundaries and even functionality type. Similarly, if the signal flows in one direction, for example, from the top of the hierarchy to the bottom in the neocortex (or the simulated neural network model), then to complete the cycle it needs to flow back up again, which is again possibly changing the functionality. It is noted that a cyclic completion can be more at the end of an activation chain than its full length and therefore another suggestion can also be made. An alternative way to complete a circuit is to have (at least) two halves that meet or join. This is attractive for a number of reasons. One is the distance reason again. It would be possible for any brain region to start sending a signal into the 'middle area' of the brain, for example and have it travel anywhere else. It might then meet a matching signal from the receiving area. The two can join to reinforce and complete a slightly different mode of circuit. This could even be at the other side of the brain and so there does not have to be a long feedback loop to the original source. This also means that when a stimulus is set off, the neocortex or some other area can independently start working to satisfy it and different regions can work in parallel. It can run through its own structures to try to match the signal from other places. The signals that meet might then actually collide to join up, instead of flow in the same direction together, but they still complete the circuit and the interaction might even facilitate the essential ingredients of resonance [1, 4, 5], if the firing rates or signals strengths also match. The signals can maybe meet at the region boundaries, such as somewhere around the concept trees layer in the computer-based model.

It is still possible to build a similar system with the existing components that would incorporate the cyclic reinforcement more. The neurons can still have only one input and one output function, but some can face each other and then be joined by synapses that fuse with another set of neurons running sideways. The opposing signals induce the lateral activity that can flow through the sideways facing neurons and even re-connect to complete some type of circuit. Figure 5 shows this schematically and might be compared with the ideas of pressure in [4]. Imagine that the two brain regions are at the top and bottom of this figure. They send signals (external activation) from their own directions to excite the top and bottom neurons respectively. These have joins or links in the form of a synapse junction that includes sideways or lateral connections. The signal is forced through the lateral connections to activate the sideways facing neurons. They might even loop back to reinforce the signal, but only very locally, or in the specific region that represents the desired search result and neural ensemble. There could be less force if the top-bottom facing neurons actually joined at the lateral neurons instead of at the junction, but their creation or initial meeting might be from a straight connection between them, which is an easier automatic join. So the whole area, including the lateral connections, might grow in a normal manner, with new neurons or synapses being added to places that are more frequently used.

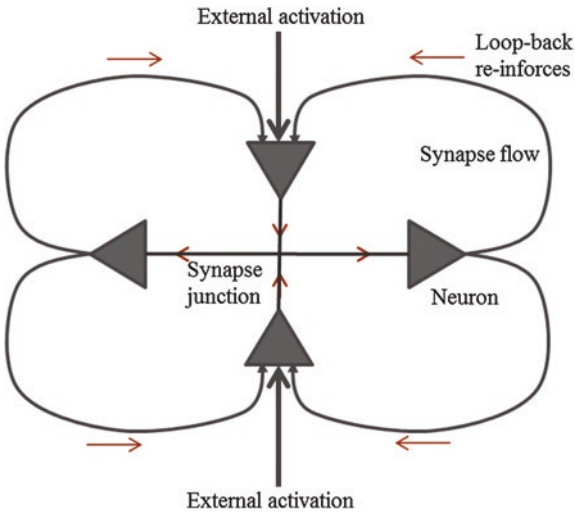


Fig. 5 Top and bottom neurons join and activate lateral ones through synapse junctions, to register enhanced signals

6 Conclusions

The purpose of this paper is to show how nested, or more specific patterns, may become the main focus in a generally excited area. They might even be used as part of more complex mathematical thought processes. Rather than the exact details of how they might be created, or link to each other, etc., the paper describes how they might be useful as a simplified design. Simulation would be easier if the interaction between the patterns only was considered, using a general equation for their relative strengths and ignore exact synaptic connections. The mechanical processes can work with a minimum of complexity and would allow these patterns to form and fire in sequence. It would also realise some level of natural order, which would be better than the very random and chaotic structures that appear to be present. It is interesting that a self-organising process might naturally prefer a nested structure, certainly to one that faces outwards.

The second purpose of this paper is to integrate the new findings into the whole cognitive model. It appears that the nested ensembles fit-in almost seamlessly with the existing ideas and through studying them, other helpful information has been obtained. In particular, there are several examples of how the processes can naturally regulate themselves and perform the type of functionality that you might think requires some level of intelligence. With respect to automatic processes, the nesting allows for the idea of terminal or end states, which can help with search processes. The act of searching into a smaller region as opposed to a larger region might also make the search process easier. Even just the signal strength can help with managing pattern transitions as part of an automatic process, so there is quite a lot that can be achieved with the basic components that are known about. As described in Sect. 5, the mechanism is still compatible with earlier work.

The self-organisation can also space itself through automatic processes. This might allow new neurons to be added in a more even manner and could help to maintain the concept integrity. A link at creation-time, to the centre of the concept can also help with finding it directly, possibly as part of a memory structure. The idea of resonance being important is also enhanced, if some form of joining is to be preferred over the less violent reinforcement through cyclic links, or complementary with it. But then, each brain area can keep its own functionality and have a sort of interface. Also, the earlier ideas of the dynamic hierarchical network joining with the static knowledge-based one is still central to the whole architecture and even small pieces of evidence from the real biological world can help to support the ideas, where established theories are not so clear.

References

1. Carpenter, G., Grossberg, S., Rosen, D.: Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Netw. **4**, 759–771 (1991)
2. Dorigo, M., Bonabeau, E., Theraulaz, G.: Ant algorithms and stigmergy. Future Gener. Comput. Syst. **16**, 851–871 (2000)
3. Grassé, P.P.: La reconstruction dun id et les coordinations internidividuelles chez Bellicositermes natalensis et Cubitermes sp. La théorie de la stigmergie: essais d'interprétation du comportment des termites constructeurs. Insectes Sociaux **6**, 41–84 (1959)
4. Greer, K.: New ideas for brain modelling. In: Iantovics, B.L., Zamfirescu, C.B., Revett, K., Gligor, A. (eds.) Special Issue on: From Natural Computing to Self-organizing Intelligent Complex Systems. IGI Global (2014). Published on arXiv at http://arxiv.org/abs/1403.1080, accepted for publication by J. Inf. Technol. Res. (JITR)
5. Greer, K.: Artificial neuron modelling based on wave shape, brain. Broad Res. Artif. Intell. Neurosci. **4**(1–4), 20–25 (2013). ISSN 2067-3957 (online), ISSN 2068-0473 (print)
6. Greer, K.: Turing: then, now and still key. In: X-S. Yang (ed.) Artificial Intelligence, Evolutionary Computation and Metaheuristics (AIECM)—Turing 2012. Studies in Computational Intelligence, vol. 427, pp. 43–62. Springer, Berlin (2013). doi:10.1007/978-3-642-29694-9_3
7. Greer, K.: Symbolic neural networks for clustering higher-level concepts. NAUN Int. J. Comput. **5**(3), 378–386, extended version of the WSEAS/EUROPMENT International Conference on Computers and Computing (ICCC'11) (2011)
8. Hawkins, J. and Blakeslee, S.: On Intelligence. Times Books, NewYork (2004)
9. Hebb, D.O.: The Organisation of Behaviour. Wiley, New York (1949)
10. Hill, S.L., Wang, Y., Riachi, I., Schürmann, F. and Markram, H.: Statistical connectivity provides a sufficient foundation for specific functional connectivity in neocortical neural microcircuits. Proc. Nat. Acad. Sci. **109**(42), E2885 (2012)
11. Kandel, E.R.: The molecular biology of memory storage: a dialogue between genes and synapses. Sci. Mag. **294**(5544), 1030–1038 (2001)
12. Rosin, D.P., Rontani, D., Gauthier, D.J. and Scholl, E.: Control of synchronization patterns in neural-like Boolean networks, ArX. Prepr. Repos. (2013). http://arxiv.org
13. Vogels, T.P., Kanaka Rajan, K., Abbott, L.F.: Neural network dynamics. Annu. Rev. Neurosci. **28**, 357–376 (2005)
14. Waxman, S.G.: Sodium channels, the electrogenisome and the electrogenistat: lessons and questions from the clinic. J. Physiol. 2601–2612 (2012)
15. Willie, C.K., Macleod, D.B., Shaw, A.D., Smith, K.J., Tzeng, Y.C., Eves, N.D., Ikeda, K., Graham, J., Lewis, N.C., Day, T.A., Ainslie, P.N.: Regional brain blood flow in man during acute changes in arterial blood gases. J. Physiol. **590**(14), 3261–3275 (2012)

Neural-like Growing Networks the Artificial Intelligence Basic Structure

Vitaliy Yashchenko

Abstract The article considers a new type of neural networks created by analogy with the neural network of the human brain and the theory of artificial intelligence which is based on these kind of networks. In this theory allocated three main functional units of integrative activity of the "brain" of intelligent robots: sensory systems; modulatory systems; motor systems. Here we consider the human sensory visual system and its model in intelligent systems. In this article are considered hypothesis about transformation the recognizable images to the same size in the fovea and image recognition on a subconscious level. Hypothesis realized in a hardware neural sensory model of a human sensory organ of visual system. As well, one can find the results of recognition of damaged images. In conclusion considers the block diagram of intelligent computer (electronic brain), which is represented as homogeneous, multiply connected, multidimensional, associative, active, growing neural matrix environment.

Keywords Multiply connected neural-like growing networks · Multiply connected multidimensional neural-like growing networks · Multiply connected receptor-effector neural-like growing network · Multiply connected multidimensional receptor-effector neural-like growing network · The theory of artificial intelligence · Sensor system · A block diagram of intelligent computer

1 Introduction

Large human brain research projects aiming to develop its model are currently conducted around the globe. Back in 1990 we set ourselves a task of developing neural networks analogous to biological ones, i.e. human brain neural networks.

V. Yashchenko (✉)
Institute of Mathematical Machines and System Problems NANU, Kiev, Ukraine
e-mail: vitaly.yashchenko@gmail.com

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_3

Such networks would allow an intellectual system or a robot to adapt to the surrounding situation, acquire knowledge and make decisions leading to the objectives fulfillment. In 1990–1995s, as a result of the functional analysis of semantic networks, artificial neural networks, biological neural networks and human brain structure a new type of neural networks was synthesized—multiply connected neural-like growing networks.

A number of pattern recognition models and a virtual model of the intellectual robot VITROM were created on the basis of multiply connected neural-like growing networks in 1999. In 2000 this model was demonstrated at the CeBIT exhibition in Hannover, and in 2002 at the exhibition in Beijing (The 5th China Beijing International High-tech Expo). Further studies resulted in the creation of the artificial intelligence theory. Multiply connected multidimensional receptor-effector neural-like growing networks, which are traced back to their functional prototype—biological neural networks, are the base universal structure of the artificial intelligence theory. At present, a study of the structural and functional schemes of the hardware implementation of the new type intellectual systems with the functionality approaching that of the human brain has been carried out.

2 Multiply Connected Neural-like Growing Networks

The new type of neural networks comprise: multiply connected neural-like growing networks; multiply connected multidimensional neural-like growing networks; multiply connected receptor-effector neural-like growing networks; multiply connected multidimensional receptor-effector neural-like growing networks [1–4]. Let us consider some of them. Neural-like growing networks are formally defined the following way.

2.1 Multiply Connected Neural-like (One-Dimensional) Growing Networks (mn-GN)

$\boldsymbol{S} = (\boldsymbol{R}, \boldsymbol{A}, \boldsymbol{D}, \boldsymbol{M}, \boldsymbol{P}, N)$, where $R = \{r_i\}$, $i = \overline{\mathbf{1}, n}$ is a finite set of receptors; $A = \{a_i\}$, $\overline{1, k}$—finite set of neural-like elements; $D = \{d_i\}$, $i = \overline{1, e}$—finite set of arcs connecting receptors with the neural-like elements and the neural-like elements with each other; $P = \{P_i\}$, $i = \overline{1, k}$ $N = h$, where P—excitation threshold of the node a_i, $P = f(m_i) > P_0$ (P_0—minimum allowed excitation threshold) provided that the set of arcs $\boldsymbol{D}$, associated with the node a_i, is correspondent to the set of weights $M = \{m_i\}$, $i = \overline{1, w}$, and m_i can take both positive and negative values (Fig. 1).

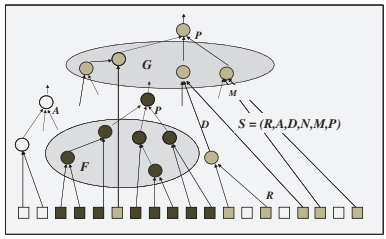


Fig. 1 Multi-connected neural-like growing network

2.2 Multiply Connected Receptor-Effector Neural-like Growing Networks (mrenGN)

$\boldsymbol{S} = (\boldsymbol{R}, \boldsymbol{A_r}, \boldsymbol{D_r}, \boldsymbol{P_r}, \boldsymbol{N_r}, \boldsymbol{E}, \boldsymbol{A_e}, \boldsymbol{D_e}, \boldsymbol{P_e}, \boldsymbol{M_e}, \boldsymbol{N_e})$, $R = \{r_i\}$, $i = \overline{1,n}$—a finite set of receptors, $A = \{a_i\}$, $\overline{1,k}$—a finite set of neural-like elements of the receptor area, $D_r = \{d_i\}$, $i = \overline{1,e}$—a finite set of arcs of the receptor area, $E = \{e_i\}$, $i = \overline{1,e}$—a finite set of effectors, $A_e = \{a_i\}$, $\overline{1,k}$—a finite set of neural-like elements of the effector area, $D_e = \{d_i\}$, $i = \overline{1,e}$—a finite set of arcs of the effector area, $P_r = \{P_i\}$, $P = \{P_i\}$, $i = \overline{1,k}$, where P_i—excitation threshold of the node a_{ir}, a_{ie} $P_i = f(m_i)$ provided that the set of arcs $\boldsymbol{D_r}$, $\boldsymbol{D_e}$, associated with the node a_{ir}, a_{ie}, is correspondent to the set of weights $M_r = \{m_i\}$, $M = \{m_i\}$, $i = \overline{1,w}$, and m_i can take both positive and negative values. $\boldsymbol{N_r}$, $\boldsymbol{N_e}$—connectivity variables of the receptor and effector areas.

2.3 Multiply Connected Multidimensional Neural-like Growing Networks (MmnGN)

$\boldsymbol{S} = (\boldsymbol{R}, \boldsymbol{A}, \boldsymbol{D}, \boldsymbol{P}, \boldsymbol{N})$, and $\boldsymbol{R} \supset \boldsymbol{Rl}, \boldsymbol{Rr}, \boldsymbol{Rv}$; $\boldsymbol{A} \supset \boldsymbol{Al}, \boldsymbol{Ar}, \boldsymbol{Av}$; $\boldsymbol{D} \supset \boldsymbol{Dl}, \boldsymbol{Dr}, \boldsymbol{Dv}$; $\boldsymbol{P} \supset \boldsymbol{Pl}, \boldsymbol{Pr}, \boldsymbol{Pv}$, where ***Rl***, ***Rr***, ***Rv*** is a finite subset of receptors; ***Al***, ***Ar***, ***Av***—a finite subset of neural-like elements; ***Dl***, ***Dr***, ***Dv***—a finite subset of arcs; ***Pl***, ***Pr***, ***Pv***—a finite subset of excitatory thresholds of the neural-like elements belonging, for example, to the linguistic, speech or visual informational dimension; ***N***—a finite set of connectivity variables.

2.4 Multiply Connected Multidimensional Receptor-Effector Neural-like Growing Networks (mmrenGN)

$\boldsymbol{S} = (\boldsymbol{R}, \boldsymbol{A_r}, \boldsymbol{D_r}, \boldsymbol{P_r}, \boldsymbol{M_r}, \boldsymbol{N_r}, \boldsymbol{E}, \boldsymbol{A_e}, \boldsymbol{D_e}, \boldsymbol{P_e}, \boldsymbol{M_e}, \boldsymbol{N_e})$; where $\boldsymbol{R} \supset \boldsymbol{Rv}, \boldsymbol{Rs}, \boldsymbol{Rt}$; $\boldsymbol{A_r} \supset \boldsymbol{Av}, \boldsymbol{As}, \boldsymbol{At}$; $\boldsymbol{D_r} \supset \boldsymbol{Dv}, \boldsymbol{Ds}, \boldsymbol{Dt}$; $\boldsymbol{P_r} \supset \boldsymbol{Pv}, \boldsymbol{Ps}, \boldsymbol{Pt}$; $\boldsymbol{M_r} \supset \boldsymbol{Mv}, \boldsymbol{Ms}, \boldsymbol{Mt}$; $\boldsymbol{N_r} \supset \boldsymbol{Nv}, \boldsymbol{Ns}, \boldsymbol{Nt}$; $\boldsymbol{E} \supset \boldsymbol{Er}, \boldsymbol{Ed}, \boldsymbol{Ed}$; $\boldsymbol{A_e} \supset \boldsymbol{Ar}, \boldsymbol{Ad1}, \boldsymbol{Ad2}$; $\boldsymbol{D_e} \supset \boldsymbol{Dr}, \boldsymbol{Dd1}, \boldsymbol{Dd2}$; $\boldsymbol{P_e} \supset \boldsymbol{Pr}, \boldsymbol{Pd1}, \boldsymbol{Pd2}$; $\boldsymbol{M_e} \supset \boldsymbol{Mr}, \boldsymbol{Md1}, \boldsymbol{Md2}$; $\boldsymbol{N_e} \supset \boldsymbol{Nr}, \boldsymbol{Nd1}, \boldsymbol{Nd2}$; here ***Rv***, ***Rs***, ***Rt*** is a finite subset of

receptors, ***Av***, ***As***, ***At***—finite subset of neural-like elements, ***Dv***, ***Ds***, ***Dt***—finite subset of arcs, Pv, Ps, Pt—finite subset of excitatory thresholds of the neural-like elements of the receptor area belonging, for example, to the visual, acoustic or tactile informational dimensions, ***N***—finite set of connectivity variables of the receptor area, ***Er***, ***Ed1***, ***Ed2***—finite subset of effectors, ***Ar***, ***Ad1***, ***Ad2***—finite subset of neural-like elements, ***Dr***, ***Dd1***, ***Dd2***—finite subset of arcs of the effector area, ***Pr***, ***Pd1***, ***Pd2***—finite set of excitatory thresholds of the neural-like elements of the effector area belonging, for example, to the speech informational dimension and the action dimension. ***N***—finite set of connectivity variables in the effector area (Fig. 2).

Neural-like growing networks are a dynamic structure that changes depending on the value and the time the information gets to the receptors, as well as on the previous state of the network. The information about the objects is presented as the ensembles of excited nodes and the connections between them. Memorization of objects and situations descriptions is accompanied by the addition of new nodes and arcs to the network when a group of receptors and neural-like elements enter into a state of excitement. The excitation is distributed wavily around the network.

Receptor-effector neural-like growing networks are a dynamic structure, which changes depending on the external information coming into the receptor field and the information generated by the effector area and transferred to the outside world.

Memorization of the external information is accompanied by the addition of new nodes and arcs to the network in the receptor zone, while the generation of information and its transfer to the outside world is accompanied by the addition of new nodes and arcs to the network in the effector area. The formation of new nodes and arcs is accompanied by the transition of an ensemble of receptors or neural-like elements, or receptors and neural-like elements of the receptor area and an ensemble of effectors or neural-like elements, or effectors and neural-like elements of the effector area into an excited state. The excitation is distributed wavily around the network.

In mrenGN the information about the outside world, its objects, their states and situations describing the relationship between them, as well as the information on the actions caused by these states is saved being reflected in the network structure, and the acquisition of the new information initiates the formation of new associative nodes and links and their redistribution between the nodes that have arisen earlier, the common parts of these descriptions and actions are generalized

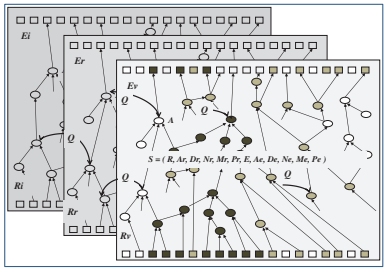


Fig. 2 Topological structure of multidimensional receptor-effector neural-like growing network

Table 1 The general comparison of neural-like growing networks and neural networks

Neural-like growing networks	Neural networks
Neural-like element. A computing device with memory	**Neural element**. Threshold element
An arbitrary input function is defined	A non-linearly processed weighted sum of inputs is defined
The network architecture is formed with the information coming to receptors. The number of links corresponds to the number of features	The network architecture is defined by its type. The number of links is redundant. Special methods of links sifting are required
Connectivity variable. Allows you to control the ratio: number of links/number of nodes in the network	**Connectivity variable**. No
Reconfigurable structure	**Fixed structure**
Multilevel structure. The number of levels (layers) is arbitrary, is generated automatically according to the input information	**Up to 3 levels (layers) are used normally**. Using more than 3 layers is not meaningful. Levels (layers) are used to increase the network capacity
Receptor fields, effector fields, receptor and effector areas. Yes	No
Various information spaces. Video, sound, etc.	No
Learning rate microseconds. Learning during the system life cycle	**Learning rate** from seconds to many hours depending on the amount of training set
Network capacity 100 %	**Network capacity** 20–30 %
Parallelism. Calculations along the activity branches are performed simultaneously in all the neural-like elements. The relatively high network performance speed increases along with the increase of the amount of information	**Parallelism**. Calculations are performed for each neural element successively. The network performance speed decreases with the increase of the amount of information

and classified automatically. Multiply multidimensional receptor-effector neural-like growing networks are designed to memorize and process the descriptions of images of objects or situations in the problem area and to generate the control actions with the help of various informational spatial representations, such as tactile, visual, acoustic, taste ones etc.

The general comparison of neural-like growing networks and neural networks is shown in Table 1.

3 Natural Intelligence Formation System

The core of human intelligence is the brain consisting of multiple neurons interconnected by synapses. Interacting with each other through these connections, neurons create complex electric impulses, which control the functioning of the whole organism and allow information recognition, learning, reasoning and structuring through its analysis, classification, location of connections, patterns and

distinctions in it, associations with similar information pieces etc. The functional organization of the brain. In the works of physiologists P.K. Anohin, A.R. Luriya, E.N. Sokolov [5] and others the functional organization of the brain includes different systems and subsystems. The classical interpretation of the interactive activity of the brain can be represented by interactions of three basic functional units:

1. information input and processing unit—sensory systems (analyzers);
2. modulating, nervous system activating unit—modulatory systems (limbic-reticular systems) of the brain;
3. programming, activating and behavioral acts controlling unit—motor systems (motion analyzer).

Due to the scope limitation of the present work we consider only the human intelligence sensory system and its models for neural-like growing networks.

Brain sensory systems (analyzers). Sensory (afferent) system is activated when a certain event in the environment affects the receptor. Inside each receptor the physical factor affecting it (light, sound, heat, pressure) is converted into an action potential, nervous impulse. Analyzer is a hierarchically structured multidimensional system. Receptor surface serves as a base of the analyzer, and cortex projection areas as its node. Each level is a set of cells, whose axons extend to the next level. Coordination between sequential layers of analyzers is organized based on divergence/convergence principle. It is known that 90 % of the perceived information comes through the human visual system, and it is quite important for us to know how the natural intelligence (human brain) perceives and processes information. Eye is a sensory organ of the human visual system, which consists of the eyeball and the optic nerve (Fig. 3).

The eyeball has a diameter of about 24 mm and is about the shape of a regular sphere but with a slightly convex front side. There are six muscles around each eyeball. The external and internal rectus muscles control the movement of the eye

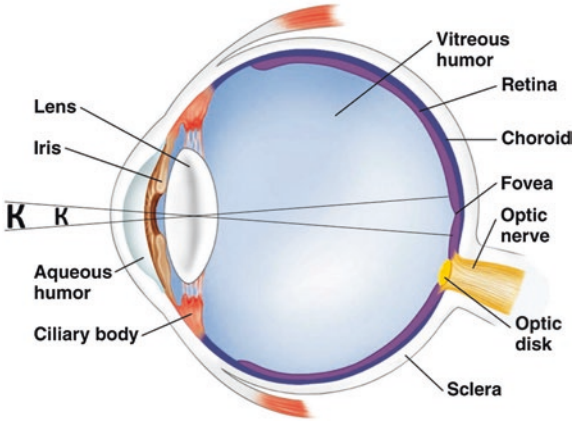


Fig. 3 Eye

to the left and to the right; the superior and inferior rectus muscles—up and down; oblique muscles roll the eyeball. But the work of the extraocular muscles does not end here. Over 30 years of studies by W. Bates show that the eye rectus muscles can shorten the eyeball along the eye optic axis, thus bringing the eye lens to the retina and the oblique muscles can squeeze the eyeball and push the eye lens from the retina. Sclera is the outer layer of the eyeball. The front part of the sclera is continuous with the transparent circle, the cornea, which acts as a converging lens with a radius of curvature of 6.82 mm.

Iris is located between the cornea and the lens. Iris regulates the intensivity of the light in such a way that the amount of light the eye receptors get is almost constant (in bright light the iris expands narrowing the pupil, in low light conditions it narrows and the pupil expands). An image is projected on the retina through the eye lens. Due to the accommodation—lens elasticity—and the eye muscles action the lens takes the shape that ensures a sharp image projection getting on the light-sensitive transducers (receptors) located in the retina. The focal length of the eye of an adult may vary from 18.7 to 20.7 mm, which enables focusing on both distant and near objects. Retina is composed of many individual elements—receptors. There are two types of the light-sensitive elements in the human eye—rods and cones. Rods and cones are not evenly distributed in the retina. There is a part in the centre of the retina that contains only cones. This area is called fovea centralis or simply fovea (Latin. fovea—pit), it provides high visual acuity. Foveola ('a small pit'—approx. 1° in diameter) located in the centre of fovea is a part with even better perception capability. The resolving power of the eye has its maximum value here, and the fovea cones are directly connected with the higher nerve centres, while the majority of the retina receptors does not have a 'direct' connection to the brain. It can be assumed that the visual information extraction is done mainly by that part of the receptors that lie in the fovea in the middle of the retina.

Hypothesis on adjusting the images being recognized to the same size in the fovea. Given that the 'human eye' system contains a small area (the central pit, fovea) where the eye visual acuity is the highest and the fovea cones are directly connected to the higher nerve centres, it can be assumed that the object being recognized (the object in the focus of attention, being looked at), e.g. 'k' or 'K', is scanned by the eye movements carrying out a systematic selection of the information about the shape, position and size of the object, is projected onto the fovea taking into account these parameters, and eventually the analysis, synthesis and comparison with the previously memorized objects on the level of excitation of neurons reflecting the attributes and properties of these objects is conducted in the higher brain divisions. This applies to objects of different sizes which are placed at the same distance from the eye. If the object is large and its projection goes beyond the fovea, the part exceeding the bounds of the fovea will not be clearly visible. This does not happen when the eye functions normally, thus the complete projection onto the fovea must be achieved by changing the focal length. And it

may be not the size of the objects projections onto the fovea area that determines their size but the level of excitation of the corresponding command neurons of eye muscles that control the shape of the lens,[1] or, according to other sources, shape of the eyeball,[2] which leads to a change in the focal length of the lens in the first case and to a change in the focal length of the eye system in the second one. The existence of the foveola enhances this function, especially for the formation of a unified view when looking at large objects, and perhaps sheds light on the phenomenon of speed reading and memorizing large amounts of information. In the cases of memorizing hundreds of pages of text in the process of turning the pages of the book the information may get to the fovea page by page at a conscious level and be stored, and the text recognition may occur at a subconscious level while scanning the information that was memorized by the foveola.

Hypothesis on the image recognition at a subconscious level. Given that the 'human eye' system contains a small area (the central pit, fovea), and there is a part with even better perception capability in the centre of the fovea—foveola, it can be assumed that the foveola is used for a detailed scanning of the image projected onto the fovea, which is followed by its recognition at a subconscious level. The information processing is conducted in the retina on six levels. In general, with some simplifications, the role of each level is clear. The first and the second levels—pigment epithelium, rods and cones—receptors of video information perception. The third level—horizontal cells that stop signal scattering in the retina by lateral inhibition in the surrounding areas. It is important for the clear highlighting of the contrasting borders in the visual image. The fourth level—amacrine cells interconnected with bipolar cells, ganglion cells and with each other. The number of these cells is particularly high in animals with high visual acuity, such as birds. The third and the fourth levels are used to increase the sensitivity/detailization correlation. The more receptors are attached to one ganglion cell, the more sensitive the system is, i.e. the weaker light signals it can detect. The more receptors will be connected to a ganglion cell, the stronger the signal at the output of the cell (visual acuity) will be. It is obvious that the increase in sensitivity leads to the decrease in detailization. Thus, the third and the fourth levels are responsible for sensitivity, sharpness and contrast of the perceived image. The fifth level—bipolar cells are the cells of the retina intermediate layer that transmit nerve impulses from receptors to ganglion cells. The sixth level—ganglion cells are tightly packed in the retina, and their dendritic fields overlap. From one to one hundred bipolar neurons can converge to one ganglion cell. From a few to tens of thousands of photoreceptors (rods and cones) can be connected to one ganglion cell through the bipolar neurons. Ganglion cells complete the 'three-neuron receptor-conducting retina system': photoreceptor—bipolar neuron—ganglion cell.

[1] According to H. Helmholtz's theory, when looking at objects at different distances, the lens optical characteristics are changed by the ciliary muscle or, according to some ophthalmologists, by the intraocular fluid movement, which does not make any difference to us, as both result in the change in the focal length.

[2] American ophthalmologist W. Bates at the turn of 19th and 20th centuries discovered that the image is constructed in the human eye by the adjustment of the length of the eye.

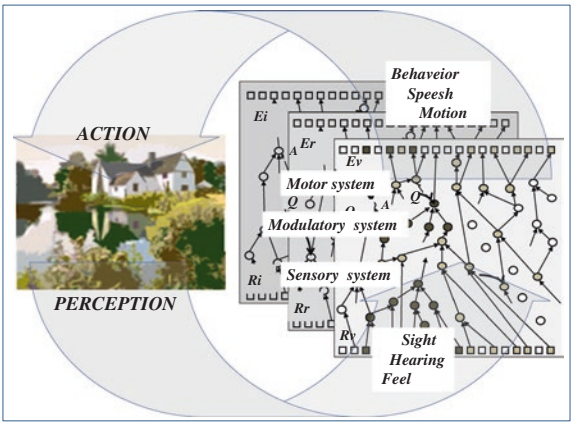


Fig. 4 Artificial intelligence system modeling scheme for multidimensional ren-GN

4 Functional Organization of the Artificial Intelligence Systems

Functional organization of the artificial intelligence system can be also represented by the interaction of three functional blocks. A schematic representation of the 'brain' of the artificial intelligence system shown in Fig. 4 consists of three functional areas of the multidimensional ren-GN.

1-sensory systems (analyzers): sensory information reception and processing area. 2-modulatory systems: modulation and system activation area and 3-motor systems: behavioral acts activation and control area.

4.1 Hardware-Neural Model of Human Visual Sensory Organ

In the hardware-neural model of human visual sensory organ some functions (levels 1–2) of the human visual system, as described in chapter "New Ideas for Brain Modelling 2", are performed by the video cameras. Third–sixth (3–6) levels are modeled by the neural network. An object image obtained from a video camera is converted into a contour image, which significantly reduces the amount of information required for classification.

In order to detect an image contour, similar to the human sensory system, information processing is performed through a neural network. Such network consists of three types of cells, which are analogous to the horizontal 1, bipolar 2 and ganglion 3 cells (Figs. 5 and 6). In case of the hardware implementation of this method a parallel processing of information in all the network occurs. The expected result is the system performance speed increase by several orders of magnitude.

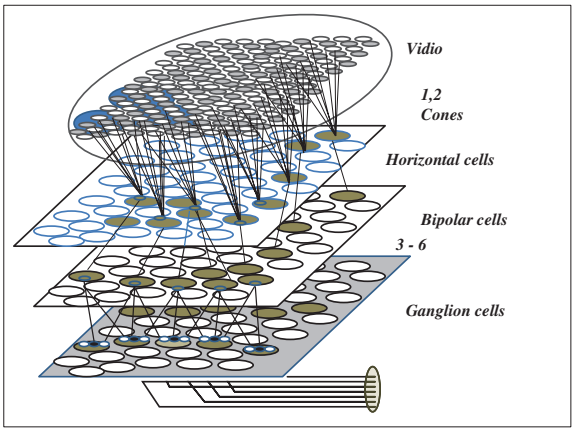


Fig. 5 Model of human visual sensory organ

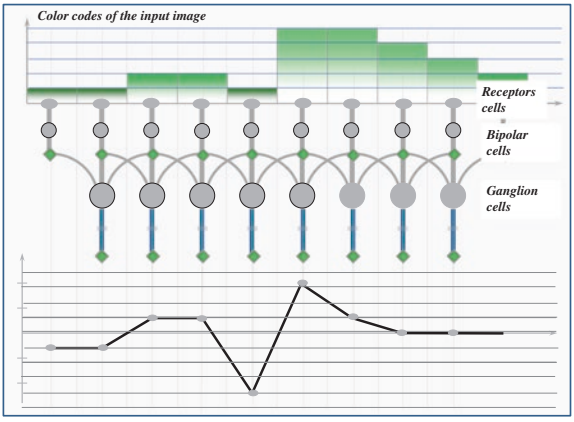


Fig. 6 Neural network for an image contour detection

Figure 6 shows a simplified scheme of a neural network of a sensor system model. Signals from a respective receptor or groups of receptors get to the dendrites of bipolar and ganglion cells through horizontal cells (not shown in the scheme), are amplified by the positive weight of the central dendrite, and the signals corresponding to the neighboring image points are slowed down by the negative weights of the lateral dendrites. When signals enter the ganglion cell, they are multiplied by the corresponding weights, are summed up and are transferred to the output. At the output of the neural network there is a sequence of numbers that by a special rule correspond to the values of codes of colours of the input image points. Data obtained at the output are presented in the graph Fig. 6. The graph shows sharp differences in the obtained function coinciding with the sharp changes in colour. Having fixed these differences we compare them with the predetermined threshold of neuronal excitation. Exceeding the threshold value indicates the presence of the point belonging to the image contour. Upon the scanning completion an internal representation—an image feature vector—is generated at

the output. Thus, each point or group of points of an image perceived by an eye corresponds to one neuron (ganglion cell). Each such neuron has several dendrites, which are connected with the neighboring neurons through the bipolar cells with the use of the negative connections. The level of excitation of neurons actually corresponds to the quality of the contour.

The system has a flexible contour detection settings adjustment mechanism: weighting matrix resizing; sensitivity—detailization function adjustment (analogue to the third and fourth levels of the human eye retina); flexible threshold value adjustment; different methods of image scanning combined; system settings saving.

The resulting contour is converted into an image feature vector. This stage is followed by the problem of image recognition and memorization. Pattern matching is the most likely model of image recognition in the higher layers of the human brain. Unknown images are stored as patterns. Recognition is performed by comparing an external image with a set of internal patterns—image feature vectors.

Hypothesis on the mechanism of pattern recognition in the upper layers of the neocortex in the human brain. Pattern matching model is the mechanism of pattern recognition in the upper layers of the neocortex in the human brain. The advantage of this model is that an external image is compared with all the patterns simultaneously.

In the process of comparison some of the patterns get activated simultaneously and the one showing the most active reaction to the image is the target object. At the same time, the model has a significant drawback: the recognition will not occur in case of a change in lighting, orientation or size of the external image regarding the pattern. Pattern is the internal representation of an image of the object being recognised, previously stored in different positions.

Consequently, if to fit an image of an object to the standard lighting and size and precisely align it with the pattern, the model works perfectly. Taking into account that in the process of visual perception and focusing on an object the video information is projected onto the fovea and the image is fit to the standard lighting and standard size (fovea), and in addition, the image is scanned in order to identify the most informative areas and perhaps fix and compare the distances between them, we can be certain that the human visual system uses a pattern matching model.

Thus, in my opinion, a pattern matching method is actively used by the human visual system. The information processing in the layers of the neocortex is not limited just to the process of matching the object to the pattern, of course. This process is much more complicated. It comprises the information processing, analysis, synthesis and comparison at various levels of biological neural network, in our case at the level of the neural-like receptor-effector growing network.

The model of the human visual system sensory organ, together with the methodology of data processing in receptor-effector neural-like growing networks, is taken as a basis for the virtual robot VITROM project design. VITROM has been in operation for many years and has proven the effectiveness of this approach.

The knowledge base of the robot contains more than 1,000 images including the images of faces from Yale FaceIMAGES_Data, Base Yale FaceIMAGES_NOTT, Indian Face Database, robot's routes through the streets and various images of animals, numbers, letters, etc. (Fig. 7).

The structure of mmren-GN is made up of a homogeneous array of neural-like elements. This allows parallel information processing, as well as the new possibilities of associative information search. Associativity of neural-like growing structures compares favorably with other types of technical devices due to the possibility of information recovery using any part of it. It largely determines the survivability of the systems created on the mren-GN basis, as in case of a breakdown of a considerable part of its elements it retains its objective functions.

Figure 8 shows the results of the corrupted images recognition, which corresponds to 25, 50 and 75 % of the actual damage of the receptor field or area. The increase of the damaged area reduces the confidence of image recognition, and in case of 75 % of damage single recognition errors occur.

Thus, the neural-like growing structures provide high reliability and survivability of systems. Experiments carried out on models of intelligent systems show that a failure of a large number of neural-like growing network elements does not result in the entire structure breakdown.

Fig. 7 Images from the robot's knowledge base

Fig. 8 Results of corrupted images recognition

5 Conclusion

Multiply connected multidimensional receptor-effector neural-like growing networks are used as a basic structure for the implementation of intelligent systems. The hardware implementation of mmrenGN is a multidimensional associative active (memory) structure, which consumes much less energy compared to the high-power computing systems used for the production of modern artificial intelligence systems. New technologies will enable the creation of the structure of the human brain size. The intelligent computer created on the basis of such a structure will differ fundamentally from computers with the von Neumann architecture. This computer will have no arithmetic logic unit, address memory or software. A new generation of intelligent computers will not need an army of programmers. These computers will learn on their own or together with a user. A computer will be a powerful amplifier of human intelligence.

The structural scheme of the electronic brain of robots is a homogeneous, multiply connected, multidimensional, associative, active, neural-like growing matrix structure. It consists of information perception blocks (1); a set of homogeneous, multiply connected, multidimensional, associative, active, neural-like temporary and long-term memory sensory matrices (2); modulation matrix (3); a set of homogeneous, multiply connected, multidimensional, associative, active, neural-like, activity memory motor matrices (4); actuators (5) (Fig. 9). The structure

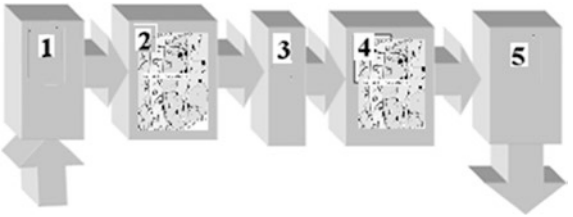


Fig. 9 Structural scheme of the intelligent computer

homogeneity and non-standard (non-von-Neumann) information processing are the strong point of neural-like growing networks. Information is processed, analyzed and stored not only in individual cells (neural-like elements) but also in the distribution of connections between them, in their power, so the state of each neural-like element is determined by the state of many other connected items. The loss of one or a few links does not have a significant influence on the result of the work of the whole system, which ensures its high reliability. High 'natural' interference immunity and functional reliability refer to both distorted (noisy) information flows and individual elements failures. This ensures high efficiency and information processing reliability, and the ability to train and retrain as conditions change allows the timely adaption.

Thinking functions modeling based on neural-like growing networks gives a positive answer to the question whether the computer can 'think'. It should be noted that the basis for the modeling of thinking in systems with artificial intelligence is the knowledge describing the world in which these systems operate.

Considering the allegations of neurophysiologists regarding the reorganization of the child's brain cell mass in the learning process, it is safe to say that multi-connected neural-like growing networks with the network structure rebuilding and reorganizing itself in the process of learning is the most adequate human brain model.

Moreover, unconditioned reflexes (system primary automatisms) providing constant contact with the environmental objects and conditioned reflexes (system secondary automatisms) capable of long-term preservation in the memory and having the 'variability' (adjustment) property according to the environment, are simulated successfully in the mmren-GN structure. These properties are the basis of individual experience, learning and knowledge acquisition.

The principle of conditioned reflex discovered by of I.P. Pavlov, which is the principal activity of the brain and which is eventually the basis for the higher nervous activity and nearly all the behavioral acts of a high-order organism, allowed the physiologists to study and gain a better understanding of the brain work. In addition, a study of the conditioned and unconditioned reflexes organizational mechanisms in the mmren-GN structure provide the prerequisites for the creation of intelligent systems and robots with the functional capabilities of high-order organisms.

The artificial intelligence theory that has been developed provides a new ground for the production of advanced thinking and self-learning computers, intelligent systems and robots. The latter may have a variety of applications in the civil and military fields, especially in unpredictable situations and hazardous

environments. The key provisions of the artificial intelligence theory were presented in the materials of International Science and Information (SAI) Conference 2014 held in London. Complete information on the artificial intelligence theory can be found in the author's monograph, Artificial Intelligence, published in Ukraine in 2013 in Russian language.

References

1. Yashchenko, V.A.: Receptor-effector neural-like growing networks—effective tool for modeling intelligence. I. Cybern. Syst. Anal. **4**, 54–62 (1995)
2. Yashchenko, V.A.: Receptor-effector neural-like growing networks—effective tool for modeling intelligence. II. Cybern. Syst. Anal. **5**, 94–102 (1995)
3. Yashchenko, V.A.: Receptor-effector neural-like growing network—efficient tool for building intelligence systems. In: Proceedings of the Second International Conference on Information Fusion, vol. II, pp. 1113–1118. Sunnyvale Hilton Inn, Sunnyvale, 6–8 July 1999
4. Yashchenko, V.A.: Secondary automatisms of intelligent systems. Artif. Intell. **3**, 432–447 (2005)
5. Sokolov, E.N.: The principle of vector coding in psychophysiology. Moscow Univ. Messenger. Ser. 14 Psychol. **4**, 3–13 (1995)

Detection of Privilege Abuse in RBAC Administered Database

Udai Pratap Rao and Nikhil Kumar Singh

Abstract In this paper, we propose an approach for detection of database privilege abuse in Role Based Access Control (RBAC) administered database. The proposed approach extracts the data dependencies among the attributes of relations of the database. Role based data dependency rule miner (RBDDRM) algorithm is used to mine role-wise data dependencies from database log. These data dependencies are considered as role profiles, which are used to detect the misuse of privileges by database users.

Keywords Database security · Role based access control · Data mining · Data dependency · Database intrusion detection · Database privilege abuse

1 Introduction

In the era of Information Technology, data is a valuable asset for all organizations. Most of the data resides in Database Management System (DBMS). Sometimes such database is of worth millions of dollars and therefore companies are much conscious about controlling access to these data. Organizations which handle such individual's data have to provide confidentiality to meet legal policies and regulations. Recently, 22 million user IDs have been stolen from Yahoo Japan [1]. Such events affect the reputation of an organization and economic loss. Therefore, protection of database from insider and outsider threats is a non-trivial task.

U.P. Rao (✉) · N.K. Singh
Department of Computer Engineering,
S. V. National Institute of Technology,
Surat 395007, Gujarat, India
e-mail: udaiprataprao@gmail.com

N.K. Singh
e-mail: nikhil.singh31@gmail.com

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_4

Malicious activity can be caused by outsider, insider or by the combined efforts of both. The outsider is one who gains the illegal access to the database and misuses it. Misuse may be stealing of data and selling it to third parties or destroying the database. There are many proposed methods to protect against outsider attacks; one of them is Authentication. Insider is one who is authenticated to access the database. Insider is authorized to access whole or part of a database. Insider may misuse his authority of accessing the database by leaking it or selling it to competitors. Insider attacks are difficult to detect compared to outsider attacks, as an insider is within their access limits, but use data with malicious intention. Database breaches are increasing day by day and are threatening the privacy of individuals and organizations. According to Verizon 2010 data breach investigation report [2], 48 % of attacks involve insider. Recently, many have hypothesized that insider crime would rise due to financial strain imposed by global economic conditions [2]. According to this investigation report out of all attacks involving insiders, 90 % were deliberate and malicious. This shows that detection of malicious insider attack in the database is of great concern.

There are many Intrusion detection systems in literature tailored to Operating System and Network attacks; however, they are insufficient to guard against database attacks. Actions which are malicious for database management system may not be malicious for Network or Operating System. Without detecting malicious transactions, database damage assessment and recovery is not possible. Therefore intrusion detection approach tailored to guard against database attacks is needed.

In this paper, we propose an approach to detect transactions which possibly abuse database privileges. Our approach is based on mining role-wise data dependencies among attributes of relations of the database. Each role has different access patterns and thus has a different set of data dependencies than other roles. We use rule-based classification, and thus our approach generates data dependencies in the form of classification rules. Classification rules generated by our approach are treated as data dependency rules. These data dependency rules are later used to detect the possibility of database privilege abuse by new transaction from the user.

The rest of the paper is organized as follows: Sect. 2 gives an overview of related work in the area of database intrusion detection. Our approach is described in detail in Sect. 3. Section 4 shows experimental results and analysis of our approach with an existing approach proposed in [3]. Lastly conclusion and future scope are presented in Sect. 5.

2 Related Work

A complete solution to data security must meet the following Confidentiality, Integrity, and Authentication (CIA) requirements [4]

1. Confidentiality refers to the protection of data against unauthorized disclosure,
2. Integrity refers to the prevention of unauthorized and improper data modification, and

3. Availability refers to the prevention and recovery from hardware and software errors and from malicious data access denials, making the database system unavailable [4].

According to Bertino and Sandhu [4], various components of database management system are used to ensure the database security. Access control mechanisms are used to restrict the user's access to a subset of the database. Access control policies help in ensuring confidentiality. The access control mechanism along with semantic integrity constraints are used to ensure data integrity. Whenever a user wants to modify data from a database, he must have rights to do that. Access control policies, checks whether the user has, such rights or not and semantic integrity constraints verifies that the updated data must be semantically correct. Semantic correctness of the modified data are checked by the set of conditions, or set of predicates. Data availability is ensured by combining benefits from recovery subsystem and concurrency control mechanism. They ensure that data is available to the user even in the situations like hardware failure, software failure and access from multiple applications simultaneously.

Despite of effective authentication mechanisms and sophisticated access control mechanisms, there are chances of attacks on the database. Sometimes external user bypasses the authentication mechanism and impersonates as some internal user. Many times legitimate user tries to misuse the authorization provided to him. Without crossing boundaries of the access granted to him; he tries to steal or modify data, such attacks are called internal attacks. Our focus is mainly on detecting the internal attacks, i.e. detection of database privilege abuse.

Many approaches have already been proposed in the field of intrusion detection. But, most of them are tailored to detect intrusion at the level of operating system or network. Comparatively fewer approaches are found in literature for intrusion detection, which is tailored to detect database attacks. Javidi et al. [5] surveyed intrusion detection approaches tailored to database. Mainly two types of approaches have been proposed for database intrusion detection; one is the signature based approach and other is an anomaly based approach. To detect intrusion, signature based approaches keep records of signatures (fingerprints) of valid transactions, while anomaly based approaches use mining techniques to extract user profiles from database log.

Chung et al. [6], proposed the system called DEtection of MIsuse in Database Systems (DEMIDS) in 1999. This approach is based on the data structure, integrity constraints (encoded in the data dictionary) and user behavior reflected in the audit logs. DEMIDS defines notion of distance measure for mining frequent itemsets from database log. These frequent item-sets are the profiles of users, which can later be used to detect database attacks. The major drawback of DEMIDS system is that it is totally theoretical concept and no experimental evidence has been presented. In 2000 Lee et al. [7], proposed database intrusion detection system for real-time database systems. They exploited real-time properties of data for detecting intrusions. They keep track of update rates of data objects. If any data object is updated before time, then it is considered as an attack. They claim

that although their study focused on sensor transactions and temporal data objects, their approach can be generalized to be applicable to user transactions and non-real-time secure data objects.

Lee et al. [8] in 2002 proposed fingerprint (signature) based approach in which they store the signature of legitimate transactions; when a new transaction is executed by a user, it is checked against previously stored fingerprints. If it matches any of previously stored signatures it is considered as valid transaction, otherwise it is considered as malicious transaction. The main drawback of signature based approaches is that, they are only possible if there are fixed applications which can access database.

In 2004 Hu and Panda [3] proposed data mining approach to detect intrusions. They used rule based classification for their system. Classification rules are data dependency rules which reflect the existing data dependencies among attributes of relations of the database. The rules generated can be seen as the normal behavior of the users. If new transaction deviates from the normal behavior, then it is detected as malicious. Hu and Panda [3] emphasize only on malicious modification of data, while malicious read operation will not be detected by their approach. In this approach, each attribute is given equal importance and there is no concept of sensitivity of attribute.

Vieira and Madeira [9] proposed signature based mechanism called Database Malicious Transaction Detector (DBMTD) in 2005. They make profiles for each transaction that can be executed by any of the applications which are permitted to access the database. They also maintain the list that which user is authorized to execute particular transaction. Authorized transactions are represented in the form of graphs, which are the signatures for the valid transactions. They have proposed manual profiling of transactions, which is not possible if the number of valid transactions by applications is too high. Each valid transaction must be uniquely identified by TID. Each time new transaction arrives, based on TID they select the transaction profile graph. A new transaction is now checked against the selected transaction profile graph, if the new transaction does not follow the graph, it is detected as malicious transaction otherwise it is considered to be legitimate transacted. The drawback of their approach is that it is intended for small and fixed applications. If applications are high in number and are not fixed, then this approach cannot be applied.

Bertino et al. [10] proposed intrusion detection scheme for RBAC-administered database in 2005. They have used nave Bayes classifier for detecting intrusion. They have proposed three levels of triplets that can be used to represent queries from database log after pre-processing. Using the role information and triplets of all queries from the log, they form role profiles, which are nothing but the probabilities calculated using the nave Bayes classifier. When new query is executed, using role profiles (probabilities) of each role, it is predicted that which role should have executed this query. If role obtained is different from the original rule which has executed this query then it is considered as an attack. Later in 2008 Kamra et al. [11] improved their approach by using quiplets [11] in place of triplets. In quiplets they maintain separate information about projected attributes and selected attributes. The drawback of their both approaches is that if the database is not RBAC-administered then they will have to maintain separate profiles for each

user, which will add large overhead in terms of memory required to store the profiles and execution time for classification.

Srivastava et al. [12] improved Data Dependency Rule Miner (DDRM) approach [3] and proposed Weighed Data Dependency Rule Miner (WDDRM) algorithm in 2006. In WDDRM weights are assigned to attributes based on their sensitivity before mining the database log. This helps in extracting sequences which contain more sensitive attributes and are very less frequent. One of the main problems with this approach is the selection of the appropriate value of support and confidence.

Mathew et al. [13] proposed data-centric approach to insider attack detection in database systems in 2010. This is a novel approach for detecting intrusion and is the first data-centric approach for detecting database intrusion. They modeled user access patterns by profiling the data points that users access, in contrast to analyzing the query expressions in prior approaches.

Rao et al. [14] improved the approach of Bertino et al. [10] in 2010. They have used one triplet per transaction instead of one triplet per query. This one triplet per transaction exploits the inter-query dependencies. They used Naive Bayes classifier for profile generation.

In 2013 Rao and Patel [15] proposed an effective database recovery mechanism that includes the domain specific knowledge with dependency relationships to decide the set of affected transactions. In the recovery phase, they considered only affected and malicious transactions for rollback and skipped the good transactions.

Kamra and Bertino [16] proposed an intrusion response system for relational database. They proposed an interactive Event-Condition-Action type response policy language that makes it very easy for the database security administrator to specify appropriate response actions for different circumstances depending upon the nature of the anomalous request. The two main issues that they address in context of such response policies are that of policy matching, and policy administration. For the policy matching problem, they proposed two algorithms that efficiently search the policy database for policies that match an anomalous request. The other issue that they address is that of administration of response policies to prevent malicious modifications to policy objects from legitimate users.

3 Our Contribution

We have proposed RBDDRM algorithm which uses a different form of transaction representation after pre-processing to minimize the undesirable rule extraction. A conditional rule is proposed to detect the malicious read operations on the database and also a scheme (max disobeyed confidence) is proposed which uses confidence of rules to find the severity of the malicious activity. Now we will discuss our proposed transaction representation, conditional rules and max disobeyed confidence. And in the next section we will discuss the RBDDRM algorithm in detail.

3.1 Transaction Representation

Hu and Panda [3] used database log as input to generate data dependency rules, they pre-process the transactions present in the database log for mining the log. Consider the following transaction containing two queries,

$$Sid = Select\ s\_id\ from\ student\ where\ s\_name = \text{`}abc\text{'}$$

$$Update\ marks\ set\ m\_phy = 90,\ m\_chm = 92,\ m\_mat = 95\ where\ m\_id = Sid$$

after pre-processing output is,

$$\langle r(s\_name), r(s\_id), r(m\_id), w(m\_phy), w(m\_chm), w(m\_mat)\rangle$$

Here, r represents read operation or condition on attribute, while w represents write operation on attribute. Above output represents the sequence in which attributes are accessed or updated. But, when we consider updating of m_phy, m_chm and m_mat in a transaction they are updated together and not in sequence, i.e. sequence in which these attributes are updated is meaningless. Also, if there is a transaction which mentions the attributes in a different sequence than the result after pre-processing will be different, this should not be like that. For example, the same transaction is represented as follows:

$$Sid = Select\ s\_id\ from\ student\ where\ s\_name = \text{“}abc\text{”}$$

$$Update\ marks\ set\ m\_mat = 95,\ m\_chm = 92,\ m\_phy = 90\ where\ m\_id = Sid$$

after pre-processing output is,

$$\langle r(s\_name), r(s\_id), r(m\_id), w(m\_mat), w(m\_chm), w(m\_phy)\rangle$$

Here, we can see that after pre-processing there are different representations of the same transaction. Also, if such transactions are frequent, this can lead to generation of rule like, immediately after updating m_phy there must be update operation on m_chm and subsequently on m_mat. As we all know such rules are not meaningful as these update operations should not necessarily take effect in that strict sequence.

We can solve this type of problem by proposing the new representation style of transaction after pre-processing. We can better understand it by an example. Consider the same example of two queries in a transaction shown above.

$$Sid = Select\ s\_id\ from\ student\ where\ s\_name = \text{“}abc\text{”}$$

$$Update\ marks\ set\ m\_phy = 90, m\_chm = 92, m\_mat = 95\ where\ m\_id = Sid$$

according to our new proposed representation after pre-processing output should be in the form—

$$\langle r(s\_name), r(s\_id), r(m\_id), w(m\_phy, m\_chm, m\_mat)\rangle$$

Here, w represents simultaneous write operations on set of attributes present in the parentheses after it; r represents simultaneous read operations on set of attributes present in the parentheses after it. This representation eliminates the possibility of meaningless rule extraction which is discussed above.

3.2 Conditional Rule

The approach proposed by Hu and Panda [3] is only concerned about malicious modifications to the database; it cannot detect the malicious read operations on the database. So, to detect the malicious read operation on the databases we have proposed one more rule as conditional rule. Hu and Panda [3] is extracting two types of rules; read rule and write rule at the end of their Data Dependency Rule Miner (DDRM) algorithm. We can better understand those rules with the examples:

$$w(a) \rightarrow \langle r(b), r(c), r(d) \rangle \tag{1}$$

$$w(a) \rightarrow \langle w(b), w(c), w(d) \rangle \tag{2}$$

In the above represented two rules; Eq. 1 represents read rule while Eq. 2 represents write rule.

Read rule can be easily generated from read sequence like $\langle r(b), r(c), r(d), w(a) \rangle$, and $\langle r(b), r(e), r(c), r(f), r(d), w(a) \rangle$ by putting last operation on LHS and all other operations on RHS in the same sequence [3]. Read rule represents that transaction may need to perform all read operations on RHS in order before writing to attribute on LHS. For example—read rule in Eq. 1 conveys that before updating attribute a; there must be read operations on b, c and d attributes in the sequence. Sequence $\langle r(b), r(c), r(d), w(a) \rangle$ is following the read rule in Eq. 1. While sequence like $\langle r(c), r(b), r(d), w(a) \rangle$, $\langle r(b), r(e), r(d), w(a) \rangle$ are not following the read rule in Eq. 1. Write rule can be easily generated from write sequence like $\langle w(a), w(b), w(c), w(d) \rangle$ and $\langle w(a), w(e), w(b), w(c), w(f), w(d) \rangle$ by putting first operation on LHS and all other operations on RHS in the same sequence [3]. Write rule represents that transaction may need to perform all write operations on RHS in order after writing to attribute on LHS. For example—write rule in Eq. 2 conveys that after updating attribute a; there must be updates on attributes b, c and d in the same sequence. Sequences like $\langle w(a), w(b), w(c), w(d) \rangle$ and $\langle w(a), w(e), w(b), w(c), w(f), w(d) \rangle$ are following the write rule in Eq. 2. While sequences like $\langle w(a), w(c), w(b), w(d) \rangle$ and $\langle w(a), w(b), w(e), w(d) \rangle$ are not following the write rule in Eq. 2.

Here, we can observe that both rules are on write operations, means if any transaction has only read operations, then it does not need to follow any rule. So, if a malicious user steals the information from a database (within his access limits), then it will not be detected by these rules. Hence, we have introduced new type of rule; which is based on read or writes operations on the attributes of

database. We call these rules as conditional rules. According to these rules, there must be condition, operation on a set of attributes immediately before reading or writing on any attribute. For example

$$w(a) \rightarrow c(b, c, d) \tag{3}$$

$$r(a) \rightarrow c(b, c, d) \tag{4}$$

Both rules are conditional rules. They are somewhat different from read and write rules in the sense that they can have only one condition operation on set of attributes on RHS; while in read, write rules there can be sequence of read and write operations respectively on RHS.

Conditional rule in Eq. 3 conveys that there must be condition operation on attributes b, c, d immediately before writing to attribute a. For example—

UPDATE table_name SET $a =$ "*xyz*" *where* $b =$ "*xyz*" *and* $c =$ "*xyz*" *and* $d =$ "*xyz*"

For this query representation after pre-processing would be like $\langle c(b, c, d), w(a)\rangle$ follows the conditional rule in Eq. 3. While query—

UPDATE table_name SET $a =$ "*xyz*" *where* $=$ "*xyz*" *and* $d =$ "*xyz*"

for which representation after pre-processing would be like $\langle c(b, d), w(a)\rangle$ does not follow the conditional rule in Eq. 3.

Conditional rule in Eq. 4 conveys that there must be condition operation on attributes b, c, d immediately before reading to attribute a. This rule is similar to the conditional rule in Eq. 3. But the only difference is this rule is on read operation on attribute a. Thus, such rules help in detecting malicious read operations on the attributes of the database.

One thing to notice in these conditional rules is that condition operation on set of attributes should be immediately before the read or write operation. Hence, sequence like $\langle c(b, c, d), r(e), w(a)\rangle$ does not follow first rule. We discriminate read operation on attributes from conditions on attributes. If attributes are used for selecting the tuples from database, then we call condition operation on those attributes. Let us consider transaction of two queries:

select a, b from t_name where c = "*pqr*" *and d =* "*uvw*"

update t_name set a = "*xyz*" *and b =* "*xyz*" *where c =* "*pqr*" *and d =* "*uvw*"

Both the queries of a transaction shown above have condition operation on attributes c and d; first query has read operation on attributes a and b while second query has write operation on attributes a and b. Read, write and condition operations are represented by operation type along with unordered set of attributes on which operation is performed. Condition operation on attributes c and d is represented as $c(c, d)$; while read and write operations on attributes a and b are represented as $r(a, b)$ and $w(a, b)$ respectively.

A transaction on database is represented as sequence of operations on attributes of relations of database. An example transaction of two queries shown

in beginning of this section is represented as $\langle c(c,d), r(a,b), c(c,d), w(a,b)\rangle$. Condition operation has been shown before read or writes operations within same query; because tuples are selected first and then read or writes operation on some attributes of selected tuples is performed.

3.3 Max Disobeyed Confidence

We have proposed a scheme which uses confidence of rules to find the severity of the malicious activity. We simply use maximum function to obtain the highest confidence from confidences of disobeyed rules. This maximum confidence from disobeyed rules shows the severity of the malicious activity. It is very simple approach but this simple approach is based on the observation. If a transaction disobeys rule with 80 % confidence means, that transaction might be one of the transactions from 20 % of transactions which have no confidence in that rule. Let say there is a rule $r(a) \rightarrow c(b)$ with 80 % confidence. This means out of all transactions reading attribute a, 80 % of transactions have conditional operation on attribute b immediately before reading attribute a. while 20 % of transaction which does not support this rule; does not have conditional operation on attribute b immediately before reading attribute a. so, transaction detected as malicious might be from these 20 % of transactions which are not malicious. But, if transaction disobeys rule with 100 % confidence then there are more chances of that transaction being malicious

4 Role Based Data Dependency Rule Miner

Our work is based on relational database model [17] with RBAC access control mechanism [18]. We have used AproriAll algorithm [19] (except maximal phase) for extracting the sequences using normal definition of support. Then with the help of these generated sequences we are extracting read, write and conditional rules.

4.1 Terminologies

Definition 1 A sequence is an ordered list of operations. Sequence is represented as $\langle o_1, o_2, \ldots, o_n\rangle$. Here o_i represents read (r), write (w) or condition (c) operation on set of attributes. A sequences1 $\langle o_{11}, o_{12}, \ldots, o_{1n}\rangle$ (except conditional sequence) is said to be contained in sequence s2 $\langle o_{21}, o_{22}, \ldots, o_{2m}\rangle$ if there exist integers $i_1 < i_2 < \cdots < i_n$ such that $o_{1k} \subseteq o_{2i_k}$ and operations o_{1k} and o_{2i_k} are same for $1 \leq k \leq n$ [3].

A conditional sequence $\langle c_{11}, r/w_{12} \rangle$ is said to be contained in sequence s $\langle o_{21}, o_{22}, \ldots, o_{2n} \rangle$ if there exist integers i and $j = i + 1$, such that o_{2i} is condition operation and o_{2j} is same operation as o_{12}; and also $o_{11} \subseteq o_{2i}$ and $o_{12} \subseteq o_{2j}$.

Definition 2 A sequence in which last operation is write operation on exactly one attribute and all other operations are read operations on one or more attributes is Read sequence. Read rule can be easily generated from read sequence by putting last operation on LHS and all other operations on RHS in the same sequence. Read rule represents that transaction may need to perform all read operations on RHS in order before writing to attribute on LHS. Read sequence set is collection of all read sequences [3, 12].

Definition 3 A sequence in which first operation is write operation on exactly one attribute and all other operations are write operations on one or more attributes is write sequence. Write rule can be easily generated from write sequence by putting first operation on LHS and all other operations on RHS in the same sequence. Write rule represents that transaction may need to perform all write operations on RHS in order after writing to attribute on LHS. Write sequence set is collection of all write sequences [3, 12].

Definition 4 A sequence of two operations in which second operation is read or writes operation on exactly one attribute and first operation is condition operation on one or more attributes is called as conditional sequence. Conditional rule can be easily generated from condition sequence by putting last operation on LHS and first operation on RHS. Condition rule represents that transaction may need to perform condition operation on set of attributes on RHS before reading/writing to attribute on LHS. Conditional sequence set is collection of all conditional sequences.

Definition 5 Support for sequence s, is the count of all the transactions in the dataset that contains sequence s. A rule is represented as $o_1 \rightarrow \langle o_2, o_3, o_4, \ldots., o_n \rangle$ where, o_1 is operation on exactly one attribute. There are two sides of a rule, Left Hand Side (LHS) and Right Hand Side (RHS). o_1 is on LHS and sequence $\langle o_2, o_3, o_4, \ldots., o_n \rangle$ is on RHS. A rule can be generated from each of read, write and condition sequence.

Definition 6 Confidence of a rule is the ratio of support of sequence from which rule is generated to the support of sequence on LHS (sequence of one operation) of the rule. A rule r1 is said to be contained in rule r2, if LHS of both the rules is same and sequence on RHS of r1 is contained in sequence on RHS of r2.

Definition 7 In RBAC users are not directly associated with permissions. In RBAC permissions are assigned to the roles, and roles are assigned to the users [18]. One user may be assigned more than one role too. So, RBAC-administered database has to manage two types of associations, one is the associations between user and roles and the other is the associations between roles and permissions. When the user's job is changed, the administrator has to change the association of the user and roles [18]. And when the job function of any role is changed, the

administrator needs to change the associations only between role and permissions. In the absence of RBAC, there is the burden of changing permissions of all users with that role. When we use Data Dependency Miner algorithm [3] directly on the whole database log, we end up with many rules for the roles that frequently access the database, while less or no rules for roles who rarely access the database. If database is administered using RBAC then we can exploit role information to improve performance of data dependency miner.

4.2 System Architecture of Proposed IDS

As shown in Fig. 1, our approach works in two phases viz. learning phase and detection phase. During the learning phase, we have used the database log for mining role profiles. First of all, database transactions from database log will be pre-processed for extracting the useful features. During feature extraction, raw transactions are converted to useful representation which is applicable for our approach, as discussed in the previous section.

Once the pre-processing of all transactions from the database log has been completed, pre-processed transactions are fed to profile generation algorithm. The profile generation algorithm will extract the role-profiles from those pre-processed transactions. Each role will have its own role profile (association rules) at the end of the profile generation algorithm. These profiles are then stored on permanent storage for later use. Now, at the end of learning phase, we have a role profiles for each role, which represents the normal behavior of the role.

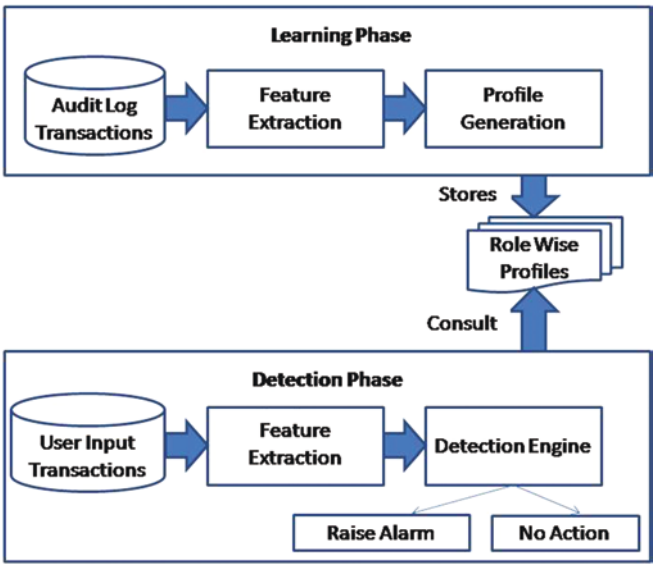


Fig. 1 Overview of our approach system architecture

During the detection phase, we check the new user transaction for possibility of privilege abuse. Transaction from the user will be pre-processed first; useful features will be extracted and will be converted to the same representation used during the learning phase. Then this pre-processed transaction will be fed to detection engine which will consult role profiles and will check the transaction against all the association rules corresponding to the role of the user. If the transaction disobeys any rule, then an alarm is raised. Otherwise, if transaction follows all the rules then it is considered as normal transaction and no action is taken in such case.

5 The Algorithm

Our approach works in two phases viz. learning phase and detection phase.

5.1 Learning Phase

After feature extraction from database log, pre-processed transactions are fed to proposed profile generation algorithm Role Based Data Dependency Rule Miner (RBDDRM). At the end of RBDDRM algorithm, role wise profiles are generated, and they are stored on permanent storage. RBDDRM algorithm is given below. Terminologies used for explaining RBDDRM algorithm are as follows:

T	set of all pre-processed database transactions from database log.
n	total number of roles, in database application.
T_k	set of pre-processed database transactions from database log executed by role k.
$\|T_k\|$	number of transactions from database log executed by role k.
RS_k	set of read sequences mined, related to role k.
WS_k	set of write sequences mined, related to role k.
CS_k	set of conditional sequences mined, related to role k.
RR_k	set of read rules generated, related to role k.
WR_k	set of write rules generated, related to role k.
CR_k	set of conditional rules generated, related to role k.
X	sequential patterns generated.
x_i	one of many sequential patterns generated.
$\|x_i\|$	length of sequential pattern x_i.

Learning Phase Algorithm

1. for each role k, $1 \leq k \leq n$
2. initialize $T_k = \{\}$
3. for each transaction t in T
4. insert t to T_k, where k is role who executed t
5. for each role k, where $|T_k| > 0$
6. initialize $RS_k = \{\}$, $WS_k = \{\}$, $CS_k = \{\}$, $RR_k = \{\}$, WR_kk $= \{\}$, $CR_k = \{\}$

7. generate sequential patterns X = {xi, | support (xi) > min support} by using AprioriAll algorithm. Using T_k as input to it.
8. for each sequential pattern x_i, where $|x_i| > 1$
9. if $|x_i| = 2$
10. if last operation in x_i is read/write operation on one element and first operation in x_i is conditional operation, then add xi to CS_k
11. if last operation in x_i is write operation on one element and all other operations in x_i are read operations, then add x_i to RS_k
12. if first operation in x_i is write operation on one element and all other operations in x_i are also write operations, then add x_i to WS_k
13. for each sequence s in CS_k
14. if confidence of conditional rule r generated from sequences > min confidence
15. if no rule in CR_k (with confidence equal to confidence of r) contains r
16. add rule r to CR_k
17. delete all rules from CR_k which has same confidence as r and are contained in r
18. for each sequence s in RS_k
19. if confidence of read rule r generated from sequence s > min confidence
20. if no rule in RR_k (with confidence equal to confidence of r) contains r
21. add rule r to RR_k
22. delete all rules from RR_k which has same confidence as r and are contained in r
23. for each sequence s in WS_k
24. if confidence of write rule r generated from sequence s > min confidence
25. if no rule in WR_k (with confidence equal to confidence of r) contains r
26. add rule r to WR_k
27. delete all rules from WR_k which has same confidence as r and are contained in r
28. store RR_k, WR_k, CR_k to permanent storage

In proposed RBDDRM algorithm, steps 1 to 4 divide the set of transactions into n subsets, each subset containing transactions executed by one role. Then steps 5 to 28 generate and store dependency rules for each subset generated by steps 1 to 4. In step 7 we use already existing AprioriAll algorithm [19], but we do not run maximal phase of that algorithm as we want all frequent sequences. Out of all sequences generated during step 7, steps 8 to 12 select useful read sequences, write sequences and conditional sequences. Steps 13 to 27 generate read, write and conditional rules from the sequences selected by steps 8 to 12. Steps 13 to 27 select rules with confidence greater than minimum confidence and discards rules with confidence less than minimum confidence. Step 28 stores them in permanent storage from where rules will be used later during detection phase.

5.2 Detection Phase

During detection phase new transaction from user is first pre-processed, then pre-processed transaction is fed to detection engine. Detection engine reads rules from permanent storage and check new transaction against dependency rules of related

role (role of user who has executed the transaction). If transaction satisfies the rules then it is normal and no action is taken otherwise it is considered as malicious and alarm is raised. Alarm is raised means entry is made to log of probable attacks. Detailed algorithm of detection engine of proposed approach is given below.

Detection Phase Algorithm

1. for each role k
2. initialize $RR_k = \{\}$, $WR_k = \{\}$, $CR_k = \{\}$
3. retrieve RR_k, WR_k, CR_k from permanent storage to memory
4. for each transaction t executed by role k, $1 \leq k \leq n$
5. max-disobeyed-confidence = 0
6. for each read operation in t
7. for each attribute a of read operation
8. for each rule r for attribute a in CR_k
9. if r is disobeyed and max-disobeyed-confidence < confidence(r)
10. max-disobeyed-confidence = confidence(r)
11. for each write operation in t
12. for each attribute a of write operation
13. for each rule r for attribute a in CR_k, RR_k and WR_k
14. if r is disobeyed and max-disobeyed-confidence < confidence(r)
15. max-disobeyed-confidence = confidence(r)
16. if max-disobeyed-confidence $\neq 0$
17. add the entry to log of possible attacks along with max-disobeyed-confidence

In detection engine algorithm, steps 1 to 3 are used to retrieve rules of all roles to their respective sets from permanent storage. Steps 4 to 17 check each new transaction for possibility of it being malicious. Step 5 sets variable max-disobeyed-confidence to zero. Max-disobeyed-confidence is used to track the highest confidence among confidences of all rules disobeyed by new transaction. Steps 6 to 10 check new transaction against conditional rules on attributes of read operations present in new transaction. Steps 11 to 15 check new transaction against read, write, conditional rules on attributes of write operations present in new transaction. Steps 16 and 17 check value of max-disobeyed-confidence, if it is zero new transaction is normal, otherwise entry is added in the log of possible attacks along with the value of max-disobeyed-confidence. Value of max-disobeyed-confidence can later be used by security personnel to guess the severity of attack. As the value of max-disobeyed-confidence is more, more is the possibility of attack.

6 RBDDRM Methodology

Learning phase and detection phase of RBDDRM algorithm can be better understood by an example. Table 1 shows pre-processed transactions from database log which are executed by role k, i.e. T_k. We have shown how learning phase

Table 1 Example transactions

TID	Transactions
T1	$\langle c(a,e), r(a,b), c(a,b), w(c), r(c,e), w(a,d)\rangle$
T2	$\langle r(a,d), w(a)\rangle$
T3	$\langle c(a), w(c), w(a)\rangle$
T4	$\langle c(a,e), r(b), c(d), w(c), r(c,d,e), w(a,d)\rangle$
T5	$\langle c(a,d,e), r(b), c(b,c), w(e)\rangle$

Table 2 Example—sequence sets generated

SeqID	Support	Sequence	Sequence set
S1	3	⟨c(a), r(b)⟩	Conditional sequence set (CS_k)
S2	2	⟨c(a), w(c)⟩	
S3	3	⟨c(e), r(b)⟩	
S4	3	⟨c(a,e), r(b)⟩	
S5	2	⟨r(a), w(a)⟩	Read sequence set (RS_k)
S6	2	⟨r(b), w(a)⟩	
S7	2	⟨r(b), w(c)⟩	
S8	2	⟨r(b), w(d)⟩	
S9	2	⟨r(c), w(a)⟩	
S10	2	⟨r(c), w(d)⟩	
S11	2	⟨r(d), w(a)⟩	
S12	2	⟨r(e), w(a)⟩	
S13	2	⟨r(e), w(d)⟩	
S14	2	⟨r(c, e), w(a)⟩	
S15	2	⟨r(c, e), w(d)⟩	
S16	2	⟨r(b), r(c), w(a)⟩	
S17	2	⟨r(b), r(c), w(d)⟩	
S18	2	⟨r(b), r(e), w(a)⟩	
S19	2	⟨r(b), r(e), w(d)⟩	
S20	2	⟨r(b), r(c, e), w(a)⟩	
S21	2	⟨r(b), r(c, e), w(d)⟩	
S22	3	⟨w(c), w(a)⟩	Write sequence set (WS_k)
S23	2	⟨w(c), w(d)⟩	
S24	2	⟨w(c), w(a, d)⟩	

for one role goes on (from step 6 to 28 of RBDDRM algorithm). Steps 1 to 5 of RBDDRM algorithm only divide the transactions from database log into subsets; one subset for each role. Table 2 shows one such subset generated after step 5.

AprioriAll algorithm [19] (except maximal phase) is applied on the above set of transactions. Several sequences are generated as a result. Steps 9 to 12 of RBDDRM algorithm selects useful sequences from sequences generated by AprioriAll algorithm. After step 12, CS_k, RS_k and WS_k have several useful

sequences which are shown in Table 2 (Minimum support value of 40 %, i.e. 2 transactions has been considered). Steps 13 to 27 generate rules from selected sequences.

Table 3 shows rules generated from sequences shown in Table 2 (Minimum confidence of 60 % is considered). At the end of step 27 of RBDDRM algorithm, we have several rules in conditional, read and write rule sets. These rules are stored on permanent storage by step 28. We can easily observe comparing Tables 2 and 3 that no rules are generated for several sequences with Sequence IDs S5, S6, S9, S11, S12, S14, S16, S18 and S20. This is because rules generated by these sequences do not have minimum confidence, i.e. 60 %. For example rule generated by S5 is $w(a) \rightarrow r(a)$, its confidence can be calculated by taking the ratio of support ($\langle r(a), w(a) \rangle$) to the support ($\langle w(a) \rangle$). Support of ($\langle r(a), w(a) \rangle$) is 2 as it is supported by only two transactions T1 and T2. Support of ($\langle w(a) \rangle$) is 4 as it is supported by only four transactions T1, T2, T3 and T4. So, confidence of rule $w(a) \rightarrow r(a)$ generated by S5 is 2/4, which is 50 % (less than minimum confidence, i.e. 60 %). So, this rule is not of our interest, and thus not present in read rule set.

Also rules represented by RID R1, R2, R6, R9–R14 in Table 3 are not present in rule sets at the end of step 27. Because they are already contained in some other rule (in corresponding rule set) with same confidence. Now we will take new transactions and will try to classify it as normal or malicious transaction with the help of rules shown in Table 3. Let say new transaction is $\langle c(a,e), r(a,b), c(a,b), w(c), r(c,e), w(a,d) \rangle$. First operation in transaction is conditional operation, so there are no rules for it. Second operation is read operation on attributes a and b. One conditional rule R3 is present for read operation on b. This rule is obeyed by given transaction. Moving further, third operation is

Table 3 Example transactions

RID	Conf (%)	Rule	SeqID	Rule set
R1	100	r(b) → c(a)	S1	Conditional rule set (CR_k)
R2	100	r(b) → c(e)	S3	
R3	100	r(b) → c(a, e)	S4	
R4	66	w(c) → c(a)	S2	
R5	100	w(c) → w(a)	S22	Write rule set (WR_k)
R6	66	w(c) → w(d)	S23	
R7	66	w(c) → w(a, d)	S24	
R8	66	w(c) → r(b)	S7	Read rule set (RR_k)
R9	100	w(d) → r(b)	S8	
R10	100	w(d) → r(c)	S10	
R11	100	w(d) → r(e)	S13	
R12	100	w(d) → r(c, e)	S15	
R13	100	w(d) → r(b), r(c)	S17	
R14	100	w(d) → r(b), r(e)	S19	
R15	100	w(d) → r(b), r(c, e)	S21	

conditional operation so there are no rules for it. Fourth operation is write operation on attribute c. R4, R5, R7 and R8 are rules on write operation on c, all of which are obeyed by the given transaction.

Then there is read operation on attributes c and e, none of read on c or e have any rules. Last operation is write operation on attributes a and d. No rule is present for write on a, but there is rule R15 for write on attribute d. R15 is also obeyed by the transaction. As transaction obeys all the rules, it is considered as normal transaction.

Now, consider transaction $\langle c(a), w(c), w(a, d)\rangle$ as new transaction. First operation is conditional operation on attribute a, so no rule on conditional operation. Second operation is write operation on attribute c. R4, R5, R7 and R8 are rules on write operation on c. Rules R4, R5 and R7 are followed by new transaction. But rule R8 is not followed, so now transaction has been detected as malicious and max_disobeyed_confidence is equal to confidence of R8, which is equal to 66 %. Now last operation is write on attributes a and d; write on attribute a has no rules, but write on attribute d has one rule R15. R15 is not followed by new transaction, which has confidence equal to 100 % which is higher than max_disobeyed_confidence. So max_disobeyed_confidence will be updated from 66 to 100 %.

7 Experimental Results

We have used TPC-C [20] (on-line transaction processing benchmark) database schema. We have manually generated 20 genuine transactions, executed by different roles. We then pre-processed those transactions, and represented them in appropriate format; required by our implementation. There are three roles in the system viz. administrator, customer and visitor.

We have used synthetic dataset for evaluating our performance. For each role synthetic dataset is generated from 20 genuine transactions of the corresponding role, which were populated to 1,000 transactions randomly. While randomly populating transactions, we have considered frequency with which all three roles interact with the database. We have also taken into account the frequency of execution of any transaction by particular role. These transactions have been used by us to extract the dependency rules as well as to evaluate False Positives and True Negatives.

Attacks were generated by randomly changing few attributes in the operations of transaction by other attributes of the same relation (same relation of schema). We have generated 100 malicious transactions in this way. These malicious transactions have been used by us to evaluate False Negatives and True Positives. We have compared our approach with existing DDRM algorithm proposed by Hu and Panda [3]. We have varied the minimum support value keeping minimum confidence value fixed at 75 % for both the approaches.

From Figs. 2, 3, 4, and 5 we observe that as minimum support increases, false positives and true positives decreases, while true negative and false negative

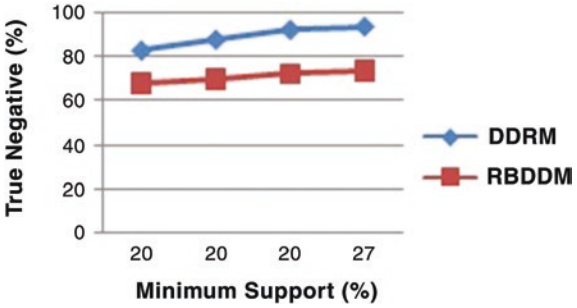


Fig. 2 TN of DDRM [3] and RBDDRM

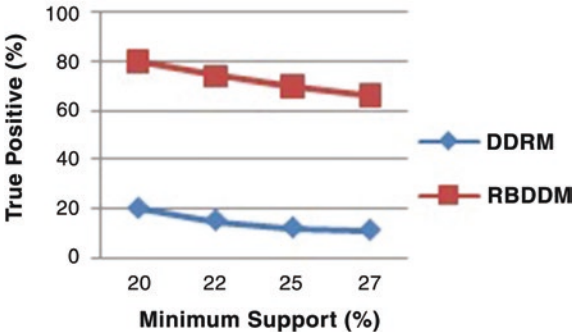


Fig. 3 TP of DDRM [3] and RBDDRM

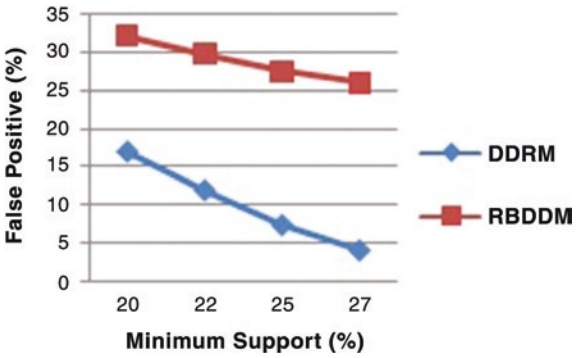


Fig. 4 FP of DDRM [3] and RBDDRM

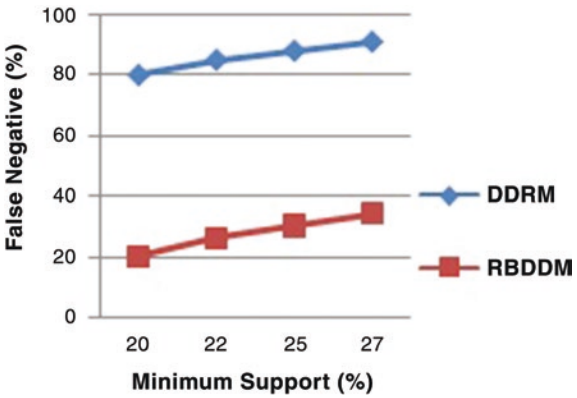


Fig. 5 FN of DDRM [3] and RBDDRM

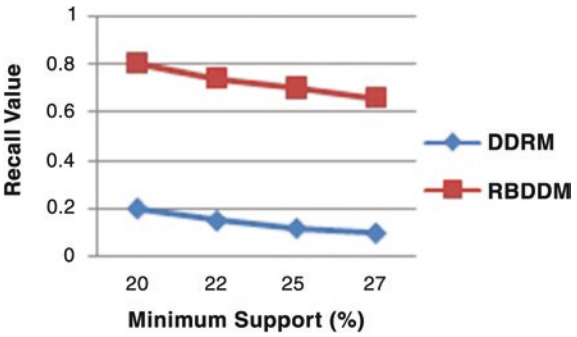


Fig. 6 Recall of DDRM [3] and RBDDRM

increases. As support increases, fewer rules are generated as there are fewer sequences with high support. So, if there are fewer rules there will be less positive results. With increase in minimum support, false positives are decreasing but it increases false negatives i.e. at higher minimum support very few normal transactions are falsely predicted as malicious, but at the same time many malicious transactions passes undetected; while at lower minimum support very few malicious transactions go undetected. At the same time many normal transactions are falsely predicted as malicious.

Figures 4 and 5 shows comparison of false positives and false negatives of both the approaches. It is observed from the graph that false positives of our approach are slightly more than existing DDRM [3] approach; but there is significant improvement in false negatives as false negatives in our approach are less than DDRM [3]. Improvement is not seen in false positives may be because of new conditional rule type introduced in our approach; many such conditional rules are generated leading to more false positives.

Figure 6 shows comparison of DDRM [3] and RBDDRM, considering recall as metric. It is observed that recall capacity of our approach is better than DDRM (i.e. our approach will detect more attacks as compared to DDRM [3]). As the minimum support increases, recall capacity of both approaches decreases. At higher minimum support fewer rules will be generated in both approaches leading to higher false negatives and thus lower recall value.

8 Conclusion

In this paper, we proposed data mining based approach i.e. Role Based Data Dependency Rules Miner (RBDDRM) to detect malicious transactions in RBAC-administered database. Our approach concentrates on extracting role-wise data dependencies (role profiles). Data dependency rules (association rules) extracted by our approach are used as classification rules for detecting anomaly. The experiment on synthetic dataset shows that our approach performs better than DDRM algorithm [3].

Acknowledgment This research work is supported by Institute Research Grant (Ref. No.: Dean (R&C)/1503/2013-14, dated: 17-02-2014) of S.V. National Institute of Technology Surat (Gujarat) 395007-India.

References

1. 22 million user IDs may have been stolen from Yahoo Japane. http://www.infosecurity-magazine.com/view/32498/22-million-user-ids-may-havebeen-stolen-fromyahoojapan?utm_medium=twitterutm_source=twitterfeed (2013) [Online; Accessed July 2013]
2. 2010 Data Breach Investigations Report. http://www.verizonenterprise.com/resources/reports/rp_2010-data-breach-report_en_xg.pdf (2013) [Online; Accessed July 2013]
3. Hu, Y., Panda, B.: A data mining approach for database intrusion detection. In: Proceedings of the 2004 ACM Symposium on Applied Computing, pp. 711–716. ACM (2004)
4. Bertino, E., Sandhu, R.: Database security-concepts, approaches, and challenges. Dependable Secure Comput. IEEE Trans. **2**(1), 2–19 (2005)
5. Javidi, M.M., Rafsanjani, M.K., Hashemi, S., Sohrabi, M.: An overview of anomaly based database intrusion detection systems. Ind. J. Sci. Technol. **5**(10), 3550–3559 (2012)
6. Chung, C.Y., Gertz, M., Levitt, K.: Demids: a misuse detection system for database systems. In: Integrity and Internal Control in Information Systems, pp. 159–178. Springer, Berlin (2000)
7. Lee, V.C., Stankovic, J.A., Son, S.H.: Intrusion detection in real-time database systems via time signatures. In: Real-Time Technology and Applications Symposium, 2000. RTAS 2000. Proceedings of Sixth IEEE, pp. 124–133. IEEE (2000)
8. Lee, S.Y., Low, W.L., Wong, P.Y.: Learning fingerprints for a database intrusion detection system. In: Computer Security ESORICS 2002, pp. 264–279. Springer, Berlin (2002)
9. Vieira, M., Madeira, H.: Detection of malicious transactions in dbms. In: Dependable Computing, 2005. Proceedings of 11th Pacific Rim on International Symposium on IEEE, p. 8. IEEE (2005)
10. Bertino, E., Terzi, E., Kamra, A., Vakali, A.: Intrusion detection in RBAC-administered databases. In: Computer Security Applications Conference, 21st Annual, p. 10. IEEE (2005)
11. Kamra, A., Terzi, E., Bertino, E.: Detecting anomalous access patterns in relational databases. VLDB J. **17**(5), 1063–1077 (2008)
12. Srivastava, A., Sural, S., Majumdar, A.K.: Weighted intra-transactional rule mining for database intrusion detection. In: Advances in Knowledge Discovery and Data Mining, pp. 611–620. Springer, Berlin (2006)
13. Mathew, S., Petropoulos, M., Ngo, H.Q., Upadhyaya, S.: A data-centric approach to insider attack detection in database systems. In: Recent Advances in Intrusion Detection, pp. 382–401. Springer, Berlin (2010)
14. Rao, U.P., Sahani, G.J., Patel, D.R.: Detection of malicious activity in role based access control (RBAC) enabled databases. J. Inf. Assur. Sec. **5**(6), 611–617 (2010)
15. Rao, U.P., Patel, D.R.: Incorporation of application specific information for recovery in database from malicious transactions. Inf. Secur. J. Glob. Perspect. **22**(1), 35–45 (2013)
16. Kamra, A., Bertino, E.: Design and implementation of an intrusion response system for relational databases. Knowl. Data Eng. IEEE Trans. **23**(6), 875–888 (2011)
17. Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM **13**(6), 377–387 (1970)
18. Ferraiolo, D.F., Barkley, J.F., Kuhn, D.R.: A role-based access control model and reference implementation within a corporate intranet. ACM Trans. Inf. Syst. Secur. **2**(1), 34–64 (1999)
19. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Data Engineering, 1995. Proceedings of the Eleventh International Conference on IEEE, pp. 3–14 (1995)
20. TPC Council-TPC Benchmark C Standard Specification Version 5.1

Learning-Based Leaf Image Recognition Frameworks

Jou-Ken Hsiao, Li-Wei Kang, Ching-Long Chang and Chih-Yang Lin

Abstract Automatic plant identification via computer vision techniques has been greatly important for a number of professionals, such as environmental protectors, land managers, and foresters. In this chapter, we propose two learning-based leaf image recognition frameworks for automatic plant identification and conduct a comparative study between them with existing approaches. First, we propose to learn sparse representation for leaf image recognition. In order to model leaf images, we learn an over-complete dictionary for sparsely representing the training images of each leaf species. Each dictionary is learned using a set of descriptors extracted from the training images in such a way that each descriptor is represented by linear combination of a small number of dictionary atoms. Second, we also propose a general bag-of-words (BoW) model-based recognition system for leaf images, mainly used for comparison. We experimentally compare the two learning-based approaches and show unique characteristics of our sparse representation-based framework. As a result, efficient leaf recognition can be achieved on public leaf image dataset based on the two proposed methods. We also show that the proposed sparse representation-based framework can outperform our BoW-based one and state-of-the-art approaches, conducted on the same dataset.

Keywords Plant identification · Leaf image recognition · Dictionary learning · Bag-of-words (BoW) · Sparse representation

J.-K. Hsiao · L.-W. Kang (✉) · C.-L. Chang
Department of Computer Science and Information Engineering,
National Yunlin University of Science and Technology, Yunlin, Taiwan
e-mail: lwkang@yuntech.edu.tw

L.-W. Kang
Graduate School of Engineering Science and Technology-Doctoral Program,
National Yunlin University of Science and Technology, Yunlin, Taiwan

C.-Y. Lin
Department of Computer Science and Information Engineering, Asia University,
Taichung, Taiwan

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_5

1 Introduction

Recently, with global warming, rapid urban development, biodiversity loss, and environmental damage, there has been great demand for applying advanced computer vision techniques to broaden botanical knowledge. Automatic plant identification technique is one of them, which is of great importance for a number of professionals, such as environmental protectors, land managers, foresters, agronomists, and amateur gardeners [1].

Plant identification is a fairly difficult task even for experienced botanists, considering the huge number of species existing in the world [2], as examples shown in Fig. 1. The task is generally based on the observation of the morphological characteristics of a plant, such as stems, roots, flowers, and leaves. Most important information about the taxonomic identity for a plant is usually contained in its leaves [2], as examples shown in Fig. 2. Therefore, similar to most existing image-based plant identification [1–15] or retrieval [16–18] approaches, the proposed framework focuses on the recognition of leaf images which can be further applied to plant identification.

To achieve leaf image recognition, the first step is to extract feature(s) from leaf images to be recognized. Similar to feature extraction from general images, a leaf image can be characterized by extracting its color [13, 17], texture [1, 2, 13, 17],

Fig. 1 Examples of number of plant species

Fig. 2 Examples of number of leaf species

and shape [1–18] features. Nevertheless, the color of a leaf may vary with the seasons and climatic conditions, while most species of leaves have similar colors. Hence, only shape and texture features are shown to be applicable in leaf image recognition or retrieval in the literature.

On the other hand, to achieve recognition of leaf images, it is usually required to train a classifier in advance using some training leaf images. Most recently developed plant identification frameworks rely on content-based image retrieval techniques (usually with *k*-nearest neighbor, i.e., *k*-NN, classification) [1–8], while several ones are based on neural network-based classification [9, 10]. Moreover, a novel classification method, called move median centers (MMC) hypersphere classifier was proposed in [11], and another similar one called moving center hypersphere (MCH) classifier was also proposed in [12].

In this chapter, we propose two learning-based leaf image recognition frameworks for automatic plant identification and conduct a comparative study between them with existing approaches. First, we propose to formulate leaf image recognition as a sparse representation problem. Sparse representation (or sparse coding) techniques have been shown to be efficient in solving several computer vision problems, such as face recognition, action recognition, image denoising and enhancement [19–28]. For the proposed sparse representation-based leaf recognition method, in the learning stage, we learn a dictionary for sparely representing the training leaf images in each plant species. In the recognition stage, we calculate the sparse representations of a test image with respect to each learned dictionary of species to find the leaf category with the largest correlation between them.

Moreover, we also present a general bag-of-words (BoW) model-based recognition system for leaf images, mainly used for comparison. BoW model has been also widely applied in the applications of image recognition, classification, and retrieval [29–32]. In the learning stage of our BoW-based framework, we train a codebook consisting of a number of representative codewords for representing the training leaf images in each plant species. Then, we train a support vector machine (SVM) [33] classifier for leaf image classification. In the recognition stage of our BoW-based framework, we quantitatively represent each test leaf image based on the trained codebook, followed by recognizing the image with the trained SVM classifier.

The main contribution of this work is three-fold: (i) to the best of our knowledge, we are among the first to propose a sparse representation framework for leaf image recognition; (ii) the proposed framework is adapted to newly added leaf species without retraining classifiers and suitable to be highly parallelized as well as integrated with any leaf image descriptors/features; and (iii) benefited from the property of sparse representation, the proposed method would be robust to inaccurate feature extraction of leaf images, while providing more compact and richer representation for leaf images.

The rest of this chapter is organized as follows. In Sect. 2, we briefly review the concepts of sparse representation and dictionary learning techniques, which form the basis of our sparse representation framework for leaf image recognition. Section 3 presents the proposed leaf image recognition framework via sparse

representation. Section 4 introduces the proposed BoW model-based recognition system for leaf images, used for comparison. In Sect. 5, experimental results are demonstrated. Finally, Sect. 6 concludes this chapter.

2 Sparse Representation and Dictionary Learning

As an example shown in Fig. 3, sparse coding (SC) [27, 28] is a technique of finding a sparse representation for a signal by solving a small number of nonzero or significant coefficients corresponding to the atoms in a dictionary. To learn a dictionary for sparsely representing each signal, such as a feature vector or an image patch extracted from an image, we collect a set of training exemplars, y_j, j = 1, 2, …, P, to learn a dictionary D sparsifying y_j by solving the following optimization problem [25, 26]:

$$\min_{D,\, x_j} \frac{1}{P} \sum_{j=1}^{P} \left(\frac{1}{2} \left\| y_j - D x_j \right\|_2^2 + \lambda \left\| x_j \right\|_1 \right), \tag{1}$$

where x_j denotes the sparse coefficient vector of y_j with respect to D and λ is a regularization parameter. Equation (1) can be efficiently solved by performing a dictionary learning algorithm, such as the online dictionary learning [25] or K-singular value decomposition (K-SVD) [26] algorithms.

After obtaining the dictionary D, for each signal Q_j to be sparsely represented with respect to D, its sparse coefficient θ_j can be obtained by solving the following optimization problem [25, 26]:

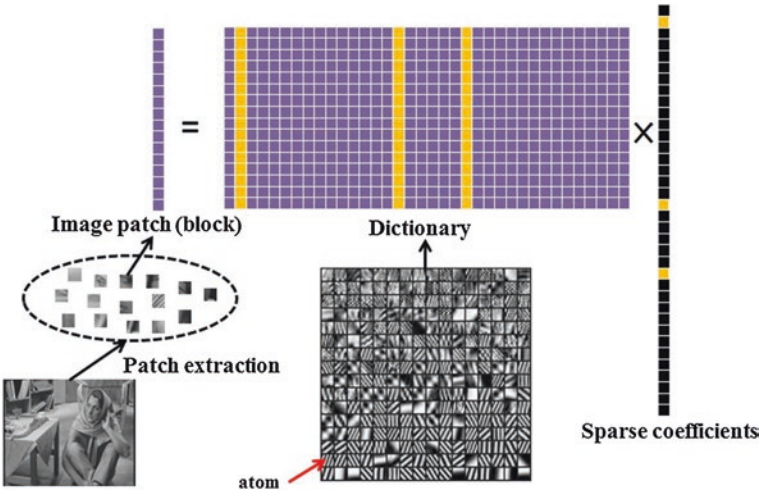


Fig. 3 An example of illustrating the sparse representation of image patches (blocks) based on a given dictionary

$$\min_{\theta_j} \left(\frac{1}{2} \left\| Q_j - D\theta_j \right\|_2^2 + \lambda \left\| \theta_j \right\|_1 \right). \tag{2}$$

That is, θ_j is a sparse coefficient vector for sparsely representing Q_j with respect to D, and Q_j can be approximately recovered via $D\theta_j$.

In this study, we propose to apply dictionary learning to learn a dictionary for representing each category of leaf images. Then, the recognition of an input leaf image can be achieved by analyzing the correlation between the input image and each leaf class, derived from calculating the sparse representation of the input image with respect to the learned dictionary of each class. The detailed method of our first leaf image recognition framework (sparse representation-based) will be elaborated in Sect. 3.

3 Proposed Leaf Image Recognition Framework via Sparse Representation

In this section, we first introduce the problem formulation of our first framework in this chapter, followed by presenting the proposed leaf image recognition framework via sparse representation (or sparse coding), consisting of the learning and recognition stages.

3.1 Problem Formulation of Our First Framework

In our first framework of this study, we consider total C species of leaves, where we learn the dictionary D_i for sparsely representing the i-th class of leaf images, i = 1, 2, …, C. To recognize a test leaf image I, we formulate the problem as a sparse representation problem and solving it by calculating the histogram h_i^I derived from the sparse representation of I with respect to each D_i, and identify the one with the largest correlation between h_i^I and D_i, i = 1, 2, …, C. The detailed method shall be elaborated below.

3.2 Learning Stage of Proposed First Framework

As illustrated in Fig. 4, in the learning stage of the proposed first framework, for each class L_i (the ith class) of leaf images, we select a number of training images to learn a dictionary D_i for representing this class. We extract a set of P_i descriptors (or feature vectors) $\{y_{ij} \in R^n\}_{j=1}^{P_i}$ from all training images in each L_i, $i = 1, 2, \ldots, C$, where any shape, texture, or hybrid image features can be used here, such as scale-invariant feature transform (SIFT) [34]. For learning the dictionary $D_i \in R^{n \times m}$ for compactly representing L_i, we apply dictionary learning to minimize [25]:

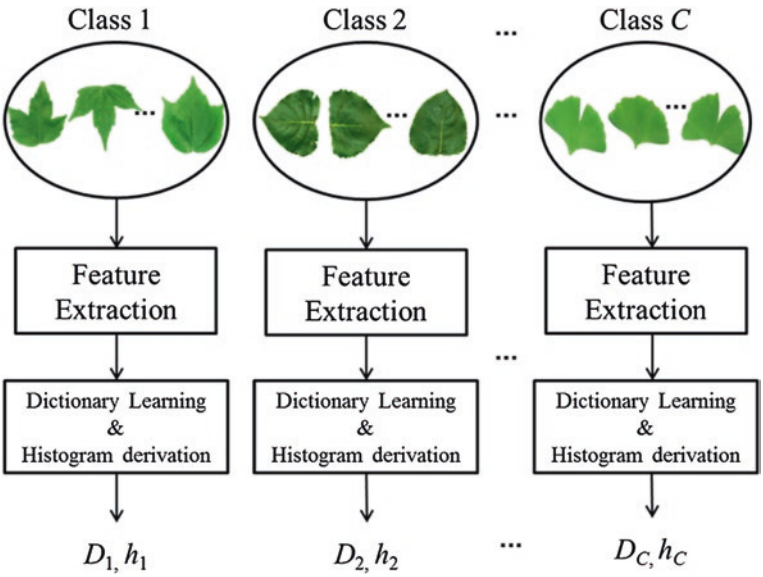


Fig. 4 The learning stage of the proposed first leaf image recognition framework via sparse representation

$$\min_{D_i \in R^{n\times m},\, x_{ij} \in R^m} \frac{1}{P_i} \sum_{j=1}^{P_i} \left(\frac{1}{2} \left\| y_{ij} - D_i x_{ij} \right\|_2^2 + \lambda \left\| x_{ij} \right\|_1 \right), \tag{3}$$

where x_{ij} denotes the sparse coefficients of y_{ij} with respect to D_i and λ is a regularization parameter. After obtaining D_i, we calculate the histogram h_i associated with the sparse coefficients $\left\{ x_{ij} \in R^m \right\}_{j=1}^{P_i}$ of $\left\{ y_{ij} \right\}_{j=1}^{P_i}$ as:

$$h_i = \frac{1}{P_i} \sum_{j=1}^{P_i} x_{ij}. \tag{4}$$

As an example, for the ith class, the learning stage is summarized in Fig. 5. Then, for each class L_i, we store its dictionary D_i and histogram h_i for leaf image recognition task described in Sect. 3.3.

3.3 Recognition Stage of Proposed First Framework

As illustrated in Fig. 6, for a test leaf image I to be recognized, we first extract its set of descriptors (image feature vectors) $\left\{ Q_j \in R^n \right\}_{j=1}^{q}$ and the corresponding

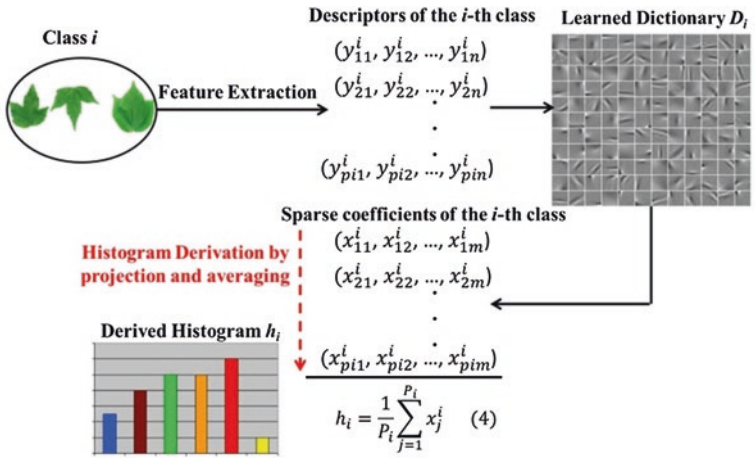


Fig. 5 Summarization of the learning stage of the proposed leaf image recognition framework via sparse representation

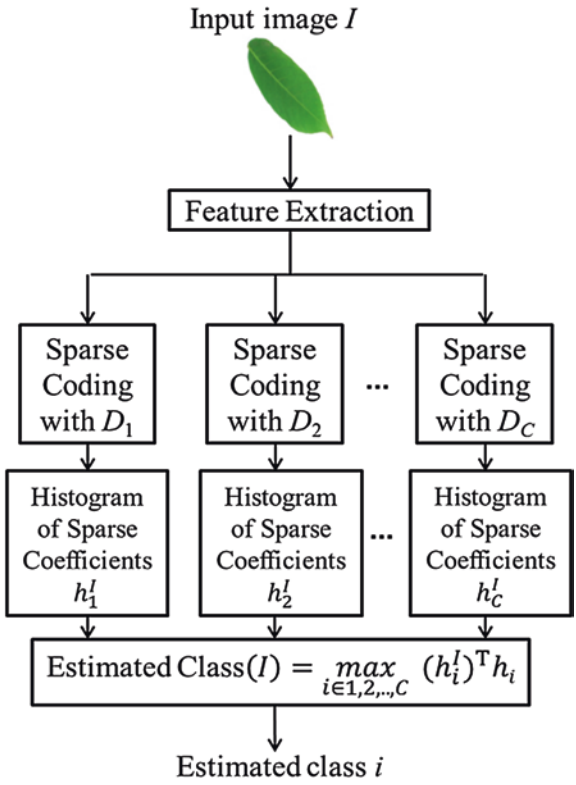


Fig. 6 The recognition stage of the proposed leaf image recognition framework via sparse representation

sparse coefficients $\{\theta_{ji} \in R^m\}_{j=1}^{q}$ with respect to D_i, $i = 1, 2, \ldots, C$, to find the sparse representation of I with respect to each class L_i as:

$$(\theta_{ji})^* = \arg\min_{\theta_{ji} \in R^m} \left(\frac{1}{2} \|Q_j - D_i \theta_{ji}\|_2^2 + \lambda \|\theta_{ji}\|_1 \right), \tag{5}$$

where $(\theta_{ji})^*$ denotes the solution minimizing (5). We then calculate the histogram h_i^I of I associated with its sparse coefficients $\{\theta_{ji}\}_{j=1}^{q}$ with respect to each D_i as:

$$h_i^I = \frac{1}{q} \sum_{j=1}^{q} \theta_{ji}. \tag{6}$$

Then, the class that the image I belongs to can be decided as:

$$\text{Estimated class}(I) = \arg \max_{i \in 1, 2, \ldots, C} = \left(h_i^I\right)^{\mathrm{T}} h_i. \tag{7}$$

That is, the class associated with the maximum correlation derived from (7) with the input image will be decided to be the class that the input image belongs to.

4 Proposed BoW Model-Based Leaf Image Recognition Framework Used for Comparison

To evaluate the performance of the proposed first framework via sparse representation, we present another framework based on well-known bag-of-words (BoW) image model. As illustrated in Fig. 7, in the learning stage of our BoW-based framework, we first extract a set of descriptors (or feature vectors) from all training leaf images of C classes for training a codebook consisting of K representative codewords. Based on the BoW image model [29], we apply the

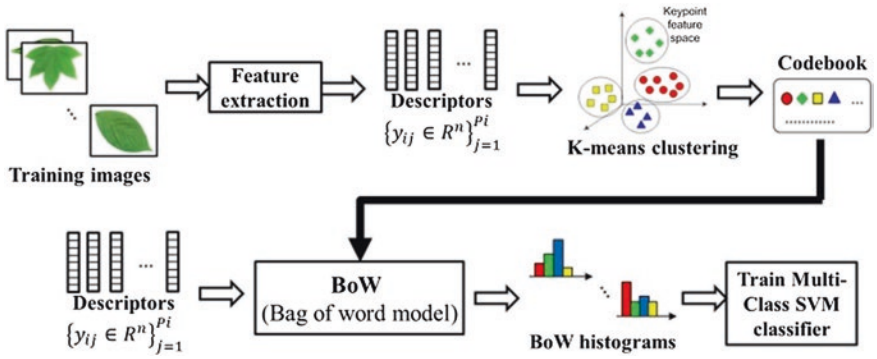


Fig. 7 An illustrated example of the learning stage in the proposed BoW-based leaf image recognition framework

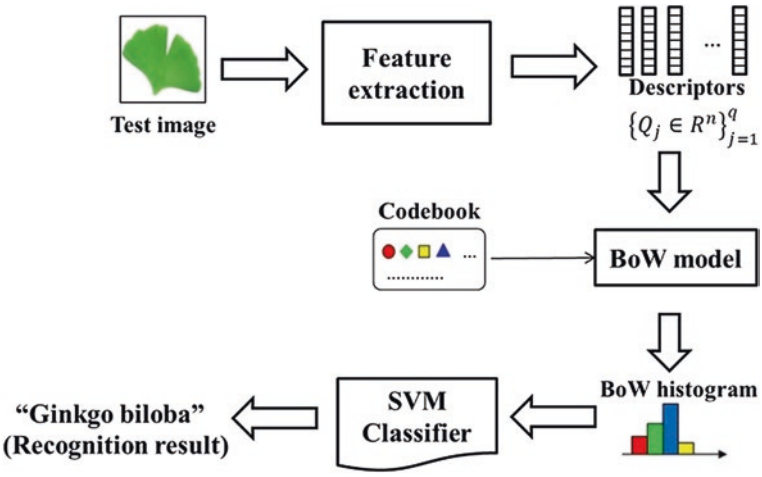


Fig. 8 An illustrated example of the recognition stage in the proposed BoW-based leaf image recognition framework

K-means clustering algorithm [35] to cluster all of the descriptors into *K* clusters with a cluster center for each cluster to form a codebook of *K* codewords. After obtaining the codebook, we quantize each training descriptor into its closest codeword in the codebook. Then, for each training image, we calculate a BoW histogram by counting the frequencies that each codeword is used. That is, each training image has been converted into a histogram. Then, we train a SVM classifier with *C* classes of leaf images.

As illustrated in Fig. 8, in the recognition stage of our BoW-based framework, for each test leaf image to be recognized, we first extract a set of descriptors and quantize each descriptor to its closest codeword in the codebook. Then, we can obtain the BoW histogram of this image. Finally, based on the trained SVM classifier, recognition of the image can be achieved. Comparisons between the proposed sparse representation-based approach and the proposed BoW-based approach in detail will be presented in Sect. 5.

5 Experimental Results

5.1 Experiment Settings

In this work, the two proposed recognition methods were implemented in MATLAB® R2013a (64 bits version) on a personal computer equipped with Intel® Core™ i5-2410M processor and 4 GB DDR2 memory. Moreover, the Matlab implementation of the employed K-SVD dictionary learning tool is available online from [26], while the Matlab interface implementation of the employed SVM classifier is also available online from [33].

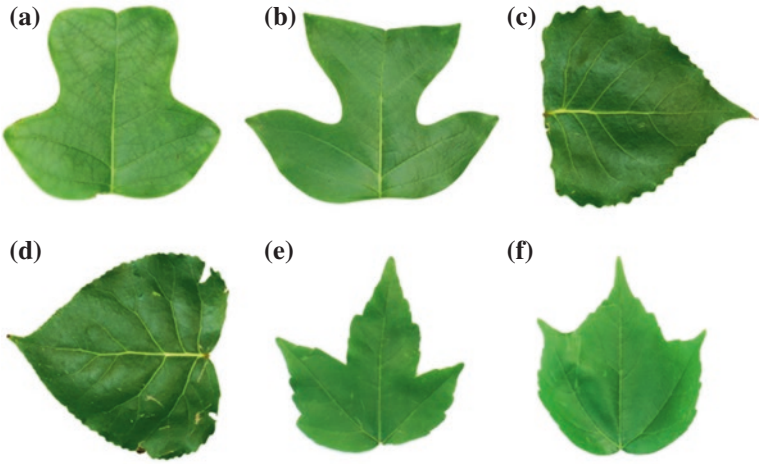


Fig. 9 Examples of leaf images in the dataset released by Wu et al. [9]: **a**, **b** Liriodendron chinense (Hemsl.) Sarg. (Chinese tulip tree); **c**, **d** Populus x canadensis Moench (Canadian poplar); and **e**, **f** Acer buergerianum Miq. (trident maple)

To evaluate the performances of the proposed leaf image recognition frameworks via sparse coding (denoted by Proposed-SC) and BoW (denoted by Proposed-BoW), respectively, we used the leaf image dataset, "*Flavia* leaf image dataset," released by Wu et al. [9] which has been a popular leaf image dataset for research purpose. The dataset consists of 32 classes of leaf images ($C = 32$), where each class contains 40–60 images, as illustrated in Fig. 9.

In the learning stage of our framework via SC, we randomly selected 30 images per class for dictionary learning, while the rest images were used for testing. For simplicity, the feature used was SIFT [34] with length $n = 128$ for each descriptor (or feature vector), which can be replaced by any shape, texture, or hybrid features. Moreover, for each learned dictionary D_i, we evaluated the four sizes (number of atoms) of 128, 256, 512, and 1,024 atoms ($m = 128$, 256, 512, 1,024), respectively. The sparsity and the number of training iterations used in the employed K-SVD dictionary learning algorithm [26] are empirically set to 12 and 100, respectively, to achieve the best tradeoff between recognition performance and computational complexity based on our experiments.

On the other hand, to fairly compare the proposed SC-based framework and our BoW-based framework described in Sect. 4, the evaluated codebook sizes for BoW are also set to 128, 256, 512, and 1,024, respectively, obtained by using the *K*-Means clustering algorithm [35] with 100 iterations based on the same training images and the SIFT feature [34]. The comparison of recognition performances between Proposed-SC and Proposed-BoW is shown in Fig. 10, where each data point was obtained by averaging the results of five runs.

Moreover, we also compare our method with the probabilistic neural network-based approach proposed in [9] (denoted by PNN), move median centers

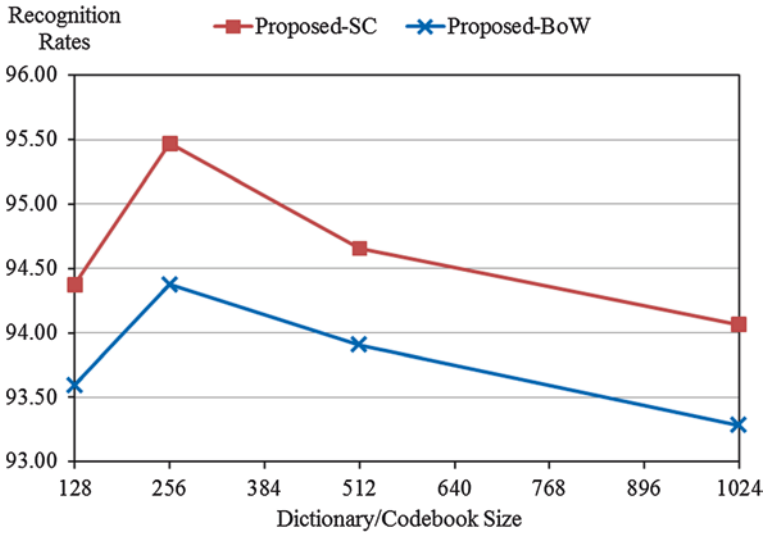


Fig. 10 Performance comparisons between Proposed-SC and Proposed-BoW methods

(MMC)-based approach proposed in [11] (denoted by MMC), hybrid feature with PNN-based approach proposed in [13] (denoted by HPNN), and combinatorial shape feature-based approach proposed in [14] (denoted by CShape), conducted on the same leaf image dataset.

6 Results

Table 1 lists the recognition rates obtained by PNN [9], MMC [11], Proposed-BoW, and Proposed-SC, respectively. In Table 1, the recognition rates obtained by Wu et al. [9] and Du et al. [11] were reported in [9], while those of Proposed-BoW and Proposed-SC were their respective best ones from Fig. 10 (the dictionary/codebook size is set to 256).

It can be observed from Fig. 10 that the proposed method via SC outperforms the proposed method via BoW used for comparison conducted on the evaluated

Table 1 Recognition rates of leaf images

Method	Rate (%)
PNN [9]	90.31
MMC [11]	91
HPNN [13]	93.75
CShape [14]	94.62
Proposed-BoW	94.38
Proposed-SC	95.47

dataset [9]. Compared with the BoW-based approach, the main advantage of our SC-based approach is that it is not required to re-train classifiers with newly leaf image class added, while in the BoW-based approach, both the codebook and the SVM classifier are required to be re-trained.

It can be also observed from Table 1 that both the proposed methods via SC and BoW outperform (or are comparable with) the four existing approaches used for comparisons (PNN [9], MMC [11], HPNN [13], and CShape [14]). Moreover, our methods only used 960 training images, while the PNN method [9] used 1,800 training images for neural network training. On the other hand, only single feature (SIFT [34]) is currently used in our methods to achieve these performances. More complex leaf image features may be properly integrated with our frameworks to achieve better performances.

In addition, to investigate some erroneous recognition cases, we illustrate an example in Fig. 11, where the two leaf images (Figs. 11a, b) of different species are identified as the same class. The main reason should be that only single feature was employed in this study without enough distinguishability. To cope this problem, hybrid features (e.g., integration of several shape and texture features) would be a solution.

6.1 Discussions

In this section, we discuss the main reason that the proposed sparse coding-based approach can outperform the proposed BoW-based approach, in terms of the inherent respective properties of the two image models. In the BoW-based framework, as depicted in Fig. 12a, each input image descriptor will be quantized into its closest codeword in the pre-trained codebook, which can be viewed as a special case of sparse representation, i.e., extremely sparse. That is, using the terminology of sparse coding, only one atom (with corresponding nonzero/significant coefficient of value "1") in the pre-learned dictionary is used to represent the input

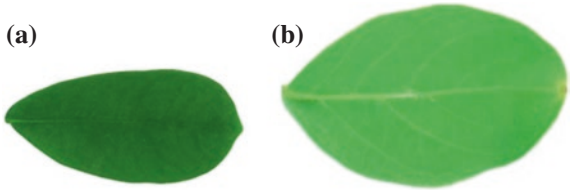


Fig. 11 An illustrated example of erroneous recognition: **a** the leaf species of Indigofera tinctoria L; and **b** the leaf species of Lagerstroemia indica (L.) Pers

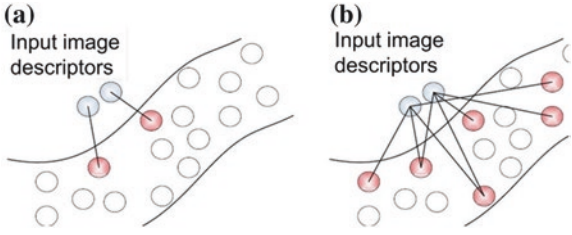


Fig. 12 An illustrated example of: **a** BoW-based; and **b** sparse coding-based frameworks

descriptor. On the other hand, as depicted in Fig. 12b, each input image descriptor will be represented by a few numbers of atoms (with corresponding nonzero/significant coefficients) in the pre-learned dictionary. Therefore, the sparse coding-based framework can provide richer representation for each image descriptor than the BoW-based framework, resulting in better recognition performance. To further achieve better performance with sparse coding, more constraints (e.g., locality [36] or group sparsity [37] conditions) may be incorporate into the regulation term to ensure that similar input image descriptors will have similar sparse representations based on a given dictionary.

7 Conclusions

In this chapter, we have proposed two leaf image recognition frameworks via sparse representation and BoW image model, respectively, and conducted the comparative studies between them. By learning dictionary for sparsely representing each species of leaf images, accurate recognition can be achieved. Besides better recognition performance obtained by our sparse representation-based framework, several unique characteristics benefited from the sparse coding theory include: (i) the proposed framework is adapted to newly added leaf species without retraining classifiers and suitable to be highly parallelized as well as integrated with any leaf image descriptors/features; and (ii) the proposed method would be robust to inaccurate feature extraction of leaf images, while providing more compact and richer representation for leaf images. For future works, we will integrate more advanced image features with our sparse representation-based framework and implement our system as a mobile application to achieve mobile visual search (e.g., [7, 8], as illustrated in Fig. 13). The proposed framework via sparse coding can be also extended to the applications of recognizing other types of images/videos.

Fig. 13 An example of mobile visual leaf image recognition/retrieval applications

References

1. Mzoughi, O., Yahiaoui, I., Boujemaa, N., Zagrouba, E.: Advanced tree species identification using multiple leaf parts image queries. In: Proceedings of IEEE International Conference on Image Processing, pp. 3967–3971, Melbourne, Sept 2013
2. Mouine, O., Yahiaoui, I., Verroust-Blondet, A.: Advanced shape context for plant species identification using leaf image retrieval. In: Proceedings of ACM International Conference on Multimedia Retrieval, June 2012
3. Mzoughi, O., Yahiaoui, I., Boujemaa, N.: Petiole shape detection for advanced leaf identification. In: Proceedings of IEEE International Conference on Image Processing, pp. 1033–1036, Orlando, FL, USA, Sept 2012
4. Yahiaoui, I., Mzoughi, O., Boujemaa, N.: Leaf shape descriptor for tree species identification. In: Proceedings of IEEE International Conference on Multimedia and Expo, pp. 254–259, Melbourne, July 2012
5. Mouine, S., Yahiaoui, I., Verroust-Blondet, A.: A shape-based approach for leaf classification using multiscale triangular representation. In: Proceedings of ACM International Conference on Multimedia Retrieval, pp. 127–134, Dallas, Texas, USA, Apr 2013
6. Caballero, C., Aranda, M.C.: Plant species identification using leaf image retrieval. In: Proceedings of ACM International Conference on Image and Video Retrieval, pp. 327–334, July 2010
7. Kumar, N., Belhumeur, P.N., Biswas, A., Jacobs, D.W., Kress, W.J., Lopez, I.C., Soares, J.V.B.: Leafsnap: a computer vision system for automatic plant species identification. In: Proceedings of European Conference on Computer Vision, pp. 502–516, Florence, Italy, Oct 2012
8. Mouine, S., Yahiaoui, I., Verroust-Blondet, A., Joyeux, L., Selmi, S., Goëau, H.: An android application for leaf-based plant identification. In: Proceedings of ACM International Conference on Multimedia Retrieval, Dallas, Texas, USA, Apr 2013
9. Wu, S.G., Bao, F.S., Xu, E.Y., Wang, Y.-X., Chang, Y.-F., Xiang, Q.-L.: A leaf recognition algorithm for plant classification using probabilistic neural network. In: Proceedings of IEEE International Symposium on Signal Processing and Information Technology, pp. 11–16, Giza, Egypt (the leaf image dataset available from http://sourceforge.net/projects/flavia/files/), Dec 2007
10. Hossain, J., Amin, M.A.: Leaf shape identification based plant biometrics. In: Proceedings of IEEE International Conference on Computer and Information Technology, pp. 458–463, Dhaka, Dec 2010
11. Du, J.-X., Wang, X.-F., Zhang, G.-J.: Leaf shape based plant species recognition. Appl. Math. Comput. **185**(2), 883–893 (2007)
12. Wang, X., Huang, D.-S., Du, J.-X., Xu, H., Heutte, L.: Classification of plant leaf images with complicated background. Appl. Math. Comput. **205**(2), 916–926 (2008)
13. Kadir, A., Nugroho, L.E., Susanto, A., Santosa, P.I.: Leaf classification using shape, color, and texture features. Int. J. Comput. Trends Technol. **1**(3), 225–230 (2011)
14. Sari, C., Akgul, C.B., Sankur, B.: Combination of gross shape features, fourier descriptors and multiscale distance matrix for leaf recognition. In: Proceedings of International Symposium on ELMAR, pp. 23–26, Zadar, Croatia, Sept 2013
15. Du, J.-X., Zhai, C.-M., Wang, Q.-P.: Recognition of plant leaf image based on fractal dimension features. Neurocomputing **116**, 150–156 (2013)
16. Wang, Z., Chi, Z., Feng, D.: Shape based leaf image retrieval. IEEE Proc. Vis. Image Sig. Process. **150**(1), 34–43 (2003)
17. Kebapci, H., Yanikoglu, B., Unal, G.: Plant image retrieval using color, shape and texture features. Comput. J. **54**(9), 1475–1490 (2011)
18. Fotopoulou, F., Laskaris, N., Economou, G., Fotopoulos, S.: Advanced leaf image retrieval via multidimensional embedding sequence similarity (MESS) method. Pattern Anal. Appl. **16**(3), 381–392 (2013)

19. Wagner, A., Wright, J., Ganesh, A., Zhou, Z., Mobahi, H., Ma, Y.: Toward a practical face recognition system: robust alignment and illumination by sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **34**(2), 372–386 (2012)
20. Guha, T., Ward, R.K.: Learning sparse representations for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **34**(8), 1576–1588 (2012)
21. Huang, D.-A., Kang, L.-W., Wang, Y.-C.F., Lin, C.-W.: Self-learning based image decomposition with applications to single image denoising. IEEE Trans. Multimedia **16**(1), 83–93 (2014)
22. Chen, D.-Y., Chen, C.-C., Kang, L.-W.: Visual depth guided color image rain streaks removal using sparse coding. IEEE Trans. Circuits Syst. Video Technol. **24**(8), 1430–1455 (2014)
23. Kang, L.-W., Lin, C.-W., Fu, Y.-H.: Automatic single-image-based rain streaks removal via image decomposition. IEEE Trans. Image Process. **21**(4), 1742–1755 (2012)
24. Yeh, C.-H., Kang, L.-W., Chiou, Y.-W., Lin, C.-W., Fan Jiang, S.-J.: Self-learning-based post-processing for image/video deblocking via sparse representation. J. Vis. Comm. Image Rep. **25**(5), 891–903 (2014)
25. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. J. Mach. Learn. Res **11**, 19–60 (2010)
26. Aharon, M., Elad, M., Bruckstein, A.M.: The K-SVD: an algorithm for designing of over-complete dictionaries for sparse representation. IEEE Trans. Sig. Process. **54**(11), 4311–4322 (Matlab source code available from http://www.cs.technion.ac.il/~ronrubin/software.html) (2006)
27. Bruckstein, A.M., Donoho, D.L., Elad, M.: From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Rev. **51**(1), 34–81 (2009)
28. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature **381**(13), 607–609 (1996)
29. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Proceedings of ECCV International Workshop on Statistical Learning in Computer Vision, Prague (2004)
30. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, pp. 2169–2178 (2006)
31. Yang, J., Jiang, Y.-G., Hauptmann, A.G., Ngo, C.-W.: Evaluating bag-of-visual-words representations in scene classification. In: Proceedings of ACM Multimedia Information Retrieval, pp. 197–206 (2007)
32. Hsu, C.-Y., Kang, L.-W., Liao, H.-Y.M.: Cross-camera vehicle tracking via affine invariant object matching for video forensics applications. In: Proceedings of IEEE International Conference on Multimedia and Expo, San Jose, CA, USA, July 2013
33. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), article no. 27 (Matlab interface source code available from http://www.csie.ntu.edu.tw/~cjlin/libsvm/), Apr 2011
34. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis **60**(2), 91–110 (2004)
35. Hartigan, J.A., Wong, M.A.: A k-means clustering algorithm. Appl. Stat. **28**(1), 100–108 (1979)
36. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, pp. 3360–3367, San Francisco, CA, USA, June 2010
37. Tsai, C.-Y., Huang, D.-A., Yang, M.-C., Kang, L.-W., Wang, Y.-C.F.: Context-aware single image super-resolution using locality-constrained group sparse representation. In: Proceedings of IEEE Visual Communications and Image Processing Conference, San Diego, CA, USA, Nov 2012

Massively Parallel Feature Selection Based on Ensemble of Filters and Multiple Robust Consensus Functions for Cancer Gene Identification

Anouar Boucheham and Mohamed Batouche

Abstract Currently, cancer prevails as a prime health matter worldwide. Selecting the appropriate biomarkers for early cancer detection might improve patient care and have often driven revolutions in medicine. Statistics and machine learning techniques have been broadly investigated for biomarker identification, especially feature selection where researchers try to identify the most distinguishing genes that can achieve better predictive performance of cancer subtypes. The robustness of the selected signature remains a crucial goal in personalized medicine. Ensemble and parallel feature selection are promising techniques to overcome this problem in which they have seen an increasing use in biomarker discovery. We focus in this chapter on the principal aspects of using ensemble feature selection in biomarker discovery. Furthermore, we propose a massively parallel meta-ensemble of filters (MPME-FS) to select a robust and parsimonious subset of genes. Two types of filters (ReliefF and Information Gain) are investigated in this study. The performances of the proposed approach in terms of robustness, classification power and the biological meaning of the selected signatures on five publicly available cancer datasets are explored. The results attest that the MPME-FS approach can effectively identify a small subset of biomarkers and improve both robustness and classification accuracy.

Keywords Bioinformatics · Biomarker discovery · Ensemble feature selection · Meta learning · ReliefF · Information gain · Cancer classification

A. Boucheham (✉) · M. Batouche
Computer Science Department, College of NTIC, Constantine 2 University,
MISC Laboratory, 25000 Constantine, Algeria
e-mail: anouar.boucheham@univ-constantine2.dz

M. Batouche
e-mail: mohamed.batouche@univ-constantine2.dz

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_6

1 Introduction

Over the last two decades, the general rate of deaths to new cancer cases persists as high as 49 % overall. Thus, current bioinformatics efforts are focusing on biomarker discovery which is the key element of personalized medicine, where the genetic constitution is used to guide therapeutic approaches [1]. Therefore, the discovery of more effective cancer biomarkers is urgently needed, since the development and the effective use of biomarkers in clinical practice will certainly lead to tailor treatments for the disease in an individual [2].

The application of omics high-throughput technologies for cancer biomarker discovery is being rapidly expanded in current biomedical research, including DNA microarrays, Next Generation Sequencing (NGS) and MicroRNAs which are able to capture a substantial fraction of a cell state [3]. These technologies allow monitoring the expression levels of thousands of genes simultaneously in healthy and diseased cells, as well as are essential to biomarker discovery.

Gene expression data can effectively help to differentiate between cancer subtypes and then serve as an effective tool for diagnostic purposes in clinical practice. However, the identification of the smallest possible set of genes that could be used as biomarkers is a crucial problem in bioinformatics and personalized medicine. These genes must be the most informative for cancer prediction through supervised classification models [4]. The identification is generally referred to as a feature selection problem which is desirable to provide the features that contribute most to both classification and prediction [5]. It is a vital preprocessing step in data mining tasks, to reduce the effect of noise and improve the quality of data processing as well as considered to be one of current challenges in statistical machine learning for high-dimensional data.

A major challenge in the analysis of gene expression data is due to their sizes: a very small number of samples, of the order of tens, versus thousands of genes associated to all samples. This is commonly known as the "curse-of-dimensionality" which is also characterized by a large number of irrelevant, redundant and noisy genes that mislead or impede diagnosis efficiency [6]. Thus, only a fraction of genes contains useful biological interpretations and further gives a high accuracy for cancer diagnosis. Another challenge concerns the biological variations in real clinical tests which require the development of more stable feature selection methods [7]. In other words, selection of informative genes and an appropriate assessment of robustness, classification accuracy and biological meaning of the results are the most important matters in this field.

Ensemble-based learning is a robust and popular technique, due to the immense success of many ensemble methods in bioinformatics applications. It has the broad advantage of overcoming the curse-of-dimensionality in gene expression data, thereby offer higher accuracy and stability than conventional feature selection algorithm can achieve. Therefore, the use of ensemble methods to feature selection problem has been one of the recent growing trends. It consists of performing multiple diverse selectors with different subsamples, and then aggregates their results using a consensus function to obtain a final best subset of biomarkers [8]. Another benefit of applying ensemble feature selection, that it is naturally susceptible to parallelism, as well as we can easily undertake their parameters in parallel. The parallel

implementation of ensemble methods can certainly speedup the computational time of the selection and allow solving large-scale problems by involving multiprocessors to execute the different parts of the ensemble in parallel [9].

This chapter will focus on the different aspects of the application of ensemble feature selection methods to biomarker discovery from gene expression data. Furthermore, we propose a massively parallel meta-ensemble based feature selection method that can select robust and accurate biomarkers from DNA-microarrays datasets and can be generalized to several genomics studies. Two types of filter-based feature selection algorithms are investigated in this study: ReliefF and Information Gain. We also discuss the results in terms of robustness, classification power and the biological meaning of the selected signatures.

2 Application of Ensemble Feature Selection to Biomarker Discovery

In analogy with ensemble methods in supervised machine learning which combine multiple learned models to achieve high classification accuracy such as bagging and boosting [10]. Ensemble feature selection has received much attention recently. We mainly present here the different aspects to be considered in ensemble-based feature selection for biomarker discovery in which can help researchers to classify any method of them. The main critical problems in this category of methods are both the construction of diverse local selectors and the consensus function used to combine the different subsets of features [7, 11]. Therefore, the first aspect to be examined is the diversity design within the ensemble. This criterion divides ensemble feature selection methods into three classes:

- Ensemble based on data diversity: where we run the same selector with different subsamples generated from the original dataset [12, 13].
- Ensemble based on functional diversity: where different selectors are performed on the whole set of data (without sampling) [14].
- Ensemble based on data and functional diversity: here both data and functional diversity are combined in which multiple feature selection algorithms are performed on different subsamples.

Another aspect to be considered in ensemble feature selection methods is the representation used by the different selectors, since the notation of the results is not the same in all feature selection algorithms. This has a great impact on the consensus function to be used to aggregate the results [12]. Typically, we can observe three types of representations:

- Feature subset representation: subset containing only selected features (generally with different size)
- Feature ranking representation: subset of ranked features (a threshold is necessary for the selection)
- Feature weighting representation: a subset of pairs feature/weight which can easily converted to feature ranking representation.

Recent studies have focused on ensemble methods using wrapper-based selectors [15, 16]. It prompts us to consider the dependence of the selectors to any classifier as an important aspect in ensemble feature selection methods. This criterion influences the quality of solutions within the ensemble and the overall computational cost of the selection, as well as it divides ensemble feature selection methods into:

- Filter/ranking-based ensembles: they are simple, fast and independent of any classifier [14].
- Wrapper-based ensembles: they are very computationally intensive and have the risk of over-fitting due to high dimensionality of data as well as include the interaction between feature subset search and the mining algorithm. Moreover, they have the ability to take into account feature dependencies [17].
- Embedded-based ensembles: they use internal information of the classifier to perform selection and show a better computational complexity than wrapper methods [13].
- Hybrid-based ensembles: they are a combination of filter and wrapper methods which use the ranking information obtained using filters to guide the search in the optimization algorithms used by wrapper methods [16].

3 Massively Parallel Meta-Ensemble Feature Selection

Feature selection is an important preprocessing step in many machine learning applications including bioinformatics and computational biology, where it is generally used to find the smallest subset of features that extremely increases the performance of the classification model. In this section, we focus on ensemble of ensembles learning techniques which work by aggregating the outcomes of different ensembles into a final agreed decision through one or more consensus functions. The main objective is to attempt high performance of computer-aided diagnosis (CAD), by selecting a few genes with high predictive power and high sensibility to variations in real clinical tests. The selected biomarkers will be directly used by the CAD system for cancer diagnosis or others predictive goals.

For this purposes, a new parallel framework of feature selection is explored (see Fig. 1). In analogy with meta-ensemble models for supervised learning [10], the proposed approach is designed as an ensemble of ensembles of different selectors which perform selection in parallel through various ensembles and two consensus functions. In the following, we introduce our parallel framework for biomarker discovery in detail. We first, formulate both problem and representation of our solution under the proposed framework and then the general framework is explored including the parallel construction of the ranked lists as well as the related consensus functions.

Accordingly, biomarker discovery from gene expression is the problem of selecting subset of representative biomarkers from a large dataset. Given a set X of

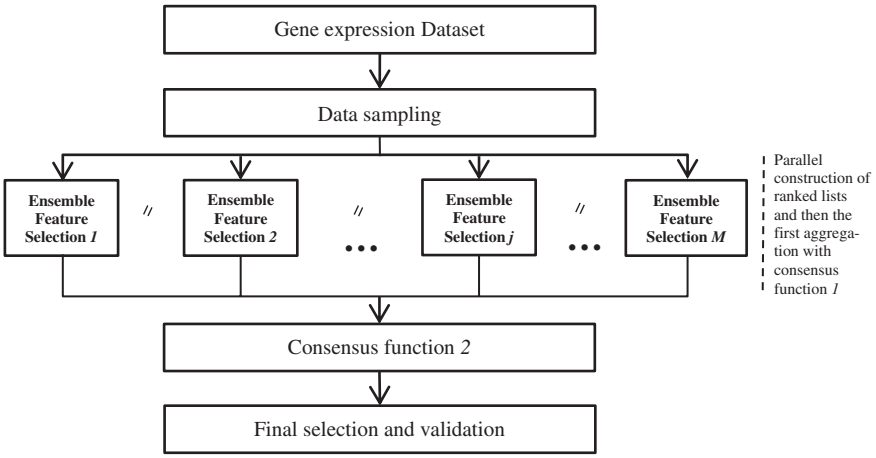


Fig. 1 Parallel model of the MPME-FS

K features with K very large, the problem consists in finding out the minimal subset $X_s^* \subset X$ that contains the more relevant yet non redundant features. Ensemble feature selection is a promising technique for addressing these complex structures of data and alleviates the problems of small sample size and high dimensionality [18].

The use of an ensemble of ensembles of filters leads to several subsets. Let us denote by X_{sj}^i the subset j of selected features using filter i. Therefore, two matters need to be addressed. The first one is related to the importance of each feature and the second one is related to the way subsets are aggregated to lead to the final subset of features. In order to properly deal with these two issues, we propose a two stage approach that uses ranking and consensus functions. At a first step various subsets of ranked lists of features are constructed using several filters then aggregation of these subsets is performed at two levels to form ensembles and then the meta-ensemble. More formally, the output of the first step can be represented as:

$$X_{sj}^i = \left\{ \left(f_{ij}^k, w_{ij}^k \right) \text{ where } i,j = 1 \ldots (N,M) \text{ and } k = 1 \ldots K \right\} \tag{1}$$

f_{ij}^k represents the rank of the feature k in the ensemble j using filter i. Its relevance is given by the weight w_{ij}^k. The global weight of feature k within the ensemble j is denoted as w_j^k. Three subsets of pairs of features and their weights are needed.

- The first is $Lbest_j$ that represents the local best features in each ensemble.

$$Lbest_j = \left\{ \left(k, w_j^k \right) \text{ where } j = 1 \ldots M \text{ and } k = 1 \ldots K \right\} \tag{2}$$

- This subset is the result of an aggregation process over $\mathrm{X}_{\mathrm{sj}}^{i=1\ldots N}$ subsets.

- The second is *Gbest*. It represents the global best features over the meta-ensemble. It is the result of an aggregation process over $Lbest_j$.
- The third is $Fbest = X_s^* \subset X$. It represents the final selected features given by MPME-FS method.

3.1 General Framework of MPME-FS for Biomarker Discovery

The general framework of MPME-FS consists of multiple ensembles of filters performed in parallel each of which employs a robust consensus function to select the best subset within each ensemble. The next step is to aggregate the outcome of all ensembles using a second consensus function and finally select features with higher scores given by all filters from all ensembles as shown in Fig. 1. The selection processes starts by the construction of M sub-samples $Si = 1 \ldots M$ from the whole dataset. Then, the parallel selection is initiated in all the M ensembles. At this stage, each ensemble j constructs N ranked lists X_{sj}^i by using filter i. To achieve the goal of both functional and data diversity when constructing the X_{sj}^i lists, we have used data partitioning with overlap allowing creating a reduced dataset to each $filter_i$.

Data perturbation involves generating subsamples by removing instances from the original datasets randomly. Knowing that, the overlap represents the percentage of samples belonging to the original dataset [11]. Subsequently, a consensus function within each ensemble is applied in order to aggregate these ranked lists X_{sj}^i and finally obtain the local best features in the ensemble j ($Lbest_j$) (see Fig. 2). Note that the construction of the ranked lists within each ensemble can be performed in parallel and through different filters (Information Gain, Gain ratio, Fisher Ratio, Symmetric uncertainty, ReliefF).

The following step consists of the aggregation of all local best features over the meta-ensemble, to construct the global best subset (Gbest) of features using the consensus function 2 alongside with the accumulation of scores associated

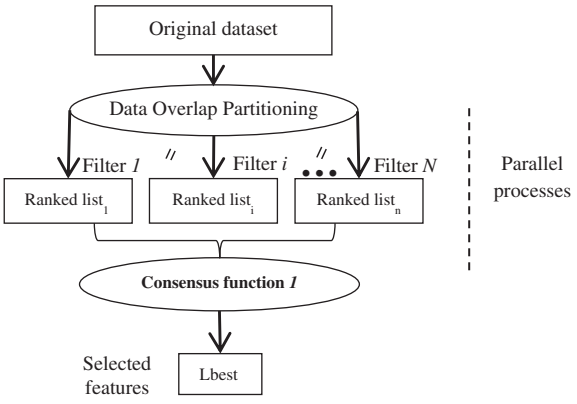


Fig. 2 Ensemble feature selection

to these features. Finally, we select the best ranked features which constitute the subset *Fbest*, from the subset *Gbest* ranked based on features' global weights

3.2 Consensus Functions

Recent work in biomarker identification has seen an increasing use of ensemble based feature selection due to their power to give higher accuracy and stability than a single algorithm can achieve. It can also dealing with small sample size and complex data structures. The key idea in ensemble methods is how to combine subsets of different selectors to lead to the final subset of features. This problem has received considerable attention in recent years [19]. Aggregation methods depend on the representation of the outcome of selectors which can be divided in three types: feature subset, feature ranking and feature weighting-score [11]. Based on these representations there exist many consensus functions, some of them include weighted voting, mean aggregation and threshold based aggregation for both rank and weighting-score representations, and counting the most frequently selected features for feature subset representation [12, 20].

Certainly, choosing the appropriate consensus function is a difficult task in ensemble methods. In our work, we use two consensus functions; the first one in the ensembles level and the second function is in the meta-ensemble level. Both are based on features ranking and their global weights which lead to a more robust and parsimonious final selection.

The first consensus function aggregates the ranked lists created by filters in the same ensemble. This function is inspired from both counting the most frequently selected features and weighted voting aggregation functions, but with hard selection by using the intersection over the entire ranked lists. For the intersection purpose, we use a threshold denoted by *TS1* in order to select only features belonging to *TS1* first ranked ones. Afterward, the weights of all selected features over the entire ensemble j denoted by $w_{i,j}^{k}$ are accumulated in order to obtain the global weights of selected features w_{j}^{k} over all ensembles [18]. More formally, the consensus function 1 can be described as follows:

$$\begin{cases} Lbest_j = \{\left(\boldsymbol{k}, \boldsymbol{w_j^k}\right)\} = \begin{cases} \bigcap_{i=1}^{N} \{\left(\boldsymbol{f_{i,j}^k}, \boldsymbol{w_{i,j}^k}\right)\} \\ \text{and } \boldsymbol{f_{i,j}^k} \leq TS1 \end{cases} \\ \text{where } \boldsymbol{w_j^k} = \sum_{i=1}^{N} w_{i,j}^{k} \end{cases}$$

By this way, we obtain the set *Lbestj* containing pairs of the best selected features with their global weights in the ensemble j $\{\left(k,\ w_{\mathrm{j}}^{\mathrm{k}}\right)\}$. This latter will be the input of the second consensus function in the meta-ensemble level, to construct the subset Gbest which contribute to the final selection.

The second consensus function consists primarily of aggregation the M subsets ($Lbest_j$) generated in the parallel previous step. The *Gbest* subset represents pairs

of features and their accumulated weights belonging to the union of the M local best subsets of features, which can be calculated as follows:

$$\begin{cases} Gbest = \left\{ \left(\boldsymbol{k}, \boldsymbol{bw^k} \right) \right\} = \bigcup_{j=1}^{M} Lbest_j \\ \text{where } \boldsymbol{bw^k} = \sum_{j=1}^{M} w_j^k \end{cases}$$

Finally, we select the best *TS2* ranked features from *Gbest* that represent the final selected features to be validated in the validation step. The pseudo-code of the whole process can be summarized as follows:

Algorithm: Massively Parallel Meta-Ensemble Feature Selection (**MPME-FS**).

Parameters:
N: ensemble size
M: meta-ensemble size
TS1: threshold of intersection
TS2: percentage of best selected features in meta-ensemble
Overlap: overlap of data sampling

Input:
D: dataset with *K* features
L: sample labels in *D*

Output:
Fbest: final best selected features

Parameters initialization:
Gbest, **Lbest** =Ø; ***Filter***= filter_i;

Parallel Meta ensemble feature selection process:

1: For (each ensemble_j / j=1 ….M) **do in parallel**

// determining Lbest_j for each ensemble_j

2: For (i=1…N) **do in parallel**
3: S_i = sampling (D, *Overlap*)
4: $(k, w_{i,j}^k)$ = *Filter* (S_i, L)
5: End For in parallel

// obtaining the rank $f_{i,j}^k$ of each feature in the ensemble_j

6: $\{(f_{i,j}^k, w_{i,j}^k)\}$ = Sort ($\{(k, w_{i,j}^k)\}$)

// aggregating results of all filters within the ensemble_j

7: ***Lbest***$_j$ = $\{(k, w_j^k)\}$ = **Consensus function 1** ($\{(f_{i,j}^k, w_{i,j}^k)\}$)

8: End For in parallel

// aggregating all *Lbest*$_j$ subsets

9: ***Gbest*** = **Consensus function 2** (***Lbest***$_j$)

// final selection

10: Sort (***Gbest***) // sort k based on bw^k
11: ***Fbest*** = select *TS2* best features from ***Gbest***

---

4 Experiments and Discussions

In the following sections, the analysis of classification performances, robustness and biological interpretation of the MPME-FS method on large feature and small sample size microarrays are presented. First, the data sets and the experimental settings used in this analysis are briefly described. Second, we analyze the classification performances in terms of accuracy, sensitivity and specificity using different classifiers. After that, we study the robustness of the selected signatures. Finally, we perform a biological interpretation of the selected genes.

4.1 Datasets and Experiment Setting

All experiments were conducted using MATLAB®'s Parallel Computing Toolbox (PCT). The proposed MPME-FS was evaluated by means of five publicly available DNA microarray datasets which can be divided into binary and multiclass types. The binary datasets are the most prominent and can separate healthy patients from cancer patients, while multiclass datasets are used to differentiate the various types of cancers based on gene expressions. Therefore, the datasets were collected from both Kent Ridge bio-medical data repository[1] and Gene Expression Model Selector, from Vanderbilt University[2]. The main datasets characteristics are shown in Table 1.

To assess the performances of our parallel meta-ensemble feature selection method, we use in the experiments two well-known and successful filters: information gain and ReliefF. Based on an empirical evaluation using different settings of the proposed method, the best parameters setting of MPME-FS which is adopted in this study is depicted in Table 2.

4.2 Classification Accuracy Analysis

The first experiment is devoted to assess the performance of the MPME-FS in terms of accuracy, sensitivity, specificity and the number of selected biomarkers using 10-fold cross validation technique. The latter is a common choice in the specialized literature [21], which splits the whole set of data into many subsets to evaluate the goodness of the selected signature.

To achieve high level evaluation of the classification ability of the selected genes, we first use different classifiers separately (SVM, KNN, ANN). Then we have employed an ensemble of different classifiers (SVM, KNN and ANN) with

[1] http://levis.tongji.edu.cn/gzli/data/mirror-kentridge.html.

[2] http://www.gems-system.org.

Table 1 Characteristics of the different datasets used for evaluation

Dataset	#Features	#Classes	#Samples
Ovarian	15,154	2	253
Leukemia	7,129	2	72
DLBCL	5,469	2	77
Colon	2,000	2	62
SRBCT	2,308	4	83

Table 2 MPME-FS parameters setting

Filters	InfoGain, ReliefF
Ensemble size	10
Meta-ensemble size	100
TS1	150
TS2	30
Overlap	80
K value in ReliefF	10

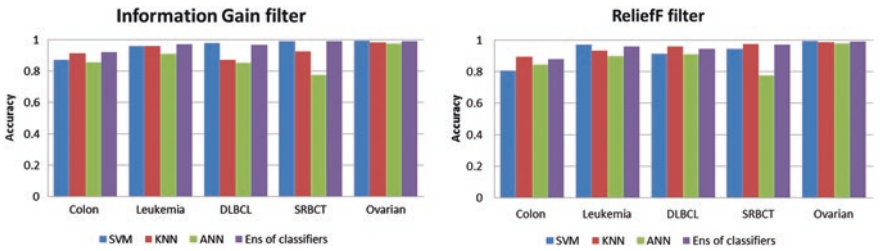


Fig. 3 Average 10 cross-validation classification accuracy using: SVM, KNN, ANN and an ensemble of different classifiers over the five datasets

majority voting as consensus function. Accordingly, the results indicated in Fig. 3 represent the average accuracies of MPME-FS given by SVM, KNN, ANN and the ensemble of classifiers described above. For comparison reasons, we have used in this experiment both Information Gain and ReliefF filters to perform selection by the proposed approach. Form this figure we observe that the MPME-FS performs better using information Gain filter than ReliefF filter in the five datasets.

A second observation which can be made is that the ensemble of classifier gives higher accuracy than single classifiers in almost cases that are not surprising since it combines the efforts of the three classifiers. Furthermore, Fig. 4 shows boxplots of MPME-FS using SVM classifier over thirty runs for both Information Gain and ReliefF (Fig. 4a, b successively). As desired, the performance variance between runs reaches an almost completely stable result through the two filters (among 0.001 and 0.04).

We also provide in Table 3 the average performance of MPME-FS using the different classifiers of both information Gain and ReliefF filters. In the last column

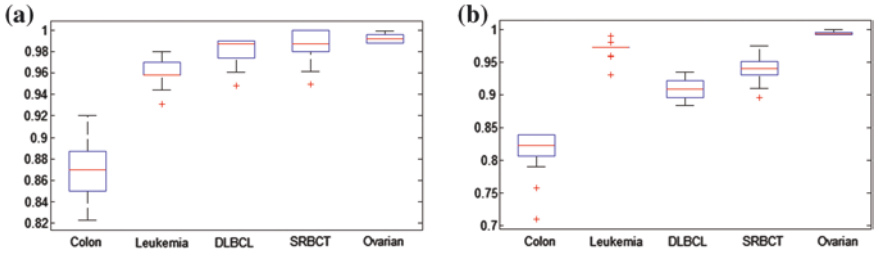


Fig. 4 Boxplots of MPME-FS method on colon, leukemia, DLBCL, SRBCT and ovarian datasets across 30 runs. **a** Informain gain filter and **b** ReliefF filter

Table 3 Average classification results in terms of sensitivity (sensi), specificity (speci) and the number of selected biomarkers (# genes) of MPME-FS using both information gain and ReliefF filters over the five datasets

		SVM		KNN		ANN		Ensemble of classifiers		# genes
		Sensi	*Speci*	*Sensi*	*Speci*	*Sensi*	*Speci*	*Sensi*	*Speci*	
InfoGain	Colon	0.875	0.804	0.925	0.907	0.871	0.831	0.914	0.909	30
	Leukemia	0.957	1	0.978	0.96	0.934	0.92	0.953	1	31
	DLBCL	0.965	1	0.827	1	0.924	0.63	0.952	1	27
	SRBCT	1	0.94	0.896	1	0.89	0.92	1	0.963	32
	Ovarian	0.998	1	0.993	0.967	1	0.978	1	0.988	39
Average		**0.959**	**0.948**	**0.923**	**0.966**	**0.923**	**0.855**	**0.963**	**0.972**	**31**
ReliefF	Colon	0.875	0.81	0.925	0.863	0.9	0.818	0.89	0.863	31
	Leukemia	0.878	0.96	1	0.84	0.872	0.88	1	0.96	33
	DLBCL	0.931	0.947	0.965	0.947	0.948	0.842	0.948	0.947	27
	SRBCT	0.931	0.944	0.965	1	0.931	0.925	0.965	0.981	33
	Ovarian	1	1	1	0.978	1	0.978	1	0.989	41
Average		**0.923**	**0.932**	**0.971**	**0.925**	**0.9302**	**0.888**	**0.960**	**0.948**	**33**

the number of selected biomarkers on the five datasets is shown. We observe that both sensitivity and specificity of our selection are convergent among the different classifiers. The previous experiments were performed on 30 independent runs to have statistically meaningful conclusions as our approach is stochastic.

4.3 Robustness Analysis

We explore and discuss in the present study the robustness of the selected signature by the MPME-FS approach. Therefore, we assess the similarity between the outputs of different independent executions of our method. The global stability is

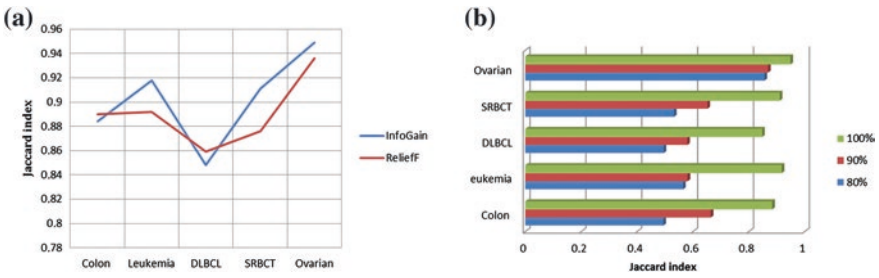


Fig. 5 Average robustness results of the MPME-FS in term of Jaccard index over 20 independent runs on the five datasets. **a** Information Gain filter versus ReliefF filter **b** Jaccard index versus perturbation rate of subsampling between the independent runs (80, 90 and 100 %)

defined as the average over all pairwise similarity comparisons between the different feature selectors as follows [12]:

$$S_{tot} = \frac{2\sum_{i=1}^{k}\sum_{j=i+1}^{k} S(f_i, f_j)}{k(k-1)} \tag{3}$$

where f_i represents the outcome of the feature selection method applied to subsample$_i$ ($1 \leq i \leq 20$), and $S(f_i, f_j)$ represents a similarity measure between f_i and f_j. Mainly, for feature subsets selection (as in our case), we use the Jaccard index (JI) which can be calculated as follows:

$$S(f_i,f_j) = \frac{|f_i \cap f_j|}{|f_i \cup f_j|} \tag{4}$$

A set of experiment assess the overall stability of the selected signature on the five datasets using both Information Gain and ReliefF filters which is shown in the Fig. 5a. Results in term of Jaccard index show that the MPME-FS performed using information gain is generally more robust over the most datasets. To provide a better robustness analysis, we assess the effect of data perturbation rate when creating subsamples on the stability of the signature. In this experiment we use Information Gain as filter over the five datasets of which the results can be seen in Fig. 5b, which indicates that the robustness decreases as the perturbation rate is decreased.

4.4 Biological Interpretation of the Results

In this section, we address biological analysis of the selected biomarkers. We focus in this experiment on the analysis of the selected biomarkers from Colon and Leukemia datasets which are widely studied in the literature. Accordingly, Tables 4 and 5 list and describe the top thirty ranked genes over 30 independent

Table 4 Description of the top thirty selected genes from colon dataset, with a complete frequency level (freq = 30) over 30 independent runs

Gene index	Accession number	Gene description
66	T71025	3′ UTR 1 84103 Human (HUMAN)
1423	J02854	Gene 1 "MYOSIN REGULATORY LIGHT CHAIN 2, SMOOTH MUSCLE ISOFORM (HUMAN); contains element TAR1 repetitive element"
1414	R64115	3′ UTR 2a 139618 ADENOSYLHOMOCYSTEINASE (Homo sapiens)
137	D25217	Gene 1 "Human mRNA (KIAA0027) for ORF, partial cds"
138	M26697	Gene 1 "Human nucleolar protein (B23) mRNA, complete cds"
241	M36981	Gene 1 "Human putative NDP kinase (nm23-H2S) mRNA, complete cds"
245	M76378	Gene 1 "Human cysteine-rich protein (CRP) gene, exons 5 and 6"
249	M63391	Gene 1 "Human desmin gene, complete cds"
267	M76378	Gene 1 "Human cysteine-rich protein (CRP) gene, exons 5 and 6"
1843	H06524	3′ UTR 1 44386 "GELSOLIN PRECURSOR, PLASMA (HUMAN)"
286	H64489	3′ UTR 2a 238846 LEUKOCYTE ANTIGEN CD37 (Homo sapiens)
365	X14958	Gene 1 Human hmgI mRNA for high mobility group protein Y
377	Z50753	Gene 1 H.sapiens mRNA for GCAP-II/uroguanylin precursor
1960	D59253	Gene 1 Human mRNA for NCBP interacting protein 1
493	R87126	3′ UTR 2a 197371 "MYOSIN HEAVY CHAIN, NONMUSCLE (Gallus gallus)"
513	M22382	Gene 1 MITOCHONDRIAL MATRIX PROTEIN P1 PRECURSOR (HUMAN)
625	X12671	Gene 1 Human gene for heterogeneous nuclear ribonucleoprotein (hnRNP) core protein A1
739	X12369	Gene 1 "TROPOMYOSIN ALPHA CHAIN, SMOOTH MUSCLE (HUMAN)"
897	H43887	3′ UTR 2a 183264 COMPLEMENT FACTOR D PRECURSOR (Homo sapiens)
765	M76378	Gene 1 "Human cysteine-rich protein (CRP) gene, exons 5 and 6"
780	H40095	3′ UTR 1 175181 MACROPHAGE MIGRATION INHIBITORY FACTOR (HUMAN)
812	Z49269	Gene 1 H.sapiens gene for chemokine HCC-1
964	T86473	3′ UTR 1 114645 NUCLEOSIDE DIPHOSPHATE KINASE A (HUMAN)
1042	R36977	3′ UTR 1 26045 P03001 TRANSCRIPTION FACTOR IIIA
1411	H77597	3′ UTR 1 214162 H.sapiens mRNA for metallothionein (HUMAN)
1494	X86693	Gene 1 H.sapiens mRNA for hevin like protein
1582	X63629	Gene 1 H.sapiens mRNA for p cadherin
1635	M36634	Gene 1 "Human vasoactive intestinal peptide (VIP) mRNA, complete cds"
1771	J05032	Gene 1 "Human aspartyl-tRNA synthetase alpha-2 subunit mRNA, complete cds"
1263	T40454	3′ UTR 2a 60221 ANTIGENIC SURFACE DETERMINANT PROTEIN OA3 PRECURSOR (Homo sapiens)

Table 5 Description of the top thirty selected genes from Leukemia dataset, with a complete frequency level (freq = 30) over 30 independent runs

Gene index	Accession number	Gene description
758	**D88270_at**	GB DEF = (lambda) DNA for immunoglobin light chain
760	D88422_at	CYSTATIN A
1144	**J05243_at**	SPTAN1 Spectrin, alpha, nan-erythrocytic 1 (alpha-fodrin)
1630	L47738_at	Inducible protein mRNA
1685	**M11722_at**	Terminal transferase mRNA
1834	**M23197_at**	CD33 CD33 antigen (differentiation antigen)
1882	**M27891_at**	CST3 Cystatin C (amyloid angiopathy and cerebral hemorrhage)
1902	M29474_at	Recombination activating protein (RAG-1) gene
2121	**M63138_at**	CTSD Cathepsin D (lysosomal aspartyl protease)
2128	M63379_at	CLU Clusterin (complement lysis inhibitor; testosterone-repressed prostate message 2; apolipoprotein J)
2288	**M76559_at**	Neuronal DHP-sensitive, voltage-dependent, calcium channel alpha-2b subunit mRNA
2354	**M92287_at**	CCND3 Cyclin D3
2363	M93056_at	LEUKOCYTE ELASTASE INHIBITOR
2402	M96326_rna1_at	Azurocidin gene
2642	**U05259_rna1_at**	MB-1 gene
3252	**U46499_at**	GLUTATHIONE S-TRANSFERASE, MICROSOMAL
4107	**X07743_at**	PLECKSTRIN
4196	**X17042_at**	PRG1 Proteoglycan 1, secretory granule
4328	**X59417_at**	PROTEASOME IOTA CHAIN
4366	**X61587_at**	ARHG Ras homolog gene family, member G (rho G)
4377	**U46499_at**	GLUTATHIONE S-TRANSFERASE, MICROSOMAL
4847	**X95735_at**	Zyxin
5171	Z49194_at	OBF-1 mRNA for octamer binding factor 1
5501	**Z15115_at**	TOP2B Topoisomerase (DNA) II beta (180kD)
6041	**L09209_s_at**	APLP2 Amyloid beta (A4) precursor-like protein 2
6281	**M31211_s_at**	MYL1 Myosin light chain (alkali)
6855	**M31523_at**	TCF3 Transcription factor 3 (E2A immunoglobulin enhancer binding factors E12/E47)
1909	M29696_at	IL7R Interleukin 7 receptor
1953	M33195_at	Fc-epsilon-receptor gamma-chain mRNA
2335	M89957_at	IGB Immunoglobulin-associated beta (B29)

runs which have a complete frequency level (freq = 30) from Colon and Leukemia datasets successively. Furthermore, the selected genes are considered informative in most well-known methods in the literature. Specially, in Leukemia dataset which has been widely studied in this area.

As a result, genes selected from Leukemia dataset listed in boldface in Table 5, were also selected among the top 25 most relevant genes by Wu et al. [4] and

they are considered useful to discriminate between the two class label AML and ALL. Moreover, eight out of these thirty top genes selected by our method, i.e., M23197_at, M27891_at, U05259_rna1_at, U46499_at, X95735_at, L09209_s_at, M31523_at and M89957_at were deemed as relevant by Zhu et al. [21]. The selected genes can now be validated by biologists through clinical trials. We expect these discoveries may offer useful information for biologists and medical experts.

5 Conclusion

In summary, we considered in this chapter the application of ensemble feature selection methods to biomarker identification. Indeed, the most reviewed ensemble feature selection methods attest that this technique is a promising direction for more stable and accurate selection in cancer gene identification. We have also proposed a massively parallel approach based on meta-ensemble of filters for biomarker discovery from high dimensional data. The MPME-FS is different from other ensemble feature selection methods since it performs a parallel selection in two steps: the first one within each ensemble by the aggregation of results of different selectors, the second step is the aggregation of the outcomes of all ensembles using a second consensus function. The final selected biomarkers are employed to construct a classification model that will be used as an effective tool to handle patients and diagnose cancer subclasses.

In addition, the proposed MPME-FS is very fast and is computationally efficient as it is massively parallel and no learning algorithm is used in the selection process. Instead, we have employed filter model which is usually exploited when the number of features becomes very large especially for high dimensional data. Clearly, the MPME-FS can be performed using any ranking based feature selection algorithm and applied to any feature selection problem.

The experiments over five DNA microarrays datasets revealed that good results can be achieved through MPME-FS in terms of classification performance and robustness. Biological analysis of the results shows that MPME-FS provides the selection of highly informative genes which have biological meanings and are also selected by the other approaches.

References

1. Zhang, X., et al.: Integrative omics technologies in cancer biomarker discovery. Omics Technol. Cancer Biomark. Discov. **129** (2011)
2. Nair, M., Sandhu, S.S., Sharma, A.K.: Prognostic and predictive biomarkers in cancer. Curr. Cancer Drug Targets (2014)
3. Mäbert, K., Cojoc, M., Peitzsch, C., Kurth, I., Souchelnytskyi, S., Dubrovska, A.: Cancer biomarker discovery: current status and future perspectives. Int. J. Radiat. Biol. (0), 1–48 (2014)

4. Wu, M.Y., Dai, D.Q., Shi, Y., Yan, H., Zhang, X.F.: Biomarker identification and cancer classification based on microarray data using laplace naive bayes model with mean shrinkage. IEEE/ACM Trans. Comput. Biol. Bioinf. (TCBB) **9**(6), 1649–1662 (2012)
5. Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A.: A review of feature selection methods on synthetic data. Knowl. Inf. Syst. **34**(3), 483–519 (2013)
6. Bolón-Canedo, V., Sánchez-Maroño, N., et al.: A review of microarray datasets and applied feature selection methods. Inf. Sci. **282**, 111–135 (2014)
7. He, Z., Yu, W.: Stable feature selection for biomarker discovery. Comput. Biol. Chem. **34**(4), 215–225 (2010)
8. Guan, D., Yuan, W., Lee, Y.K., Najeebullah, K., Rasel, M.K.: A review of ensemble learning based feature selection. IETE Tech. Rev. **31**(3), 190–198 (2014)
9. Upadhyaya, S.R.: Parallel approaches to machine learning—a comprehensive survey. J. Parallel Distrib. Comput. **73**(3), 284–292 (2013)
10. Yang, P., Hwa Yang, Y., B Zhou, B., Y Zomaya, A.: A review of ensemble methods in bioinformatics. Curr. Bioinf. **5**(4), 296–308 (2010)
11. Awada, W., Khoshgoftaar, T.M., et al.: A review of the stability of feature selection techniques for bioinformatics data. In: Information Reuse and Integration (IRI), 13th International Conference, 356–363 (2012)
12. Saeys, Y., Abeel, T., Van de Peer, Y.: Robust feature selection using ensemble feature selection techniques. In: Machine Learning and Knowledge Discovery in Databases, pp. 313–325. Springer, Berlin (2008)
13. Abeel, T., Helleputte, T., et al.: Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. Bioinformatics **26**(3), 392–398 (2010)
14. Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A.: Data classification using an ensemble of filters. Neurocomputing **135**, 13–20 (2014)
15. Yang, P., Liu, W., Zhou, B. B., Chawla, S., Zomaya, A.Y.: Ensemble-based wrapper methods for feature selection and class imbalance learning. In: Advances in Knowledge Discovery and Data Mining, pp. 544–555. Springer, Berlin (2013)
16. Xu, J., Sun, L., Gao, Y., Xu, T.: An ensemble feature selection technique for cancer recognition. Bio-Med. Mater. Eng. **24**(1), 1001–1008 (2014)
17. Ghorai, S., et al.: Cancer classification from gene expression data by NPPC ensemble. IEEE/ACM Trans. Comput. Biol. Bioinf. **8**(3), 659–671 (2011)
18. Boucheham, A., Batouche, M.: Robust biomarker discovery for cancer diagnosis based on meta-ensemble feature selection. In: The Proceedings of Science and Information Conference, IEEE, pp. 452–460 (2014). ISBN: 978-0-9893193-1-7
19. Boulesteix, A.L., Slawski, M.: Stability and aggregation of ranked gene lists. Briefings Bioinf. **10**(5), 556–568 (2009)
20. Haury, A.C., Gestraud, P., Vert, J.P.: The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. PLoS ONE **6**(12), e28210 (2011)
21. Zhu, Z., Ong, Y.S., et al.: Identification of full and partial class relevant genes. Comput. Biol. Bioinf. IEEE/ACM Trans. **7**(2), 263–277 (2010)

Relationship Discovery and Navigation in Big Graphs

Ján Mojžiš and Michal Laclavík

Abstract Relationship discovery is a challenging field, especially when handling with big graphs (tenths to hundreds vertices and edges). In this paper, we define a set of rules for relationship discovery. To evaluate them and find connections we implement these rules in Pregel Relationship Discovery (PRD) algorithm and also in our Graph Clutter Removal Tool (AGECRT). Our PRD algorithm is capable of navigation even through the opposite direction of edge, even without additional indexation. Graph visualization is important field in data overview creation. For visualization, we also propose a new edge coloring method based on hash codes obtained from vertices. When edges cross, not color but also a pen style can help to navigate edges. In this paper we join relationship discovery (represented by PRD) and graph visualization (with AGECRT). In order evaluate our rules and work we use PRD and AGECRT to present an experimental results from Freebase dataset.

1 Introduction

Many relations, which can be modeled by graphs are suitable for visualization. Social networks (e.g. co-citation graphs), traffic, internet connections and etc. In general, graphs are visualized in order to get kind of graph "overview", to easily see, recognize vertices and edges and to see relations between vertices.

But graph visualization can often be problematic for large and complex graphs [1, 2]. Visualization of such graph can introduce significant information overhead or visual clutter, simply from displaying too much data on too small area of display [3] or vertices or edges overlapping and edges crossing [4, 5], which, aside from the clutter itself, can also lead to confusion or misunderstanding. Therefore

J. Mojžiš (✉) · M. Laclavík
Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia
e-mail: upsyjamo@savba.sk

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_7

avoiding clutter and to increase the people's understanding graphs is a challenge. Visual overhead or visual clutter is more defined by Kerren [6] and also Jusufi [5], who also talks about readability of the graph.

In order to address such problems and to provide a comfortable graph visualization for user, various methods were proposed. While many of them share common properties, like the goal to visualize graph and reduce clutter, however there is not a rule, that every method can be used in every case (generalization). Therefore it is important to understand which method should be used on particular graph, regarding desired graph visualization.

We think that theme of graph coloring is not yet completely exhausted. Even despite the fact, that On modern displays we can display as much as 2^{16} or even 2^{32} colors, we are unable to visually identify and recognize all of them. It is due to human visual system limitations [7, 8]. But still, assigning different colors to vertices of different types can help to navigate and visually perceive graph relations. For example the Author type can have blue color and Publication can be colored to orange. Vertex coloring is particularly useful in large graphs, where there are many vertices and is important to recognize vertice types. In addition, in case, when edge crossing can not be completely eliminated, edge coloring is very helpful in order to navigate and follow particular edge between vertices even when edges cross.

But how colors of edges or vertices are calculated? Are they defined by the user, or they can be obtained automatically. We would like to provide a new concept of edge coloring. Instead of bounding edge colors with edges and storing edge colors inside a graph, edge colors should be calculated automatically on-the-fly and should be dependent only on connected vertices (their values). This could allow automatic color change, upon the change of vertex value. The color of edge is bounded into vertices, which it connects. On the other side, when graph layout is changed, graph is redrawn, the color of edge would remain the same every time, since the value of vertex remain intact. In this paper we would like to provide new, comfortable and easy to implement method of edge coloring. The key element in this concept is (common) hash function, able to obtain numerical (integral) value of vertex (of its name or any other vertex value).

1.1 Distributed Graph Computing

When we want to process a large graph, which contains millions vertices and edges, or simply a graph, which is so big, that it could not be loaded into main memory of a single machine, distributed computing should be considered. The computing power and memory resources, which can not be provided by a single machine, are obtained by joining machines together. Here we would like to compare two well known paradigms of distributed computing, namely MapReduce and Pregel.

MapReduce is a framework for processing parallelizable problems across huge datasets using a large number of computers (nodes), collectively referred to as a

cluster (nodes inside network) or grid (nodes outside network). MapReduce can take advantage of locality of data, processing it on or near the storage assets in order to reduce the distance over which it must be transmitted. It can be applied to significantly larger datasets than "commodity" servers can handle a large server farm can use MapReduce to sort a petabyte of data in only a few hours [9]. MapReduce achieves reliability by parceling out a number of operations on the set of data to each node in the network. Each node is expected to report back periodically with completed work and status updates.

Although MapReduce is still rather popular framework for parallel processing, Pregel was shown to be more suitable for iterative graph computation [10, 11] and, according to [12], MapReduce can lead to suboptimal performance. Performance issues regarding parallel graph processing are well discussed in [13]. Pregel computing model tries to address those issues and reduces the communication overhead (as can appear in MapReduce) by using computations consisting of a sequence of iterations, which are called supersteps. This concept is more suitable for iterative graph computation and, also for our relationship discovery goal, due to its vertex-centric paradigm. Also, an important advantage of Pregel is, that it does not transmit large amount of data (temporal results) during calculations, only small fractions—messages are transmitted. For further reading about Pregel, we refer to [12]. Despite the promising look of Pregel, existing work on Pregel algorithms [11] is rather ad hoc, which looks more like a demonstration of how Pregel can be used to solve a number of graph problems, and lacks any analysis on the cost complexity. We would like to adapt on Pregel and, in this paper, propose a new graph relationship discovery method. First, our intention is to be able to navigate even through the opposite edge direction. We would like to keep the graph structure intact, unchanged. We hold the idea, that in social network, the relationship between vertices is present regardless of edge direction. And to be able to discover such relationship, edge direction should not present a problem. Thus, our solution will work for directed as well as undirected graphs. Here the vertex-centric paradigm of Pregel is well suitable for our needs. For our next contribution, we define a set of rules, we use to find the relationship between vertices. This concept is rather new and we have found no use of it anywhere. One interesting property of it is, that it can be used to create closed paths in graph (it would create no degree 1 vertice). Thus, a relation to other vertices is preferred.

The rest structure of this paper is as follows: In the Related work, we discuss actual state-of-art solutions of graph coloring and parallel distributed computing. Next section is definition of our rules of relationship discovery and, in this section, we provide a new method of edge coloring as well along with the definition of our algorithm for relationship discovery. We join graph clutter removal and relationship discovery with the same rules. Relationship discovery implemented in PRD is the same as in AGECRT for graph clutter removal. The experiment section discuss about initial setting of environment, obtaining dataset, converting it in order to be usable. The section of results contains the results we obtained from our Pregel implementation of our rules. In order to see resulting relations, we use our Advanced Graph Clutter Removal Tool program (AGECRT). We provide charts

of progress during relations of vertices were calculated in Pregel implementation. Conclusion summarizes the results, discuss related solutions and offers new, possible heading for improvement in stated solutions.

2 Related Work

2.1 Graph Visualization

2.1.1 Coloring

From the actual well-known graph visualization and clutter removal methods, we can find graph coloring as the solution to identify edges or vertices [14, 15]. It is useful, when edge crossings cannot be completely eliminated, allowing still to identify and visually follow particular edge. From the psychological aspect, gestalt laws can be used to alleviate edges crossing problem in graph drawing [2]. The principles are based on the human visual perception and generally about our capability to achieve and keep informations in noisy environments. Herman [4] also see the edge crossings as the problem in his aesthetic criteria. But human visual system is not equally sensitive to all colors or level of contrast [7, 8]. Color graph visualizers are thus limited in the color palette, probably giving better results when combined other visualization methods, such as graph clustering [14, 16, 17], partitioning [18] or combination of both [19].

2.1.2 Clustering

We can refer to clustering as the method, which uses structural information about the graph and generally group edges or vertices into separate clusters based on their properties. In visualization terms, clustering can be used to accomplish filtering functions [4]. But Itoh et al. [16] also see the drawbacks in number of clustered categories and in number of distinguishable colors. As a solution he suggests multidimensional visualization techniques, which, in advance can be found in [20–22].

2.1.3 Edge Bundling

Another choice is the edge bundling method, which reduces the amount of edges [23, 24]. Although edge bundling technique slightly lowers correlations, it can also significantly reduce edge crossings. But the limits of edge bundling can be incompatibility of edges or complexity in case of graphs with a large number of edges. Some specific edge bundling methods would require control meshes or hierarchy, thus their usage is limited to tree-like hierarchical structures. Burch, who combine

edge bundling with lens [25] use their Rapidly-Exploring Random Trees (RRTs) to bundle edges. It reduces the visual clutter, the general graph structure is not changed. But the graph structure is not taken into account and the view itself can distort the interconnections and lead to misinterpretations. Although he states, that their RRT method is independent from graph layout, Burch also argues, that their RRT works better for circular graph layout.

2.1.4 Lens

Lenses are the next visualization method. With the use of lens, a particular region is selected and then magnified (so-called focus + context approach), user can see vertices and edges in close detail, but still preserving the original structure of the graph intact and unchanged. Lens graph view methods are thus often independent from the graph layout. Fisheye views, as an application of well-known fisheye distortion technique, are used often [26–29], followed by Magic Lens [30] or Network Lens [5]. We can find combination of lens and edge bundling methods in [25]. Although lens are useful in visual clutter reduction, there are drawbacks like distortion, and misinterpretations, caused by the fact, that graph structure is not taken into account [25].

2.1.5 Graph Layout

In contrast to lens, which arrange vertices and edges temporarily, graph layout algorithm arranges vertices and edges permanently. In order to map the relational data into a geometrical plane (while drawing), the graph layout algorithms are used. Various approaches on layouts can be found in [18]. The graph drawing algorithms can be found in [18, 31, 32]. Herman [4] suggests Reingold and Tildford layout algorithm [32], because he see it as the good example of achieving his aesthetic goals. Graph layout algorithms are also the subject in [31]. We can recommend some graph visualizer implementations, such as JGraph or Jung. Users can benefit from their open-source availability and the fact, that they often tends, aside of visualization, to implement graph computing algorithms. For example Jung implements Dijkstra shortest path algorithm

Our method, based on rules is similar to Chen and Hsieh [33] although it is not designed strictly for cliques, does not use thresholding (originally) and kernels.

2.2 Distributed Graph Computing

From the Pregel using solutions we can mention Bao and Suzumura [34]. They created X-Pregel, a graph processing system based on Google's Computing Pregel model, by using the state-of-the-art PGAS programming language X10.

They introduce two new features that do not exists in the original model to optimize the performance: (1) an optimization to reduce the number of messages which is exchanged among workers, (2) a dynamic re-partitioning scheme that effectively reassign vertices to different workers during the computation. The result is 2-times speed up on PageRank. Their system is also faster in comparison to Graph Processing System (GPS) or Graph. But they state, despite the 10× improvement on network IO just results only 1.5× runtime improvement, because there is only one worker at a time to process the re-partitioning then the decrease in network IO take places slowly.

The Generalized Iterative Matrix-Vector multiplication (GIM-V) [20] is known for its fast processing on large scale graph, but the model is difficult to program and is not flexible in comparison to Pregel.

Laclavík et al. [35] use MapReduce in order to find relationship in Enron Social Network of email archive. Their gSemSearch algorithm is designed to skip nodes with higher degree than other nodes in the stack. However, this approach can skip some relevant, higher number of neighbors containing nodes. As a graph database storage backend they use Simple Graph Database (SGDB). The key aspect here is, that due to use of SGDB [36], their solution, as is presented in their paper, is not scalable. They are unable to follow opposite direction of edge. Enron graph corpus, which they use to evaluate their solution, contains 8.3 millions vertices and 20 million edges, while we focus our solution to scale from hundred of millions vertices and above. Finally, their solution is unable to navigate through opposite direction of the edge.

There are solutions designed to store graph-like data structures and to perform queries above such data [36, 37], Neo4j, OWLIM, Shard, Jena or Sesame. But not all of them are designed for the use on distributed platform. We assume, that it is exactly distributed computing model, which can scale and use vertex-centric model (like Pregel). Messaging can be used as graph traversal and here, even navigate through the opposite direction of an edge, as we show in this paper.

In this paper we would like to join two fields of Information visualization (with AGECRT and custom edge coloring) and parallel distributed graph processing (our PRD algorithm) into one solution. As we find important not only find relation, but also present results to the end user. We thus find graph visualization suitable to create overview of the data.

3 Vertex Relationship

For the relationship discovery we propose our solution based on closed paths. Basically, there are 2 types of vertices; *interesting* and *normal*. When a vertex is considered for the relationship, it is approved in the case, if, at least one more of its neighbors is already assigned in the set of approved vertices.

For the formal definition, let there be undirected graph, which is given by an ordered pair $G = \{V, E\}$. From all neighbours of given vertex v, count only those

neighbours, which are connected to at least 1 interesting vertex (other than v). Vertices, in our solution, can be either interesting or normal. Let $S \subseteq V$ be the subset of all interesting vertices and $I = V - S$ set of all normal vertices. Let there be function to get all neighbours of vertex x such as $N(x) = \{u | \{x, u\} \in E\}$

$$H = \{u | \exists v_1, v_2, \in S : u \in I \wedge u \in N(v_1) \wedge u \in N(v_2)\} \tag{1}$$

$$h \in H : M = \{u \neq h | u \in N(h) \wedge u \in S\} \tag{2}$$

where:

H set of interconnected neighbours
S set of all interesting vertices
I set of all normal vertices
M set of interesting neighbours of h

4 Algorithm

Based on the rules, we define in previous section, we have designed custom Pregel Relationship Discovery (PRD) algorithm for Pregel computing model. PRD is capable of navigation through the opposite direction of an edge.

Vertices in PRD can be of 2 types; interesting and normal (represented by their property). Interesting property is set before PRD starts and denotes the particular set, in which we want to search for relationships.

The key element in our traversal PRD is path length. Vertices are sending messages while they are active, which means, they either receive message, or return true from compute() method. In the first superstep, vertices send either PING or INTERESTING messages to their neighbors. Only vertices marked as interesting send INTERESTING messages. In subsequent supersteps, vertices spread those messages further until desired path length is reached. Upon each next step, when a message is received, its path length is incremented by 1 in order to keep distance monitored. In our experiment, we have set the value of path length to 2. Shorter paths are excluded and if message received has value of 2, it is not spread further.

For the messaging, there are conditions; do not send message back to original sender while spreading and the length of continuous spreading is limited by the path length. We also control maximum count of messages, generated by vertices, with variable maxMessagesPerStep, which is user defined. We experimentally verified, that for our case and hardware environment condition, the maximum is 18. Note, that this limit affects only vertices and not workers. So there can be more than 18 messages generated in one worker (2 vertices ūp to 18×2 messages). In PRD, all variables are either stored directly inside a given vertex (whether vertex is interesting, …) or they are stored in generated message (type of message, …). A new relationship is found, when normal vertex received at least two messages (at sum) from at least two (different) interesting vertices. The principle of PRD is

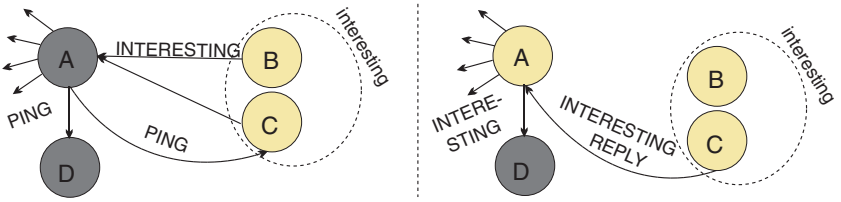


Fig. 1 Concept of message passing and spreading in PRD. *Left* step 1, both interesting vertices *B* and *C* are sending INTERESTING message to *A*. Vertex *A* sending PING to *C* and to all of its neighbors it has outgoing edge (no outgoing edge to *B*). *Right A* spreading INTERESTING message and *C* sends reply message INTERESTING_REPLY

depicted in Fig. 1. Because the source code of PRD is rather long, we have created a separate document for it, which is available at http://jamo.wz.cz/PRD/source.txt.

5 Edge Coloring

In this section we present a new edge coloring technique, composed entirely on the values of connected vertices, which lie of the edge. We propose to use hash function $h(x)$, where x is the value of vertex, to obtain numeric value of the fixed length and use it in order to refine the color of the edge. Hash function is used to obtain numeric value from of vertex value. Two hashes from two (neighboring) vertices are summed and then potentially high brightness is reduced. Contrast is then calculated and, possibly enhanced. Again, brightness is measured or even reduced. Finally we obtain color value of edge. The source code of our color calculation from hash code can be found in document online http://jamo.wz.cz/coloring/hashcolor.txt.

During our experiments we have discovered, that pen style (dash, dot, not solid) in combination with coloring can considerably affect visual perception, especially when edges cross. This can be seen in Figs. 2 and 3.

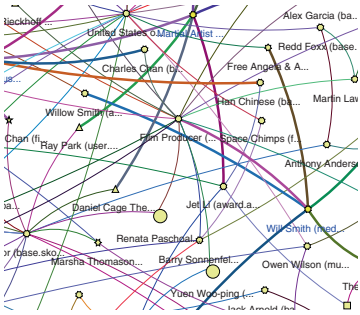


Fig. 2 Graph with *solid* edges only

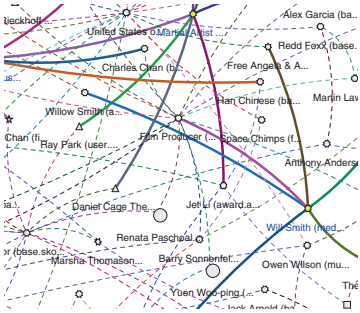


Fig. 3 Graph with *dashed* and *solid* edges

6 AGECRT

Advanced Graph Clutter Removal Tool (AGECRT) is our tool, which we use to visualize all graphs found in this paper. It implements our rules of relationship discovery (section Vertex relationship) and is capable to reduce visual clutter in graph visualizations as shown in Fig. 4.

7 Experiment

7.1 Environment

We used 5 worker machines and 1 master machine, each equipped with 21 CPU Intel Xeon, 2 GHz, 32 GB RAM and OS Ubuntu 12.04.5 LTS. PRD Was programmed on the basis of Sedge, a basic Pregel implementation in Java. PRD is our custom vertex compute() method is Sedge and is described in previous section. Sedge is designed to work with graphs, whose vertices are numerically identified. Freebase identifiers are URI-based. Thus conversions were needed. With the help of Jung graph library, which is an extensible graph storage and visualization backend, we have created and used AGECRT program in order to visualize results. Here again, conversion was needed, because AGECRT accept graphs in different format (nor Freebase RDF, nor Sedge). First major conversions was to convert from freebase URI-based mid identifiers into integer-based numerical identifiers

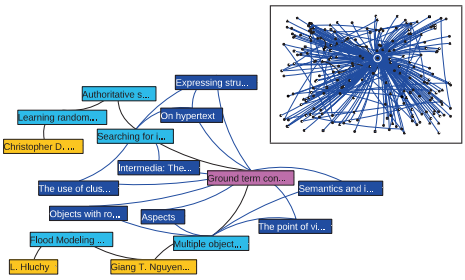


Fig. 4 Graph visual clutter reduction based on our rules from section Vertex relationship. Without rules, graph is cluttered (in frame)

compatible with Sedge. During this conversion we have excluded all object literal nodes and only object nodes matching the pattern

"<http://rdf\\.freebase\\.com\\/ns\\/(\\w\\..*?)>"

were included, all other object nodes were filtered out. After calculation in Sedge was finished, next major conversion was to convert Sedge output graph into AGECRT compatible format. In order to convert values in three major conversion successfully, minor conversions, consisting mainly of mapping key-value pairs, had to be launched as well. Freebase URI identifiers were mapped to Sedge numerical identifiers. In the last phase, where Sedge graph was converted to AGECRT format, Freebase URI-based identification of vertices was not sufficient. In order to gain particular knowledge of semantic representation of vertices visualized in AGECRT, and to be able to run conversions to and from for many times during testing process, we have created 2 key-value maps. For Freebase URI subject and object MIDs we have obtained additional textual identifications (mainly based on predicate value/type/object/name), extracted from Freebase. We also created mapping to get type information, Freebase subject and object URI identifiers were also mapped to their types (mainly/type/object/type). Important vertices in PRD were all entries found in Table 1. All other vertices were set as normal.

Originally, Freebase contained 113×10^6 subjects and 2.7×10^8 edges. After conversion into Sedge format, with excluded literal object nodes there was 36×10^6 subjects and 404×10^6 edges.

7.2 Results

Our experiment in Pregel started on 2014-Aug-21 at 15:05:55 and finished 2014-Aug-31 01:27:00 (lasted 10 days). There were 207,382 supersteps. The progress of message generation and steps is shown on Fig. 5. After it finished, we

Table 1 Two sets of entries used for AGECRT shortest path (many-to-many) and PRD (important vertices)

Freebase MID	Name	Freebase MID	Name
m.02xbw2	Gabrielle Union	m.029_l	Delroy Lindo
m.0147dk	Will Smith	m.012d40	Jackie Chan
m.0271y9f	Jaden Smith	m.05v_r84	Jackie Chan
m.01qg7c	Barry Sonnenfeld	m.01q_ph	Owen Wilson
m.01vvzb1	DMX	m.01xndd	J.J. Abrams
m.0hqly	Steven Seagal	m.042xrr	Anthony Anderson
m.0gy64rt	Samuel Steven Seagal	m.0bvb9mz	Anthony Anderson
m.02633 g	Martin Lawrence	m.0451j	Jet Li
m.01th95y	Martin Lawrence	m.05qg6 g	Zoe Saldana
m.01hhx1l	Willennium	m.05jpsx	Chi McBride
m.0b8xmc	Robinne Lee	m.029pnn	Tom Arnold
m.07y925	Marsha Thomason	m.05d79 k	Bill Duke

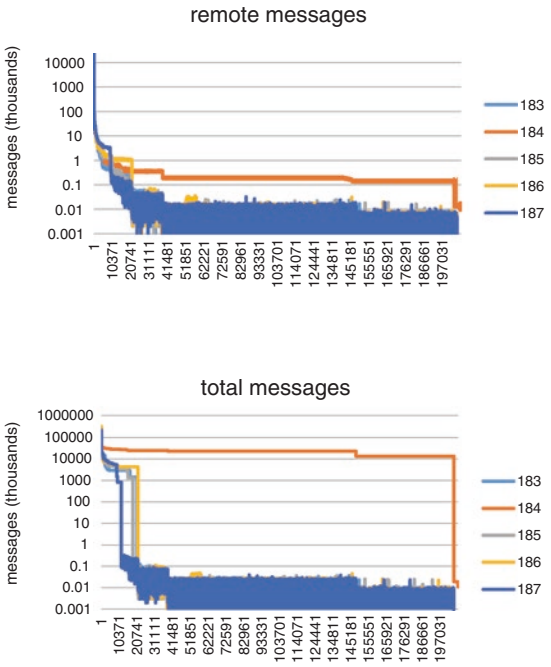


Fig. 5 Remote and total messages generated during our experiment with PRD. Charts are rotated by 90 counter clockwise. *X axis* supersteps, *Y axis* message generated (log base 10 scale)

obtained 314,668 activated vertices, from which, each was activated at least 2 times. Next step was to convert Sedge formatted output into AGECRT graph format in order to visualize resulting graph. After conversion, resulting graph was loaded into AGECRT. We have excluded some vertices from this graph, namely Gender, Male, Female, Place of Birth, Place of Death, because we were not interested in common properties. Graph contained 234,786 vertices and 262,279 edges. Because resulting graph was still very big to visualize, in AGECRT we have launched our altered Dijkstra shortest path (called many-to-many) to calculate shortest paths between 2 sets of vertices, which are found in Table 1. From the table we can see, that some entities, although clearly of the same name, have different MIDs (identifiers) in Freebase. This is a bit surprising for us and exact understanding probably requires deeper study of Freebase structure. For now, it is clear, that films referring to their actors not directly (film $\Rightarrow$ actor), but there is some type of extra vertex with additional informations, like character name and etc. (film $\Rightarrow$ extra vertex $\Rightarrow$ actor). Thus we had to search up to path length of 2, as those extra vertices are unique and each of them connects only one actor and one film.

After PRD's output was converted back to Freebase MIDs based identifiers and we obtained name and type informations, we were able to convert results again into AGECRT graph format. After this, we were able to read semantic information and results surprised us a bit. Instead of expected relationship of shared films between Jackie Chan and Owen Wilson, based on our settings, we have found

no connection between them. Also, Will Smith and Karate Kid are connected, but not so Jackie Chan. Instead, Jet Li is connected to Karate Kid through Chinese Martial arts. The reason is, that Will Smith activated Karate Kid not directly, but through several vertices (e.g. PG (USA), MID m.0kprc8), so Karate Kid was activated from Will Smith. Jet Li activated Chinese martial arts. Both Karate Kid and Jet Li have connection to Chinese Martial arts. But Jackie Chan is connected through People's Choice Award (in the Freebase original RDF graph), which has MID m.0dlskb3 and is shared only with Jackie Chan. Thus this vertex could not be activated for 2 times, as we have designed in our rules and thus was not included. This was the original reason we have set path length to 2. Perhaps, path length value could be further revised.

Further look at Fig. 6 could bring another question. Anthony Anderson, Tom Arnold, and Marsha Thomason are all connected to Actor. But why not so Will Smith or Jackie Chan? Aren't they actors? The reason is that Marsha Thomason connects directly to Actor along with Will Smith. But Marsha Thomason did not activated Actor directly. Instead Marsha activated Scott Taylor (MID m.025zlv2) and he passed message to Actor. In Actor, path has been saved, *source* is Marsha Thomason and *through* is Scott Taylor. Here we have to say, that paths, where *through* was equal to activated vertex, were excluded from results. We did this because we were interested in connections, where *through* was not same as activated vertex. Among other interesting connections, there is connection of Steven Seagal, Jackie Chan and Jet Li through Martial Artist. The MID identifiers of names mentored above are all found in Table 1. In order to extract type information (Actor/Martial artist, ...) from the record in Freebase, our parser returned the first type it found, despite there could be more than one type assigned to one

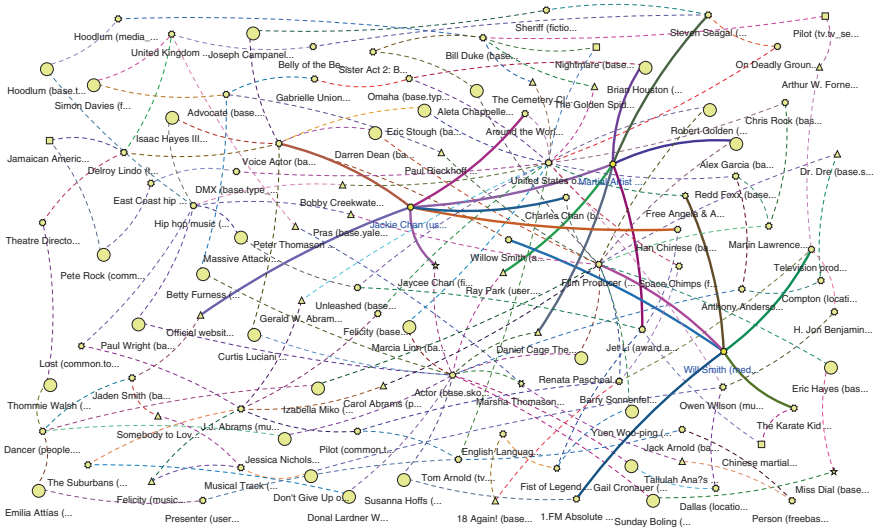


Fig. 6 Visualized resulting graph as displayed in AGECRT. Selected vertices are Jackie Actor, Martial Artist and Will Smith

Table 2 Freebase worker statistics

Worker	Subjects Cnt.	Max. Ngh. Cnt.	Tot. Ngh. Cnt.	Avg. Ngh. Cnt.
w1	21×10^6	375×10^3	81×10^6	3.80
w2	19×10^6	3.6×10^6	80×10^6	4.21
w3	21×10^6	331×10^3	80×10^6	3.84
w4	21×10^6	382×10^3	80×10^6	3.80
w5	20×10^6	210×10^3	80×10^6	3.82

Max Maximum, *Ngh* Neighbors, *Cnt* Count found on worker, similarly *Tot* Total and *Avg* Average

record. Thus extracted type information could be inaccurate. The parser can be further improved. Table 2 presents allocation of the data between workers as the result of graph split.

8 Conclusion

In this paper we have designed our rules for relationship discovery. To test and verify these rules, we have further implemented them in our Graph Clutter Removal Tool AGECRT. In graph visualization we have used them to reduce visual clutter (Fig. 4). We propose an edge coloring based on hash codes of vertices. The color information is not stored, but dynamically calculated each time edge is drawn. Edge color remains the same each time a graph is drawn, while values of two connected vertices (their hash codes) remain the same. We recommend to use edge coloring along with different pen styles, because we can identify and follow edges more easily (Figs. 2, 3 and 6).

With our PRD, we have found relationship between set of vertices Table 1. Graph of these relations was loaded into AGECRT and then Dijkstra's shortest path was calculated (again for sets from Table 1). The final result was visualized as shown in Fig. 6. Some relations were obvious, like Steven Seagan and Jet Li are connected through Martial Artist. But some of them were not found (Jackie Chan and The Karate Kid). The reason is PRD and its path length property of 2, shorter paths were eliminated. Also, final activated vertex must not have been the same as *through* vertex.

But we have proven that our PRD is capable of searching for relations (Will Smith and Jet Li). PRD is able to discover relationships regardless of edge's direction (in comparison with Laclavík et al. [35]), which we find important in RDF graphs, which are directed by nature. We see relation as an undirected edge, as it is not important, whether Bob knows Alice or it is Alice, which knows Bob. Also this advantage means, that we do not need additional index (to store backward edges). Because PRD is implemented in Sedge's Pregel implementation, which is scalable, our solution is also scalable. The disadvantage of long lasting (10 days), while results are obtained, can be reduced by adding more worker

machines. By using more machines, upper limit on messages count (in paper 18) can be set to higher values, resulting in faster completion.

For the future, we plan to improve PRD algorithm to address issues stated in this paper and create an architecture, where we join PRD, AGECRT and conversion engines, in the background, into one solution. A web page is also considered as a front-end.

Acknowledgment This work was supported by the Slovak Research and Development Agency project name CLAN, number APVV-0809-11 and by the Scientific Grant Agency of the Ministry of Education, science, research and sport of the Slovak Republic and the Slovak Academy of Sciences, project VEGA, number 2/0185/13.

References

1. Melo, C., Le-Grand, B., Aufaure, M., Bezerianos, A.: Extracting and visualising tree-like structures from concept lattices. In: 15th International Conference on Information Visualisation (IV), pp. 261–266. IEEE, Piscataway (2011)
2. Rusu, A., Fabian, A. J., Jianu, R.: Using the gestalt principle of closure to alleviate the edge crossing problem in graph drawings. In: 15th International Conference on Information Visualisation (IV), pp. 488–493. IEEE, Piscataway (2011)
3. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychol. Rev. **63**(2), 81 (1956)
4. Herman, I., Melanon, G., Marshall, M.S.: Graph visualization and navigation in information visualization: a survey. IEEE Trans. Visual. Comput. Graph. **6**(1), 24–43 (2000)
5. Jusufi, I., Dingjie, Y., Kerren, A.: The network lens: Interactive exploration of multivariate networks using visual filtering. In: 14th International Conference on Information Visualisation (IV), pp. 35–42. IEEE, Piscataway (2010)
6. Kerren, A., Ebert, A., Meyer, J.: Human-Centered Visualization Environments. Springer, Berlin (2006)
7. Mullen, K.T.: The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings. J. Physiol. **359**(1), 381–400 (1985)
8. Kelly, D.H.: Visual contrast sensitivity. J. Mod. Opt. **24**(2), 107–129 (1977)
9. Czajkowski, G., Dvorský, M., Zhao, J., Conley, M.: Sorting Petabytes with MapReduce—The Next Episode. http://googleresearch.blogspot.sk/2011/09/sorting-petabytes-with-mapreduce-next.html (2014). Google Retrieved 12 Sept 2014
10. Dayarathna, M., Suzumura, T.: A first view of exedra: a domain-specific language for large graph analytics workflows. In: WWW (Companion Volume), pp. 509–516, (2013)
11. Quick, L., Wilkinson, P., Hardcastle, D.: Using Pregel-like large scale graph processing frameworks for social network analysis. In: ASONAM, pp. 457–463, (2012)
12. Malewicz, G. et al.: Pregel: a system for large-scale graph processing. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. ACM, New York City (2010)
13. Lumsdaine, A., et al.: Challenges in parallel graph processing. Parallel Process. Lett. **17**(1), 5–20 (2007)
14. Omote, H., Sugiyama, K.: Method for drawing intersecting clustered graphs and its application to web ontology language. In: Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation, vol. 60, pp. 89–92. Australian Computer Society Inc, Australia (2006)
15. Jianu, R., Rusu, A., Fabian, A.J., Laidlaw, D.H.: A coloring solution to the edge crossing problem. In: 13th International Conference on Information Visualisation, pp. 691–696. IEEE, Piscataway (2009)

16. Itoh, T., Muelder, C., Ma, K.L., Sese, J.: A hybrid space-filling and force-directed layout method for visualizing multiple-category graphs. In: Visualization Symposium, 2009. PacificVis' 09. IEEE Pacific, pp. 121–128. IEEE, Piscataway (2009)
17. Dudas, P.M., Jongh, M.D., Brusilovsky, P.: A semi-supervised approach to visualizing and manipulating overlapping communities. In: 17th International Conference on Information Visualisation (IV), pp. 180–185. IEEE, Piscataway (2013)
18. Nguyen, Q.V., Huang, M.L.: EncCon: an approach to constructing interactive visualization of large hierarchical data. Inf. Visual. **4**(1), 1–21 (2005)
19. Misue, K., Zhou, Q.: Drawing semi-bipartite graphs in anchor + matrix style. In: 15th International Conference on Information Visualisation (IV), pp. 26–31. IEEE, Piscataway (2011)
20. Tejada, E., Minghim, R., Nonato, L.G.: On improved projection techniques to support visual exploration of multi-dimensional data sets. Inf. Visual. **2**(4), 218–231 (2003)
21. Cvek, U., Trutschl, M., Kilgore, P.C., Stone, R., Clifford, J.L.: Multidimensional visualization techniques for microarray data. In: 15th International Conference on Information Visualisation (IV), pp. 241–246. IEEE, Piscataway (2011)
22. Chalmers, M.: A linear iteration time layout algorithm for visualising high-dimensional data. In: Proceedings of Visualization'96, pp. 127–131. IEEE, Piscataway (1996)
23. Kienreich, W., Seifert, C.: An application of edge bundling techniques to the visualization of media analysis results. In: 14th International Conference on Information Visualisation (IV), pp. 375–380. IEEE, Piscataway (2010)
24. Holten, D., Van Wijk, J.J.: Force directed edge bundling for graph visualization. Computer Graphics Forum, vol. 28(3), pp. 983–990. Blackwell Publishing Ltd, Hoboken (2009)
25. Burch, M., Schmauder, H., Weiskopf, D.: Edge bundling by rapidly-exploring random trees. In: 17th International Conference on Information Visualisation (IV), pp. 28–35. IEEE, Piscataway (2013)
26. Furnas, G.W.: Generalized Fisheye Views, vol. 17, No. 4, pp. 16–23. ACM, New York City (1986)
27. Furnas, G.W.: A fisheye follow-up: further reflections on focus + context. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 999–1008. ACM, New York City (2006)
28. Tominski, C., Abello, J., van Ham, F., Schumann, H.: Fisheye tree views and lenses for graph visualization. In: 10th International Conference on Information Visualization (IV), pp. 17–24. IEEE, Piscataway (2006)
29. Sarkar, M., Brown, M.H.: Graphical fisheye views. Commun. ACM **37**(12), 73–83 (1994)
30. Bier, E.A., Stone, M.C., Pier, K., Buxton, W., DeRose, T.D.: Tool glass and magic lenses: the see-through interface. In: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, pp. 73–80. ACM, New York City (1993)
31. Battista, G.D., Eades, P., Tamassia, R., Tollis, I.G.: Algorithms for drawing graphs: an annotated bibliography. Comput. Geom. **4**(5), 235–282 (1994)
32. Reingold, E.M., Tilford, J.S.: Tidier drawings of trees. IEEE Trans. Softw. Eng. **2**, 223–228 (1981)
33. Chen, T.T., Hsieh, L.C.: The visualization of relatedness. In: 12th International Conference on Information Visualisation IV'08, pp. 415–420. IEEE, Piscataway (2008)
34. Bao, N.T., Suzumura, T.: Towards highly scalable pregel-based graph processing platform with x10. In: Proceedings of the 22nd International Conference on World Wide Web Companion, pp. 501–508. International World Wide Web Conferences Steering Committee (2013)
35. Laclavík, M., Dlugolinsk, Š., Šeleng, M., Ciglan, M., Hluchý, L.: Emails as graph: relation discovery in email archive. In: Proceedings of the 21st International Conference Companion on World Wide Web, pp. 841–846. ACM, New York City (2012)
36. Ciglan, M., Nø rvåg, K.: Sgdb—simple graph database optimized for activation spreading computation. In: Database Systems for Advanced Applications, pp. 45–56. Springer, Berlin (2010)
37. Kyrola, A., Blelloch, G.E., Guestrin, C.: GraphChi: large-scale graph computation on just a PC. OSDI, vol. 12, pp. 31–46. (2012)

A Fuzzy System for Three-Factor, Non-textual Authentication

James Stockdale, Alex Vakaloudis, Juan Manuel Escaño, Jian Liang and Brian Cahill

Abstract As text-based authentication has had its critiques, non-textual techniques have been suggested throughout the last two decades. However, it is only lately, with the wide-spread adoption of smartphones and tablet devices that they have found a compelling application. Non-textual authentication may be faster and more secure and it also introduces a new paradigm for the authentication decision. We present a three factor system based on facial recognition, gesture and device ID and we define a fuzzy matching engine to handle authentication. Preliminary results indicate that such an approach can be fast and user-friendly.

Keywords Fuzzy matching · Authentication · Biometric recognition · Gesture recognition · Multi-factor

1 Introduction

Passwords are a familiar, perhaps ubiquitous feature of everyday modern life. The conventional authentication paradigm invokes a username to identify and a password to authenticate an individual user. Typically, these two elements of the

J. Stockdale · A. Vakaloudis (✉) · J.M. Escaño · J. Liang · B. Cahill
Nimbus Centre, Cork Institute of Technology, Bishopstown, Cork, Ireland
e-mail: alex.vakaloudis@cit.ie

J. Stockdale
e-mail: james.stockdale@cit.ie

J.M. Escaño
e-mail: juanmanuel.escano@cit.ie

J. Liang
e-mail: jian.liang@cit.ie

B. Cahill
e-mail: brian.cahill@cit.ie

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_8

authentication process are in a textual form. Much has been written about what constitutes a secure password and increasingly, users are advised, if not required, to provide ever longer and more complex passwords in the interest of maintaining security.

With the modern proliferation of both hardware and software that are designed to be largely operated without the use of a keyboard, the entering of traditional, textual passwords has become something of a chore, an inconvenience and, perhaps, an anachronism. Non-textual methods of authentication have been suggested and, more recently, implemented. Well known examples include gesture recognition implemented on various smartphones and biometric systems such as Apple Inc. Touch ID and facial recognition as supported by Android based mobiles.

Non-textual authentication methods differ in a number of ways from the classical username-password approach. Key among these is that successful authentication follows not only from an exactly matching input, but from any one of the set of sufficiently matching inputs. While the textual password must exactly match the stored prototype, the non-textual input need only be sufficiently similar to the stored prototype since the exact match is exceedingly unlikely. The requirement for a proximity based match suggests that a fuzzy approach is appropriate. In this paper we describe a three-factor authentication system employing fuzzy matching to determine the degree of matching between non-textual elements of authentication data.

2 Related Work

Fuzzy logic [1] has been widely used in matching techniques [2]. Various approaches are employed, such as fuzzy transforms [3, 4], relative distance [5–7] and similarity measure [8, 9]. Typical applications of fuzzy matching are text and signature recognition, due to the ability of characters to convey the same information while taking on different graphical forms [10–14].

Since gestures cannot be repeated with precision, but can convey sufficient information to consider them *almost equal* to a stored prototype, fuzzy logic is a suitable technique to check for similarity. To recognize faces, an extraction of features can be performed using a biometric algorithm. Authentication based on a biometric factor is a widely used technique for mobile devices e.g. [15]. Fuzzy logic is also an established method for matching those features [16].

3 System Design and Development

3.1 System Description

The System developed implements a three-factor authentication service using biometric, gesture and device id as the three factors. In usage, a user is presented with a camera view of himself and required to click the screen to freeze the image. The

user then draws a simple image, the gesture, on top of the frozen image. Example gestures may be a smile, a hat, a moustache, spectacles, etc., or perhaps something more abstract. If the user is registering a new account, this procedure must be repeated a certain number of times so that the system can confirm that the biometrics and gestures are sufficiently similar. This is analogous to the *repeat password* prompt that is familiar in textual authentication systems. In addition, the system tests the new user's biometric for absence of similarity to all previously registered biometrics and returns an error is a similar biometric is found. This is analogous to a *user id already in use* message in traditional systems. If the user has already registered and is returning to login, the process is performed once and authentication (or not) is determined based on the captured biometric and gesture. In practice, each user account is also tied to a specific device. The device is determined during the first registration and subsequent logins may only be authenticated for that user when using the same device. Therefore, the three factors of authentication in our system are biometric (who I am), gesture (what I know) and device (what I have).

3.2 Fuzzy Matching Engine

Since 1975, many engineering applications have been developed based on the use of fuzzy logic [17]. Fuzzy systems handle information closer to the human way, i.e., uncertain, vague or imprecise. In the model proposed by Takagi–Sugeno (TS) [18], the structure of antecedent describes fuzzy regions in the input space, and that of consequent presents non-fuzzy functions of the model inputs. The system may be described for each rule as follows:

R_j:
IF $x_1(k)$ is $F_{1j}, \ldots,$ and $x_n(k)$ is F_{nj},
THEN:
$y_j(k) = y_j$

where y_j is a constant value $X(k) = [x_1(k)x_2(k), \ldots, x_n(k)]^T$ is the input vector of the fuzzy system in the instant k, F_{ij} is the fuzzy set respective to $x_i(k)$ on the rule j, $y_j(k)$ is the output of the model respect to the operating region associated with the rule. If $\mu_{ij}(k)$ is the membership degree of $x_i(k)$ in the fuzzy set F_{ij} and the number of implications or rules is L, the complete model can be described by

$$y(k) = \sum_{j=1}^{L} w_j(k) y_j \tag{1}$$

where

$$w_j(k) = \frac{\bar{\mu}_j(k)}{\sum_{j=1}^{L} \bar{\mu}_j(k)}, \quad \bar{\mu}_j(k) = \prod_{i=1}^{n} \mu_{ij}(k)$$

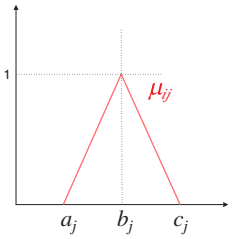


Fig. 1 Triangular function

In this application we have used triangular membership functions, defined as:

$$\mu_{ij}(k) = \begin{cases} 0 & x < a_j \\ \frac{x_i(k)-a_j}{b_j-a_j} & a_j \leq x \leq b_j \\ \frac{c_j-x_i(k)}{c_j-b_j} & b_j < x < c_j \\ 0 & x \geq c_j \end{cases} \tag{2}$$

or in a compact form,

$$\mu_{ij}(k) = \max\left[\min\left(\frac{x_i(k)-a_j}{b_j-a_j}, \frac{c_j-x_i(k)}{c_j-b_j}\right), 0\right] \tag{3}$$

where a_j, b_j and c_j are parameters which define the triangular function, as shown in Fig. 1.

A position is composed of two numbers. The prototype will be composed of N points with the positions

$$P = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$$

A first prototype to be registered is built by the average of the position of several gestures given by the registration process. The FME will make an index using the degree of membership of each pattern point to the prototype point.

There will be a fuzzy number defined for each prototype coordinate:

$$\tilde{P} = \{(\tilde{x}_1, \tilde{y}_1), (\tilde{x}_2, \tilde{y}_2), \ldots, (\tilde{x}_N, \tilde{y}_N)\}$$

The fuzzy number will be defined for the couple $\{b, d\}$ where b is the representative crisp number of $\tilde{b}$ and d will be an adjusting parameter which defines the distance $c - a$. In order to simplify the application, we will set it up with the same value for all the fuzzy numbers, calling it the *fuzziness* parameter.

Using a rule like: IF x_i IS $\tilde{x}_i$ THEN $y = 1$, the degree of membership $\mu_{\tilde{x}_i}(x_i)$ of the crisp number x_i to the fuzzy number $\tilde{x}_i$ is obtained. Applying the rule to each coordinate gives a set of $\{\mu_{\tilde{x}_i}(x_i),\ \mu_{\tilde{y}_i}(y_i)\}$. Taking into account the sequence order and calculating each degree of membership, the expression

$$\mu = \frac{\sum_{i=1}^{N} \mu_{\tilde{x}_i}(x_i) \cdot \mu_{\tilde{y}_i}(y_i)}{N} \tag{4}$$

yields a matching index for the gesture and feature vector. A threshold value can then be used to establish whether the index value represents a match or not. This parameter is referred to as the *sensitivity*.

3.3 Simulation Result Using the FME

Figures 2 and 3 show examples of matching (after adjusting fuzziness and sensitivity) using the matching index (4).

Before applying the FME, the gesture is normalised to an image of the user's face in terms of orientation and scale so that comparisons can be made. For instance, the vector formed by joining the center of the eyes is a good reference. Figure 4 show how the gesture is matched with different orientations and sizes.

3.4 Computation of a Biometric

There are many types of biometric indicators that could be used within a non-textual authentication system. For example, fingerprint, palm print, iris, DNA, etc. However, facial recognition is a desirable choice because it requires only a camera, which is now a fairly ubiquitous component of modern mobile, laptop and desktop devices. Therefore, facial recognition is a suitable component for a system that will be rolled out across a wide range of modern devices.

Since the FME prefers to work with a vector of numerical values, a simple biometric that distils a face down to five numbers was chosen. Using OpenCV's [19] object detection library and, specifically, cascade classifiers, four prominent features of the face are detected. These are namely, the nose, the mouth and the two eyes. Each is defined by the rectangular region that encloses it. By computing the distance between the centre of each of these rectangles, a vector of six values is obtained. By assuming one of these distances to be of unit length and normalising the other values against it, a biometric descriptor comprising five meaningful values remains.

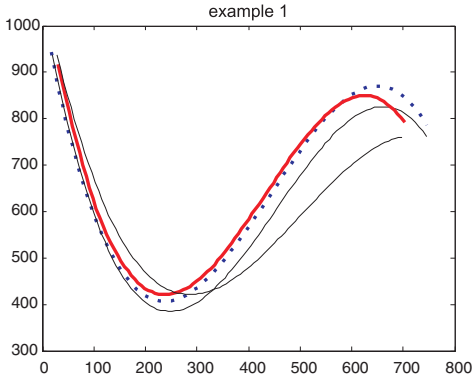


Fig. 2 Example of matching. *Dot line* prototype; *Thick solid line* matched gesture

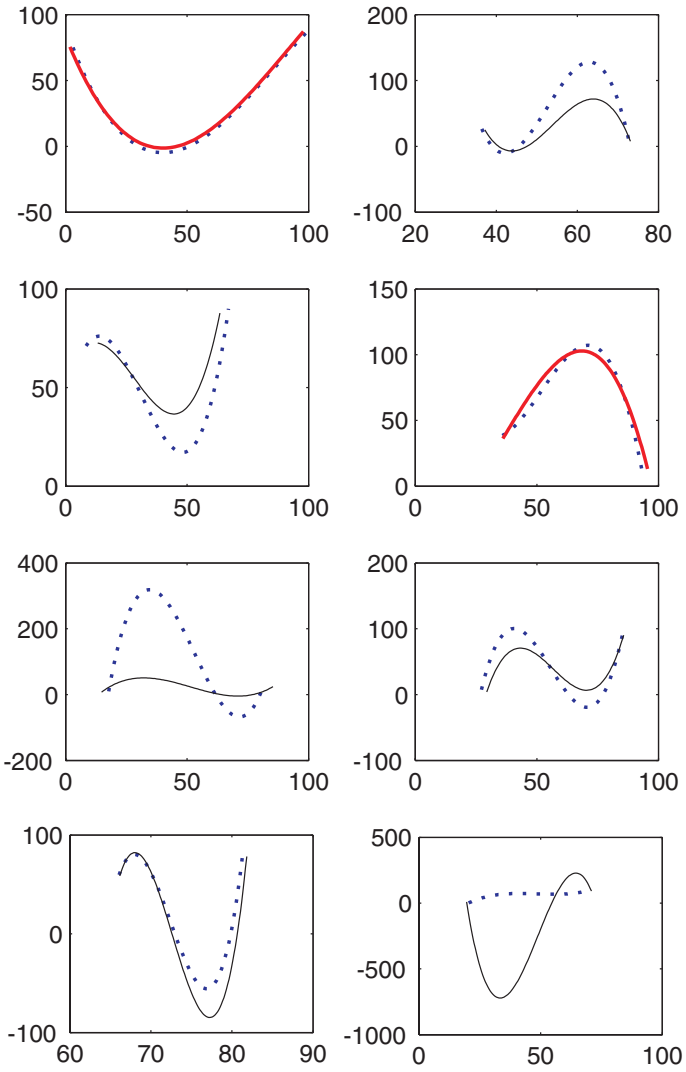


Fig. 3 Examples of matching. *Dot line* prototype; *Thick solid line* matched gesture

3.5 Gesture Capture

A gesture can be described as a sequence of coordinate pairs and is captured while the user completes a drag operation using an available pointing device. On a mobile or tablet device, this will normally be accomplished by touching and dragging on the screen while a desktop computer user may use a mouse or a trackball. On a laptop, perhaps all of these options may be available. In any case, after the

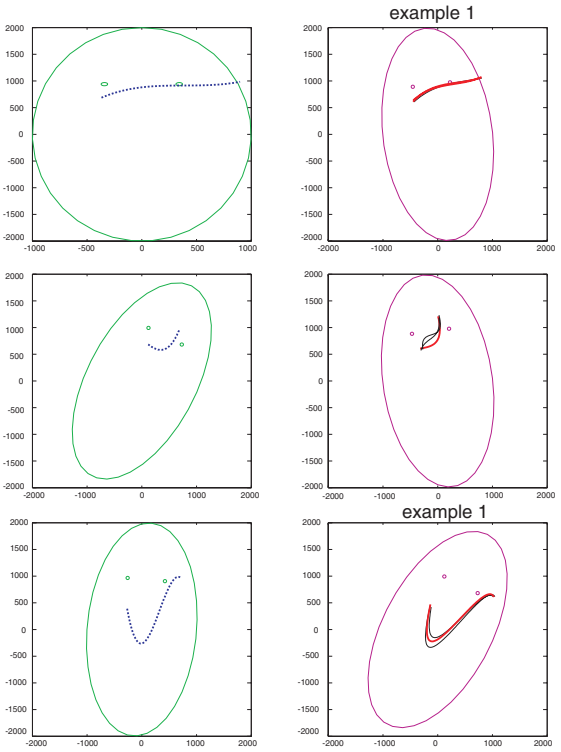


Fig. 4 Example of matching after normalisation. *Dot line* prototype; *Thick solid line* matched gesture

drag operation is completed, a sequence of coordinate pairs will have been captured. In their raw form, these normally represent absolute pixel positions on the device and there may be very many or very few pairs depending upon whether the gesture was drawn slowly or quickly.

To make the gestures more easily comparable, they are first standardised. This simply involved adding extra points or removing extraneous points in order to achieve some predetermined number of coordinate pairs. It is important that the process of standardisation does not materially alter the overall shape of the gesture. It is possible to use the FME to compare a gesture pattern to a stored prototype as long as both have been standardised to the same number of points.

The final step to ensure that gestures can be compared in a meaningful and repeatable manner is to normalise them to the biometric. In this system, a gesture will always be associated with a biometric descriptor. Normalisation takes the gesture out of the device specific, pixel based coordinate system that it originates in and converts it to a space that is determined by the size, location and orientation of the biometric. For this purpose, the vector joining the centre of the eyes is used. This vector defines the unit length along the x-axis in the normalised coordinate space. This allows for the natural variations that will result from users presenting themselves to the camera inconsistently. Perhaps sometimes to one side or to the other, perhaps sometimes closer or farther away. The result is that the image of the

face—the part of the image that generates the biometric will often appear in different parts of the overall camera frame and may take up differing proportions of it. Normalisation eliminates these differences, essentially ensuring that all biometric-gesture pairs are meaningfully comparable. In simple terms, if you have a bigger head, you need to draw a bigger hat.

3.6 Parameterisation

It has been stated that the FME is controlled by simply two parameters, namely fuzziness and sensitivity. The former is applied to the individual differences between elements within a prototype-pattern pair while the latter applies to the aggregation of the scores determined from these differences. However, within this system, it is clear that the FME is used in a number of different contexts and that different fuzziness-sensitivity parameter pairs may be needed for some of these various contexts.

Broadly speaking, the FME is used in two main roles, namely biometric recognition and gesture recognition. However, these roles are performed in two distinct life-cycle phases of the system, namely registration and authentication. Arguably, the system may be more or less lenient depending on the life-cycle phase, thus requiring different parameter pairs for the two roles. Furthermore, as has been previously stated, during registration, the biometric is checked for similarity with other biometrics in the registration process. However, prior to this it is checked for uniqueness against other stored biometrics in the database. This introduces yet another context, which is distinct from all of the others in that it tests for uniqueness (or, more correctly, absence of similarity) as opposed to similarity.

Thus, there are five distinct contexts in which the FME is used and for which an independent fuzziness-sensitivity parameter pair can be defined. Table 1 enumerates the ten possible parameter values and also shows the name ascribed to each parameter within the system. It can be seen from this table that each of the parameters is not independently variable within our system. Rather, the fuzziness for biometric matching, F_b is repeated across *all* biometric matching contexts.

Table 1 Parameters for controlling behaviour of the fuzzy matching engine

	Phase	Object	Comparison	Parameter	Name
1	Reg	Biometric	Similarity	Fuzziness	F_b
2	Reg	Biometric	Similarity	Sensitivity	S_1
3	Reg	Gesture	Similarity	Fuzziness	F_g
4	Reg	Gesture	Similarity	Sensitivity	S_2
5	Reg	Biometric	Uniqueness	Fuzziness	F_b
6	Reg	Biometric	Uniqueness	Sensitivity	S_3
7	Auth	Biometric	Similarity	Fuzziness	F_b
8	Auth	Biometric	Similarity	Sensitivity	S_4
9	Auth	Gesture	Similarity	Fuzziness	F_g
10	Auth	Gesture	Similarity	Sensitivity	S_5

Similarly, F_g, the fuzziness for gesture matching is a single value used in all gesture matching contexts. In contrast, the sensitivities for the five FME contexts are independently variable. Therefore, our system uses a total of seven parameters to control fuzzy matching; two independent fuzziness parameters (F_b, F_g) and five independent sensitivity parameters (S_{1-5}).

3.7 Gradual Migration of Prototype

By the very nature of the system, neither a biometric descriptor, nor a gesture will ever be an identical image of the stored prototype. It is expected that both kinds of pattern will differ from their prototype at all authentication attempts. However, an additional feature of our system allows for the gradual migration the prototype itself in response to the successfully authenticated patterns. A moving window retaining the previous n successfully logged in biometric descriptors and gestures is maintained. At each successful authentication, the newest pair of patterns is added to this window and the oldest is removed. From the window, a mean pattern is computed for both biometric and gesture and this becomes the new prototype that will be compared against during the next authentication attempt. This caters for the scenario that a user may register with a very carefully drawn gesture but that over time, as they become accustomed to using the system, they may adopt a more casual approach to repeating the gesture. However, the system still retains the original prototype that was registered and it is possible to raise an alarm if a user's biometric or gesture has *crept* too far from its original representation. Although we implement this functionality for both biometric and gesture, in practice, we expect that it is really only useful in the latter context.

3.8 Forgotten Gestures

Just as the user of a traditional textual authentication system may forget their secret password, it may occur that a user of our system forgets, or is unable to satisfactorily repeat their registered gesture. In this case, we offer the facility to reset the gesture component of the user's login credentials. When this happens, the user is invited to provide a new gesture, which is analogous to providing a new password within a textual system. The process is similar to registration in that the gesture must be repeated three times on top of three different images. However, in this mode, the biometric is not stored as a prototype but rather compared for similarity against the existing stored prototype. Similarly in this mode, the device id is also checked to confirm that it is the correct device. Therefore, only one of the three factors is reset while the other two serve as authentication during this process. Additional security can be provided by, for example, ensuring that the gesture reset must be performed within a certain duration after the reset is issued.

In principle, it would be possible to apply this strategy to any of the three factors. A user may wish to move their account to a different device. In this case, a device reset could be issued, allowing the user to authenticate themselves using only biometric and gesture on a new device. Similarly, but perhaps less realistically, the biometric factor could be reset, allowing the user to register a new biometric while authenticating themselves by gesture and device id.

4 Implementation and Results

The case studies for applying this research work are diverse, ranging from gaining local access on a native application to authenticating against cloud systems. We also need to consider usability and security constraints, for example whether facial detection takes place remotely or locally. Consequently, there are a number of different architectures that can be implemented around the core of the fuzzy matching engine. To cover as many cases as possible we have implemented two separate architectures discussed below.

The first approach (Fig. 5) is a cloud based architecture. We used an HTML5 client to capture the face and the gesture. The WebRTC [20] standard enables us to take a photo either by a single touch event or when a smile is detected. The user then draws their gesture, which is handled by mouse motion or touch events. Both the bitmap of the face and the array of co-ordinates for the gesture are then sent to the server with REST calls. The server uses the OpenCV library to extract the biometric data which, along with the gesture are passed to the fuzzy engine for authentication. The server is implemented using Java with the Spring Framework and runs on a Glassfish 4 Server, while data are stored in a CouchDB database.

The advantages of this architecture are firstly security, since authentication and data storage are performed remotely and secondly, flexibility deriving from a cloud-like deployment. On the other hand, it is dependent on network connectivity. The client-server architecture means that many unsuitable pictures may be sent to the server before a face is detected. This can means that until the user is familiar

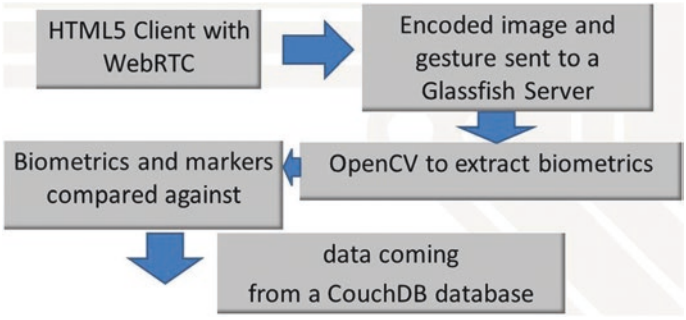


Fig. 5 Cloud implementation

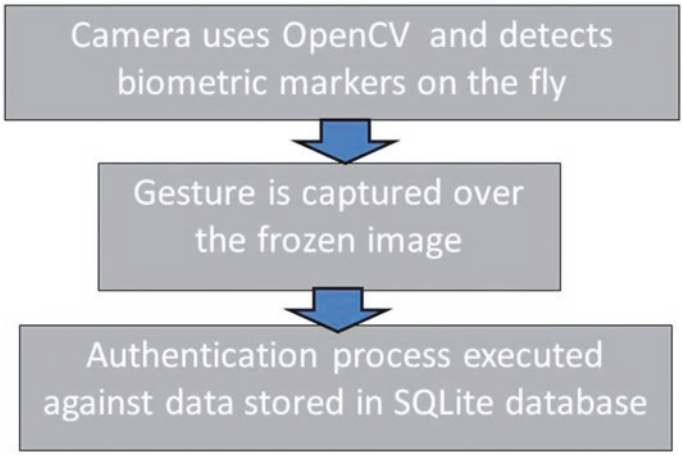


Fig. 6 Android implementation

with the conditions under which successful recognition are likely, using the system can be slow.

The second implementation (Fig. 6) concerns a native application currently for Android devices. OpenCV is once again employed however, the biometric extraction happens in real time as every frame in the video stream is assessed. Preliminary results show that after a short period of self-training, the time required for a user to achieve a sufficient picture is much less than one second. The picture of the face is then frozen for the gesture to take place and similarly to the previous case, data are passed to the fuzzy engine. In this implementation, a SQlite database is used. The strong points of this method are speed and non-reliance on network connectivity. However, having the authentication data locally may be a security vulnerability.

We chose to work on these cases in order to produce a set of modules that could be used in a hybrid implementation in the future. For example, we could use the OpenCV on a native application which sends data to a remote server.

5 Case Study

A pilot study has been conducted for the HTML5 implementation having the main focus to engage with a cohort of people from age 18 to 65 and to observe their interaction with the user experience. All participants were furnished with a basic list of instructions and asked to complete a short online survey.

The pilot accommodated participants with a variety of devices to engage with the system, namely a laptop with external webcam, a tablet and a smart phone. During the course of this 3 day pilot, 19 participants successfully registered on the

system. Of these, 17 participants experienced at least one successful login while 2 participants failed to login. In observing the participants, problems arose when using the laptop and webcam, due primarily to the positioning of the webcam and the laptop touchpad. Eye contact with the system is limited in this scenario. Best results were observed when using a tablet and smart phones. Nine participants successfully completed the online survey, five participants did not complete the survey while five participants failed to engage with the survey at all.

The most obvious of these finding was the necessity to use a tablet or a smart phone. A laptop with external webcam connected will work but will require additional patience and attention to detail from the end user. Most participants agreed that their experience was a positive one. Most agreed that they could use such an access system when using their laptops, tablets, kindle, and smart phones. They were not so confident in using the system when under time constraints or in a scenario when others are waiting to access the same device (an ATM for example).

6 Conclusion

We presented a system for multi-factor authentication based on a fuzzy matching engine. We applied fuzzy matching in two factors namely, biometric (facial recognition) and knowledge (gesture). Non-textual authentication differs from the traditional username-password approach; there is no unique matching and moreover there is a weak dependency between the biometric and the knowledge part. We exploited the latter one by normalising the gesture over the face.

We also defined the parameters that influence the security of this fuzzy-based approach and we outlined its implementation both as a cloud-based or native application. While possible areas of application are limitless, for the foreseeable future we consider e-learning and people with special needs.

Future work involves developing a training system for automatically defining values for fuzziness and sensitivity given certain security constraints, the use of other biometric techniques (e.g. fingerprints) and incorporation within the core of operating systems.

Acknowledgments This work was supported by Enterprise Ireland and carried out under the intellectual property of Sensipass Ltd. Patent Publication No. WO/2012/164385 Method and Computer Program for Providing Authentication to Control Access to a Computer System, Roman Sirota (UA), Michael J. Hill (US) and Thomas R. Ruddy (US).

References

1. Ross, T.J.: Fuzzy Logic with Engineering Applications, Wiley, New York (2004). ISBN: 0470860758. http://www.worldcat.org/isbn/0470860758
2. Chi, Z., Yan, H., Pham, T.: Fuzzy algorithms: with applications to image processing and pattern recognition. In: Advances in Fuzzy Systems—Applications and Theory, vol. 10. World Scientific (1996). ISBN: 9810226977, 9789810226978

3. Martino, F.D., Sessa, S.: Image matching by using fuzzy transforms. Adv. Fuzzy Syst. **2013**(760704), 10 (2013). doi:10.1155/2013/760704
4. Perfilieva, I.: Fuzzy Transforms, Transactions on Rough Sets II, vol. 3135, pp. 63–81. Lecture Notes in Computer Science. Springer (2005). ISBN: 978-3-540-23990-1
5. Bloch, I: Fuzzy relative position between objects in image processing: a morphological approach. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (1999)
6. Bloch, I., Ralescu, A.: Directional relative position between objects in image processing: a comparison between fuzzy approaches. Pattern Recogn. **36**, 1563–1582 (2003)
7. Tan, Q., Akimoto, M.: Fuzzy matching for robot localization. In: Proceedings of IROS. IEEE (1996). ISBN: 96. 0-7803-3213-X
8. Jinwen, T., Jianzhong, H., Jian, L., Dchua, L.: Image matching based on fuzzy information. In: 3rd International Conference on Signal Processing 1996, vol. 2, pp. 946–949. 14–18 Oct 1996. doi:10.1109/ICSIGP.1996.566246
9. Wu, H., Chen, Q., Yachida, M.: Face detection from color images using a fuzzy pattern matching method. IEEE Trans. Pattern Anal. Mach. Intell. **21**(6), 557–563 (1999)
10. Surajit C., Kris, G., Venkatesh, G., Rajeev, M.: Robust and efficient fuzzy match for online data cleaning. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD'03) (2003)
11. Zvi G., Alberto, A.: Pattern Matching Algorithms. Oxford University Press, Oxford (1997). ISBN: 0-19-511367-5
12. Mustafa, A.A.Y.: Fuzzy shape matching with boundary signatures. Pattern Recogn. Lett. **23**, 14731482 (2002)
13. Ukkonen, E.: Algorithms for approximate string matching. Inf. Control **64**, 10018 (1985). doi:10.1016/S0019-9958(85)80046-2
14. Li, Z.K., Xu, L.J., Fang, J., Peng, Q.J., Wang, M.: Research on the surrounding traffic flow of railway station based on License Plate Recognition and fuzzy matching. IEEE (2011). 978-1-61284-109-0
15. Schultz P.T., Sartini, R.A.: Multi factor authentication method and system for multi-factor biometric authentication. US 20130227651 A1 (2012)
16. Vyas, R., Garg, G.: Face recognition using feature extraction and neuro-fuzzy techniques. Int. J. Electron. Comput. Sci. Eng. (2013). ISSN: 2277–1956
17. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
18. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst., Man Cybern. **15**(1), 116132 (1985). http://www.hi.cs.meiji.ac.jp/takagi/paper/TS-MODEL.tar.gz
19. http://www.openCV.org/
20. http://www.webrtc.org/
21. http://argodata.com/solutions/analytics/fuzzy-search/
22. http://dev.w3.org/html5/html-author/
23. Huang, X et al.: A generic framework for three-factor authentication: preserving security and privacy in distributed systems. IEEE Trans. Parallel Distrib. Syst. **22**(8), 1390–1397 (2011)
24. Pulli, K., Baksheev, A., Kornyakov, K., Eruhimov, V.: Real-time computer vision with OpenCV. Commun. ACM (CACM) **55**(6), 61–69 (2012)

Efficient Graph-Based Volumetric Segmentation

Dumitru Dan Burdescu, Marius Brezovan, Liana Stanescu and Cosmin Stoica Spahiu

Abstract The emergence and increasing importance of digital society increased the role of software applications in smart environments. Associated with these paradigms are a multitude of applications that generate and require analysis of massive volumes of diverse, heterogeneous, complex, and distributed data. The problem of partitioning images into homogenous regions or semantic entities is a basic problem for identifying relevant objects. There is a wide range of computational vision problems for 2D images that could use of segmented images. However the problems of 3D image segmentation and grouping remain great challenges for computer vision. Visual segmentation is related to some semantic concepts because certain parts of a scene are pre-attentively distinctive and have a greater significance than other parts. Many approaches aim to create large regions using simple homogeneity criteria based only on color or texture. However, 3D applications for such approaches are limited as they often fail to create meaningful partitions due to the computation complexity. We are introducing new algorithm for spatial segmentation based on Virtual Tree-Hexagonal Structure constructed on the image voxels. Then the paper depicts a Spatial Segmentation Algorithm. Spatial Segmentation Algorithm contains many other algorithms but only Color-based segmentation algorithm is presented based on the limited space of paper. Then the paper describes the Computational Complexity Analysis of the Color-Based Spatial Segmentation Algorithm.

D.D. Burdescu (✉) · M. Brezovan · L. Stanescu · C.S. Spahiu
University of Craiova, Bd. Decebal 107, Craiova, Romania
e-mail: burdescu_dumitru@software.ucv.ro

M. Brezovan
e-mail: brezovan_marius@software.ucv.ro

L. Stanescu
e-mail: stanescu_liana@software.ucv.ro

C.S. Spahiu
e-mail: stoica_cosmin@software.ucv.ro

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_9

Keywords Spatial segmentation · Graph-based segmentation · Color segmentation

1 Introduction

The emergence and increasing importance of digital society, cyber-physical systems, and semantic, pervasive, and mobile computing are expanding the role of software and applications in smart or intelligence environments. Associated with these paradigms are instruments, sensors, and a multitude of applications that generate and require analysis of massive volumes of diverse, heterogeneous, complex, and distributed data.

The problem of partitioning images into homogenous regions or semantic entities is a basic problem for identifying relevant objects. There is a wide range of computational vision problems for planar images that could use of segmented images. However the problems of volumetric image segmentation and grouping remain great challenges for computer vision. For instance intermediate-level vision problems motion estimation and tracking require determination of salient objects from frames. The major concept used in graph-based volumetric segmentation method is the concept of homogeneity of volumes and thus the edge weights are based on color distance.

Visual segmentation is related to some semantic concepts because certain parts of a scene are pre-attentively distinctive and have a greater significance than other parts. Many approaches aim to create large regions using simple homogeneity criteria based only on color or texture. However, spatial applications for such approaches are limited as they often fail to create meaningful partitions due to either the complexity of the scene or difficult lighting conditions. Higher-level problems such as object recognition and image indexing can also make use of segmentation results in matching, to address problems such as figure-ground separation and recognition by parts. In both intermediate level and higher-level vision problems, contour detection of objects in real images is a fundamental problem.

For example, salient objects are defined as visually distinguishable image compounds that can characterize visual properties of corresponding object classes and they have been proposed as an effective middle-level representation of image content. An important approach for salient object detection is segmentation for planar and volumetric images, and developing an accurate image segmentation technique which partitions image into salient visual objects is an important step toward salient object detection. As a consequence we consider that a volumetric segmentation method can detect visual objects from images if it can detect at least the most objects.

We are introducing new method for volumetric segmentation based on Virtual Tree-Hexagonal Structure constructed on the image voxels. We develop a visual feature-based method which uses a spatial graph constructed on cells of prisms with tree-hexagonal structure containing less than half of the image voxels in order to determine a forest of spanning trees for connected component representing visual objects. Thus the volumetric image segmentation is treated as a spatial graph partitioning problem.

We determine the spatial segmentation of a color image in two distinct steps: a pre-segmentation step when only color information is used in order to determine an initial volumetric segmentation, and a syntactic-based segmentation step when we define a predicate for determining the set of nodes of connected components based both on the color distance and geometric properties of volumes representing visual objects.

The novelty of our contribution concerns: (a) the virtual cells of prisms with tree- hexagonal structure used in the unified framework for volumetric image segmentation, (b) the using of maximum spanning trees for determining the set of nodes representing the connected components in the pre-segmentation step, (c) a method to determine the thresholds used both in the pre-segmentation and in the spatial segmentation step, and (d) an automatic stopping criterion used in the volumetric segmentation step.

In addition our volumetric segmentation algorithm produces good results from both from the perspective perceptual grouping, and from the perspective of determining homogeneous in the input images. We refer the term of perceptual grouping as a general expectation for volumetric segmentation algorithms to produce perceptually coherent segmentation of volumes at a level comparable to humans.

Of course into Volumetric Segmentation Method there are many other algorithms but only Color-based segmentation algorithm and Syntactic segmentation algorithm are designed based on the space of paper. Based on number of the tree-edges of the input spatial graph G = (V, E) of the color-based algorithm, and the number of the vertices of input graph we say and prove that the time of Volumetric Segmentation Algorithm is linear.

Our previous works for digital planar images are related to other works in the sense of pair-wise comparison of region similarity. The key to the whole algorithm of volumetric segmentation is the honeycomb cells. We present the original and efficient algorithm of volumetric segmentation methods and honeycomb used is the first run into Segmentation Volumetric Method.

1.1 Related Work

In this section we briefly consider some of the related works that are most relevant related to our approach.

Someone determined the normalized weight of an edge by using the smallest weight incident on the vertices touching that edge [1]. Other methods for planar images [2, 3] use adaptive criterion that depends on local properties rather than global ones. In contrast with the simple graph-based methods, cut-criterion methods capture the non-local cuts in a graph are designed to minimize the similarity between pixels that are being split [4, 5]. The normalized cut criterion [5] takes into consideration self similarity of regions. An alternative to the graph cut approach is to look for cycles in a graph embedded in the image plane. In [6, 7] the quality of each cycle is normalized in a way that is closely related to the

normalized cuts approaches. Other approaches to digital planar image segmentation consist of splitting and merging regions according to how well each region fulfills some uniformity criterion. Such methods [8] use a measure of uniformity of a region. In contrast [2, 3] use a pair-wise region comparison rather than applying a uniformity criterion to each individual region. Complex organizing phenomena can emerge from simple computation on these local cues [9]. A number of approaches to segmentation are based on finding compact regions in some feature space [10]. Recent techniques for planar digital images using feature space regions [11, 12] first transform the data by smoothing it in a way that preserves boundaries between regions. We use different measures for internal contrast of a connected component and for external contrast between two connected components than the measures used in [13].

Our previous works [11, 14–16] are related to the works in [2, 3] in the sense of pair-wise comparison of region similarity. In these papers we extend our previous work by adding a new step in the spatial segmentation algorithm that allows us to determine regions closer to it.

The internal contrast of a component C represents the maximum weight of edges connecting vertices from C, and the external contrast between two components represents the maximum weight of edges connecting vertices from these two components. These measures are in our opinion closer to the human perception. We use maximum spanning tree instead of minimum spanning tree in the pre-segmentation step in order to manage external contrast between connected components.

2 Constructing a Virtual Tree-Hexagonal Structure

The low-level system for spatial image segmentation and boundary extraction of visual objects described in this section can be designed to be integrated in a general framework of indexing and semantic image processing. The framework uses color and geometric features of image volumes in order to: (a) determine visual objects and their spatial surface, and also (b) to extract specific color and geometric information from these objects to be further used into a higher-level image processing system.

The pre-processing module is used mainly to blur the initial RGB spatial image in order to reduce the image noise by applying a spatial Gaussian kernel [17]. Then the segmentation module creates virtual cells of prisms with tree-hexagonal structure defined on the set of the image voxels of the input spatial image and a spatial grid graph having tree-hexagons as cells of vertices. In order to allow a unitary processing for the multi-level system at this level we store, for each determined component C, the set of the tree-hexagons contained in the region associated to C and the set of tree-hexagons located at the boundary of the component. In addition for each component the dominant color of the region is extracted. This color will be further used in the post-processing module if any. The surface

extraction module determines for each segment of the image its boundary. The boundaries of the de determined visual objects are closed surfaces represented by a sequence of adjacent tree-hexagons. At this level a linked list of voxels representing the surface is added to each determined component. The post-processing module (if any) extracts representative information for the above determined visual objects and their surfaces in order to create an efficient index for a semantic image processing system.

A volumetric image processing task contains mainly three important components: acquisition, processing and visualization. After the acquisition stage an image is sampled at each point on a three dimensional grid storing intensity or color information and implicit location information for each sample. We do not use a hexagonal lattice model because of the additional actions involving the double conversion between square and tree-hexagonal voxels. However we intent to use some of the advantages of the tree-hexagonal grid such as uniform connectivity. This implies that there will be less ambiguity in defining spatial surface and volumes [18]. As a consequence we construct a virtual tree-hexagonal structure over the voxels of an input image, as presented in Fig. 1. This virtual tree-hexagonal grid is not a tree-hexagonal lattice because the constructed hexagons are not regular.

Let I be an initial volumetric image having the dimension $h \times w \times z$ (e.g. a matrix having h rows, w columns and z deep of matrix voxels). In order to construct a tree-hexagonal grid on these voxels we retain an eventually smaller image with:

$$\begin{aligned} h' &= h - (h-1) \bmod 2, \\ w' &= w - w \bmod 4, \\ z' &= z. \end{aligned} \tag{1}$$

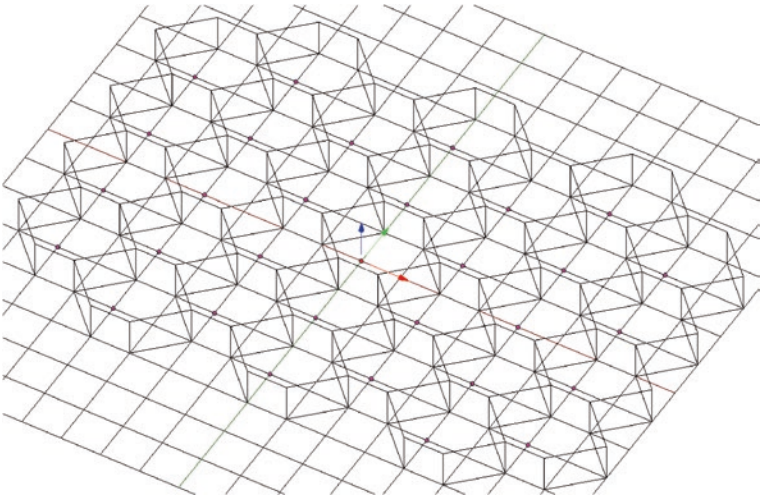


Fig. 1 Virtual tree-hexagonal structure constructed on the image voxels

In the reduced image at most the last line of voxels and at most the last three columns and deep of matrix of voxels are lost, assuming that for the initial image $h > 3$ and $w > 4$ and $z \geq 1$, that is a convenient restriction for input images.

Each tree-hexagon from the tree-hexagonal grid contains 16 voxels: such 12 voxels from the frontier and four interior frontiers of voxels. Because tree-hexagons voxels from an image have integer values as coordinates we select always the left up voxel from the four interior voxels to represent with approximation the gravity center of the tree-hexagon, denoted by the pseudo-gravity center.

We use a simple scheme of addressing for the tree-hexagons of the tree-hexagonal grid that encodes the spatial location of the pseudo-gravity centers of the tree-hexagons as presented in Fig. 1.

Let $h \times w \times z$ the three dimension of the initial volumetric image verifying the previous restriction. Given the coordinates $\langle l, c, d \rangle$ of a voxel p from the input volumetric image, we use the linearized function,

$$ip_{h,w,z}(l, c, d) = (l - 1) \times w \times z + (c - 1) \times z + d, \tag{2}$$

in order to determine an unique index for the voxel.

It is easy to verify that the function ip defined by the Eq. 2 is bijective. Its inverse function is given by:

$$ip^{-1}_{h,w,z}(k) = \langle l, c, d \rangle, \tag{3}$$

where:

$$l = k/(w \times z), \tag{4}$$

$$c = (k - (l - 1) \times w \times z)/z, \tag{5}$$

$$d = k - (l - 1) \times w \times z + (c - 1) \times z. \tag{6}$$

Relations 4, 5, and 6 allow us to uniquely determine the coordinates of the voxel representing the pseudo-gravity center of a tree-hexagon specified by its index (its address). In addition these relations allow us to determine the sequence of coordinates of all sixteen voxels contained into a tree-hexagon with an address k.

The sub-sequence ps of the voxels representing the pseudo-gravity center and the function ip defined by the relation 2 allow to determine the sequence of the tree-hexagons that is used by the segmentation and surface detection algorithms. After the processing step the Relations 3, 4, 5, and 6 allow to up-date the voxels of the spatial initial spatial image for the visualization step.

Each tree-hexagon represents an elementary item and the entire virtual tree-hexagonal structure represents a spatial grid graph, $G = (V, E)$, where each tree-hexagon H in this structure has a corresponding vertex $v \in V$. The set E of edges is constructed by connecting tree-hexagons that are neighbors in a 8-connected sense. The vertices of this graph correspond to the pseudo-gravity centers of the

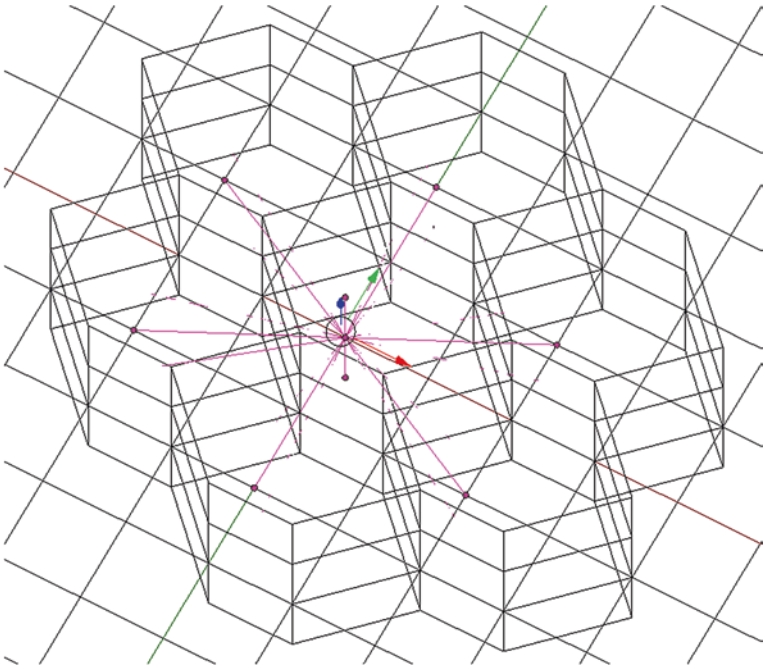


Fig. 2 The grid graph constructed on the pseudo-gravity centers of the tree-hexagonal grid

hexagons from the tree-hexagonal grid and the edges are straight lines connecting the pseudo-gravity centers of the neighboring hexagons, as presented in Fig. 2.

There are two main advantages when using tree-hexagons instead of all voxels as elementary piece of information:

- The amount of memory space associated to the graph vertices is reduced. Denoting by np the number of voxels of the initial spatial image, the number of the resulted tree-hexagons is always less than $np/8$, and thus the cardinal of both sets V and E is significantly reduced;
- The algorithms for determining the visual objects and their surfaces are much faster and simpler in this case.

We associate to each tree-hexagon H from V two important attributes representing its dominant color and the coordinates of its pseudo-gravity center, denoted by $c(h)$ and $g(h)$. The dominant color of a tree-hexagon is denoted by $c(h)$ and it represents the color of the voxel of the tree-hexagon which has the minimum sum of color distance to the other twenty voxels. Each tree-hexagon H in the tree-hexagonal grid is thus represented by a single point, $g(h)$, having the color $c(h)$. By using the values $g(h)$ and $c(h)$ for each tree-hexagon information related to all voxels from the initial image is taken into consideration by the spatial segmentation algorithm.

3 Volumetric Segmentation Algorithm

Let $V = \{h_1, \ldots, h_{|V|}\}$ be the set of tree-hexagons constructed on the spatial image voxels as presented in previous section and $G = (V, E)$ be the undirected spatial grid-graph, with E containing pairs of honey-beans cell (tree-hexagons) that are neighbors in a 8-connected sense. The weight of each edge $e = (h_i, h_j)$ is denoted by $w(e)$, or similarly by $w(h_i, h_j)$, and it represents the dissimilarity between neighboring elements h_i and h_j in a some feature space. Components of an image represent compact volumes containing voxels with similar properties. Thus the set V of vertices of the graph G is partitioned into disjoint sets, each subset representing a distinct visual object of the initial image.

As in other graph-based approaches [15] for planar images we use the notion of segmentation of the set V. A segmentation, S, of V is a partition of V such that each component $C \in S$ corresponds to a connected component in a spanning sub-graph $GS = (V, ES)$ of G, with $ES \subseteq E$.

The set of edges $E - ES$ that are eliminated connect vertices from distinct components. The common boundary between two connected components $C^{'}, C^{''} \in S$ represents the set of edges connecting vertices from the two components:

$$cb(C', C'') = \{(hi, hj) \in E | hi \in C', \quad hj \in C''\}. \tag{7}$$

The set of edges $E - ES$ represents the boundary between all components in S. This set is denoted by $bound(S)$ and it is defined as follows:

$$bound(S) = \bigcup_{C', C'' \in S} cb(C', C''). \tag{8}$$

In order to simplify notations throughout the paper we use Ci to denote the component of a segmentation S that contains the vertex $hi \in V$.

We use the notions of segmentation too fine and too coarse as defined in [2] that attempt to formalize the human perception of salient visual objects from an image. A segmentation S is too fine if there is some pair of components $C^{'}, C^{''} \in S$ for which there is no evidence for a boundary between them. A segmentation S is too coarse when there exists a proper refinement of S that is not too fine. The key element in this definition is the evidence for a boundary between two components.

The goal of a segmentation method is to determine a proper segmentation, which represent visual objects from a volumetric image.

Definition 1 Let $G = (V, E)$ be the undirected spatial graph constructed on the tree-hexagonal structure of an image, with $V = \{h_1, \ldots, h_{|V|}\}$. A proper segmentation of V, is a partition S of V such that there exists a sequence $\langle S^i, S^{i+1}, \ldots, S^{f-1}, S^f \rangle$ of segmentations of V for which:

- $S = S^f$ is the final segmentation and S^i is the initial segmentation,
- S^j is a proper refinement of S^{j+1} (i.e., $S^j \subset S^{j+1}$) for each $j = i, \ldots, f - 1$,
- segmentation S^j is too fine, for each $j = i, \ldots, f - 1$,
- any segmentation S^l such that $S^f \subset S^l$, is too coarse,
- segmentation S^f is neither too coarse nor too fine.

In the above definition S^a is a refinement of S^b in the sense of partitions, i.e. every set in S^a is a subset of one of the sets in S^b. We say that S^a is a proper refinement of S^b if S^a is a refinement of S^b and $S^a \neq S^b$. In the case of a proper refinement, S^a is obtained by splitting one or more components from S^b, or similarly, S^b is obtained by merging one or more components from S^a. Let $C^{'}$, $C^{''} \in S^a$ be two components obtained by splitting a component $C \in S^b$. In this case $C^{'}$ and $C^{''}$ have a common boundary, $cb(C', C'') \neq \emptyset$.

Our segmentation algorithm starts with the most refined segmentation, $S^0 = \{\{h_1\}, \ldots, \{h_{|V|}\}\}$ and it constructs a sequence of segmentations until a proper segmentation is achieved. Each segmentation S^j is obtained from the segmentation S^{j-1} by merging two or more connected components for there is no evidence for a boundary between them. For each component of a segmentation a spanning tree is constructed and thus for each segmentation we use an associated spanning forest.

The evidence for a boundary between two components is determined taking into consideration some features in some model of the image. When starting, for a certain number of segmentations the only considered feature is the color of the volumes associated to the components and in this case we use a color-based region model. When the components became complex and contain too much tree-hexagons, the color model is not sufficient and geometric features together with color information are considered. In this case we use a syntactic based with a color-based region model for volumes. In addition syntactic features bring supplementary information for merging similar volumes in order determine salient objects.

For the sake of simplicity we will denote this region model as syntactic-based region model.

As a consequence, we split the sequence of all segmentations,

$$S_{if} = \langle S^0, S^1, \ldots, S^{k-1}, S^k \rangle, \tag{9}$$

in two different subsequences, each subsequence having a different region model,

$$\begin{aligned} S_i &= \langle S^0, S^1, \ldots, S^{t-1}, S^t \rangle, \\ S_f &= \langle S^t, S^{t+1}, \ldots, S^{k-1}, S^k \rangle, \end{aligned} \tag{10}$$

where S_i represents the color-based segmentation sequence, and S_f represents the syntactic-based segmentation sequence.

The final segmentation St in the color-based model is also the initial segmentation in the syntactic-based region model.

For each sequence of segmentations we develop a different algorithm. Moreover we use a different type of spanning tree in each case: a maximum spanning tree in the case of the color-based segmentation, and a minimum spanning tree in the case of the syntactic-based segmentation. More precisely our method determines two sequences of forests of spanning trees,

$$\begin{aligned} F^i &= \langle F_0, F_1, \ldots, F_{t-1}, F_t \rangle, \\ F^f &= \langle F_{t'}, F_{t'+1}, \ldots, F_{k'-1}, F_{k'} \rangle, \end{aligned} \tag{11}$$

each sequence of forests being associated to a sequence of segmentations.

The first forest from F^i contains only the vertices of the initial graph, $F_0 = (V, \emptyset)$, and at each step some edges from E are added to the forest $F_l = (V, E^l)$ to obtain the next forest, $F_{l+1} = (V, E_{l+1})$. The forests from F^i contain maximum spanning trees and they are determined by using a modified version of Kruskal's algorithm [19], where at each step the heaviest edge (u, v) that leaves the tree associated to u is added to the set of edges of the current forest.

The second subsequence of forests that correspond to the subsequence of segmentations Sf contains forests of minimum spanning trees and they are determined by using a modified form of Boruvka's algorithm. This sequence uses as input a new graph, $G^{'} = (V^{'}, E^{'})$, which is extracted from the last forest, F^t, of the sequence F_i. Each vertex v from the set V corresponds to a component C_v from the segmentation S_t (i.e. to a region determined by the previous algorithm). At each step the set of new edges added to the current forest are determined by each tree T contained in the forest that locates the lightest edge leaving T. The first forest from F^f contains only the vertices of the graph $G^{'}$, $F_{t'} = (V', \emptyset)$.

In this section we focus on the definition of a logical predicate that allow us to determine if two neighboring volumes represented by two components, $C_{l'}$ and $C_{l''}$, from a segmentation S_l can be merged into a single component C_{l+1} of the segmentation S_{l+1}.

Two components, $C_{l'}$ and $C_{l''}$, represent neighboring (adjacent) volumes if they have a common spatial surface:

$$\begin{aligned} adj(C_{l'}, C_{l''}) &= \text{true}, \quad \text{if } cb(C_{l'}, C_{l''}) \neq \emptyset, \\ adj(C_{l'}, C_{l''}) &= \text{false}, \quad \text{if } cb(C_{l'}, C_{l''}) = \emptyset. \end{aligned} \tag{12}$$

We use a different predicate for each region model, color based and syntactic-based respectively.

$$PED(e, u) = \sqrt{w_R(R_e - R_u)^2 + w_G(G_e - G_u)^2 + w_B(B_e - B_u)^2}, \tag{13}$$

where the weights for the different color channels, w_R, w_G, and w_B verify the condition $w_R + w_G + w_B = 1$. Based on the theoretical and experimental results on spectral and real world data sets, Gijsenij et al. [20] is concluded that the PED distance with weight-coefficients ($w_R = 0.26$, $w_G = 0.70$, $w_B = 0.04$) correlates significantly higher than all other distance measures including the angular error and Euclidean distance.

In the color model volumes are modeled by a vector in the RGB color space. This vector is the mean color value of the dominant color of tree-hexagons belonging to the regions.

The evidence for a spatial surface between two volumes is based on the difference between the internal contrast of volumes and the external contrast between them [2, 16]. Both notions of internal contrast and external contrast between two volumes are based on the dissimilarity between two colors.

Let h_i and h_j representing two vertices in the graph $G = (V, E)$, and let $w_{col}(h_i, h_j)$ representing the color dissimilarity between neighboring elements h_i and h_j, determined as follows:

$$w_{col}(h_i, h_j) = PED(c(h_i), c(h_j)), \text{ if } (h_i, h_j) \in E, \\ w_{col}(h_i, h_j) = \infty, \quad \text{otherwise}, \tag{14}$$

where $PED(e, u)$ represents the perceptual Euclidean distance with weight-coefficients between colors e and u, as defined by Eq. 13, and $c(h)$ represents the mean color vector associated with the tree-hexagon h. In the color-based segmentation, the weight of an edge (h_i, h_j) represents the color dissimilarity, $w(h_i, h_j) = w_{col}(h_i, h_j)$.

Let S_l be a segmentation of the set V. We define the internal contrast or internal variation of a component $C \in S_l$ to be the maximum weight of the edges connecting vertices from C:

$$IntVar(C) = \max_{(h_i,h_j)\in C}(w(h_i, h_j)). \tag{15}$$

The internal contrast of a component C containing only one tree-hexagon is zero: $IntVar(C) = 0, \text{ if} |C| = 1$.

The external contrast or external variation between two components, $C^{'}, C^{''} \in S$ is the maximum weight of the edges connecting the two components:

$$ExtVar(C', C'') = \max_{(h_i,h_j)\in cb(C',C'')}(w(h_i, h_j)). \tag{16}$$

We chosen the definition of the external contrast between two components to be the maximum weight edge connecting the two components and not to be the minimum weight, as in [2] because: (a) it is closer to the human perception (in the sense of the perception of the maximum color dissimilarity), and (b) the contrast is uniformly defined (as maximum color dissimilarity) in the two cases of internal and external contrast.

The maximum internal contrast between two components, $C^{'}, C^{''} \in S$ is defined as follows:

$$IntVar(C', C'') = \max(IntVar(C'), \ IntVar(C'')). \tag{17}$$

The comparison predicate between two neighboring components $C^{'}$ and $C^{''}$ (i.e., $adj(C^{'}, C^{''}) = true$) determines if there is an evidence for a boundary between $C^{'}$ and $C^{''}$ and it is defined as follows:

$$diff_{col}(C', C'') = true, \quad \text{if } \ ExtVar(C', C'') > IntVar(C', C'') + \tau(C', C'') \\ diff_{col}(C', C'') = false, \quad \text{if } \ ExtVar(C', C'') = IntVar(C', C'') + \tau(C', C''), \tag{18}$$

with the the adaptive threshold $\tau(C', C'')$ is given by

$$\tau(C', C'') = \tau/(\min(|C'|, |C''|), \tag{19}$$

where $|C|$ denotes the size of the component C (i.e. the number of the tree-hexagons contained in C) and the threshold τ is a global adaptive value defined by using a statistical model.

The predicate $diff_{col}$ can be used to define the notion of segmentation too fine and too coarse in the color-based region model.

Definition 2 Let $G = (V, E)$ be the undirected spatial graph constructed on the tree-hexagonal structure of a volumetric image and S a color-based segmentation of V. The segmentation S is too fine in the color-based region model if there is a pair of components $C', C'' \in S$ for which $adj(C', C'') = true \wedge diff_{col}(C', C'') = false$.

Definition 3 Let $G = (V, E)$ be the undirected spatial graph constructed on the tree-hexagonal structure of a volumetric image and S a segmentation of V. The segmentation S is too coarse if exists a proper refinement of S that is not too fine.

There are many existing systems for arranging and describing colors, such as RGB, YUV, HSV, LUV, CIELAV, Munsell system, etc. We decided to use the RGB color space because it is efficient and no conversion is required. Although it also suffers from the non-uniformity problem where the same distance between two color points within the color space may be perceptually quite different in different parts of the space, within a certain color threshold it is still definable in terms of color consistency. We use the perceptual Euclidean distance with weight-coefficients (*PED*) as the distance between two colors.

Let $G = (V, E)$ be the initial graph constructed on the tree-hexagonal structure of a volumetric image. The proposed segmentation algorithm will produce a proper segmentation of V according to the Definition 1. The sequence of segmentations, S_{if}, as defined by Eq. 9, and its associated sequence of forests of spanning trees, F^{if}, as defined by Eq. 11, will be iteratively generated as follows:

- The color-based sequence of segmentations, S^i, as defined by Eq. 10, and its associated sequence of forests, F^i, as defined by Eq. 11, will be generated by using the color-based region model and a maximum spanning tree construction method based on a modified form of the Kruskal's algorithm.
- The syntactic-based sequence of segmentations, S^f, as defined by Eq. 10, and its associated sequence of forests, F^f, as defined by Eq. 11, will be generated by using the syntactic-based model and a minimum spanning tree construction method based on a modified form of the Boruvka's algorithm.

The general form of the segmentation procedure is presented in Algorithm 1

Algorithm 1 Segmentation algorithm

```
1: procedure SEGMENTATION(l, c, d, P, H, Comp)
2:     Input l, c, d, P
3:     Output Comp
4:     H ← CREATEHEXAGONALSTRUCTURE(l, c, d, P)
5:     G ← CREATEINITIALGRAPH(l, c, d, P, H)
6:     CREATECOLORPARTITION(G, H)
7:     G′ ← EXTRACTGRAPH(G)
8:     G′ ← CREATESYNTACTICPARTITION(G, G′)
9:     Comp ← EXTRACTFINALCOMPONENTS(G′)
10: end procedure
```

The input parameters represent the image resulted after the pre-processing operation: the array P of the spatial image voxels structured in l lines, c columns and d depths. The output parameters of the segmentation procedure will be used by the surface extraction procedure: the tree-hexagonal grid stored in the array of tree-hexagons H, and the array Comp representing the set of determined components associated to the salient objects in the input spatial image.

The color-based segmentation and the syntactic-based segmentation are determined by the procedures *CREATECOLORPARTITION* and *CREATESYNTACTICPARTITION* respectively.

The color-based and syntactic-based segmentation algorithms use the tree-hexagonal structure H created by the function *CREATEHEXAGONALSTRUCTURE* over the voxels of the initial spatial image, and the initial triangular grid graph G created by the function *CREATEINITIALGRAPH*. Because the syntactic-based segmentation algorithm uses a graph contraction procedure, *CREATESYNTACTICPARTITION* uses a different graph, G, extracted by the procedure *EXTRACTGRAPH* after the color-based segmentation finishes.

Both algorithms for determining the color-based and syntactic based segmentation use and modify a global variable (denoted by *CC*) with two important roles:

- to store relevant information concerning the growing forest of spanning trees during the segmentation (maximum spanning trees in the case of the color-based segmentation, and minimum spanning trees in the case of syntactic based segmentation),
- to store relevant information associated to components in a segmentation in order to extract the final components because each tree in the forest represent in fact a component in each segmentation S in the segmentation sequence determined by the algorithm.

In addition, this variable is used to maintain a fast disjoint set-structure in order to reduce the running time of the color based segmentation algorithm. The variable CC is an array having the same dimension as the array of hexagons H, which contains as elements objects of the class Tree with the following associated fields:

$$(isRoot, parent, compIndex, frontier, surface, color)$$

The field *isRoot* is a boolean value specifying if the corresponding tree-hexagon index is the root of a tree representing a component, and the field parent represents the index of the tree-hexagon which is the parent of the current tree-hexagon. The rest of fields are used only if the field *isRoot* is true. The field *compIndex* is the index of the associated component.

The field surface is a list of indices of the tree-hexagons belonging to the associated component, while the field frontier is a list of indices of the tree-hexagons belonging to the frontier of the associated component. The field color is the mean color of the tree- hexagon colors of the associated component.

The procedure *EXTRACTFINALCOMPONENTS* determines for each determined component C of *Comp*, the set $sa(C)$ of tree-hexagons belonging to the component, the set $sp(C)$ of tree-hexagons belonging to the frontier, and the dominant color $c(C)$ of the component.

4 Color-Based Segmentation Algorithm

Let $G = (V, E)$ be the undirected spatial graph constructed on the tree-hexagonal structure of a volumetric input image. The proposed color-based segmentation algorithm will produce a proper segmentation of V according to the Definition 1, where the notion of segmentation too fine is given by the Definition 2. The sequence of segmentations, $\langle S^0, S^1, \ldots, S^{t-1}, S^t \rangle$, and its associated sequence of growing forests, $\langle F_0, F_1, \ldots, F_{t-1}, F_t \rangle$, will be iteratively generated, based on a maximum spanning tree construction method. We use a modified form of the Kruskal's algorithm presented in Algorithm 2, where the trees generated at each step represent the connected components of volumetric segmentation.

The input parameters of the color-based segmentation algorithm are the initial graph G and the array H of the tree-hexagons from the tree-hexagonal grid. The output parameter is the list *Bound* of edges representing the boundary of the final spatial segmentation.

The global parameter threshold τ is determinate by using Algorithm 1. This value is used at the line 18 of Algorithm 2, where the expression $\tau(t_i, t_j)$ is given by the Relation 19, where t_i and t_j representing the components C_{t_i} and C_{t_j} respectively.

Because we use maximum spanning trees instead of minimum spanning trees the list of the edges $E(G)$ is sorted in non-increasing edge weight. The forest of spanning trees is initialized in such a way each element of the forest contains exactly one tree-hexagon.

The expression $\tau(t_i, t_j) = \tau/(\min(|C_{t_i}|, |C_{t_j}|)$ at the line 18 of Algorithm 2 is very important at the beginning of the algorithm because initially the components considered contains only one tree-hexagon and in this case

$$IntVar(C_{t_i}, C_{t_j}) = 0 \ \wedge \ \tau \ (\min(|C_{t_i}|, |C_{t_j}|) = \tau.$$

In order to consider an edge (h_i, h_j) to belonging to the non-boundary class of edges and in consequence to merge the components C_{t_i} and C_{t_j} corresponding to h_i and h_j respectively, it is necessary that $w(h_i, h_j) < \tau$.

When the components grow and both components C_{t_i} and C_{t_j} contain more than one tree-hexagon, the external variation between C_{t_i} and C_{t_j} decreases, and in this case the decision for merging or non-merging C_{t_i} and C_{t_j} is affected more by their size than by the global threshold τ.

For each segmentation S_l determined by Algorithm 2 and for each connected component C of the corresponding spanning graph G_l there is a unique maximum spanning tree, $F_l(C)$, that maximize the sum of edge weights for this component.

Algorithm 2 Color-based segmentation

```
1: procedure createColorPartition(G, H)
2:                                          ▷ G = (V, E), H = {h_1, ..., h_|V|}
3:     τ ← determineThreshold(G)
4:     Bound ← ⟨⟩                                          ▷ Initialize Bound
5:     for all i ← 1, |V| do
6:         makeSet(h_i)                    ▷ Initialize the disjoint set data structures
7:     end for
8:                                                          ▷ At this point l ← 0
9:                                          ▷ and S^0 ← {{h_1}, ..., {h_|V|}}
10:    sort(E, E_π)
11:                                         ▷ E_π = ⟨e_π1, ..., e_π|E|⟩ is the sorting of E
12:                                                  ▷ in order of non-increasing weight
13:    for all k ← 1, |E| do
14:                                      ▷ Let e_πk = (h_i, h_j) be the current edge in E_π
15:        t_i ← findSet(h_i)
16:        t_j ← findSet(h_j)
17:        if t_i ≠ t_j then
18:            if w(h_i, h_j) ≤ intVar(t_i, t_j) + τ(t_i, t_j) then
19:                union(t_i, t_j, w(h_i, h_j))
20:                                                          ▷ l ← l + 1
21:                                  ▷ S^l ← S^(l−1) − {{C_ti}, {C_tj}} ⋃ {C_ti ∪ C_tj}
22:            else
23:                Add the edge (h_i, h_j) the the list Bound
24:                                          ▷ bound(S^l) ← bound(S^(l−1)) ∪ {(h_i, h_j)}
25:            end if
26:        else
27:                                                  ▷ Do nothing, t_i ∈ C_tj
28:        end if
29:    end for
30: end procedure
```

The forest of all maximum spanning trees associated to the segmentation S^l is

$$Fl = \bigcup_{C \in S^l} Fl(C), \tag{20}$$

and algorithm makes greedy decisions about which edges to add to F_l.

Every time when an edge is added to the maximum spanning tree a union of the two partial spanning trees containing the two vertices of the edge is made. In this way the sequence of the edges contained in the forest F_l of spanning trees is implicit determined at the line 13 of Algorithm 2.

Conversely for each spatial tree T from the forest F_l, the set of all vertices of the initial graph contained in the tree T is denoted by $Set(T)$ and it represents the connected component of S_l associated to maximum spanning tree T:

$$T = F_l(Set(T)). \tag{21}$$

The functions *MAKESET*, *FINDSET* and *UNION* used by the segmentation algorithm implement the classical *MAKESET*, *FIND−SET* and *UNION* operations for

disjoint set data structures with union by rank and path compression [19]. In addition the function call, *UNION*(t_i, t_j, $w(h_i, h_j)$), performs the following operation, assuming that t_i is the root of the new spanning tree resulted by combining the spanning trees represented by t_i and t_j:

- determining $CC[t_i].surface$ as the concatenation of the lists $CC[t_i].surface$ and $CC[t_j].surface$,
- determining $CC[t_i].frontier$ as a list of indices of tree-hexagons belonging to the frontier of the new component $\{C_{t_i} \cup C_{t_j}\}$,
- determining $CC[t_i].color$ as the value $(n_i c_i + n_j c_j)/(n_i + n_j)$, where $c_i = CC[t_i].color$, and n_i represents the number of elements in the tree $CC[ti]$.

Let n be of the input the number of the vertices of the input spatial graph $G = (V, E)$ of the color-based volumetric segmentation algorithm, $n = |V|$.

The computational complexity of the color-based segmentation algorithm is given by $T(CREATECOLORPARTITION) = O(n * \log(n))$.

5 Syntactic-Based Volumetric Segmentation Algorithm

Let $G = (V, E)$ be the undirected spatial graph constructed on the tree-hexagonal structure of a volumetric image. The global parameter threshold is determinate by using Algorithm 1. In order to determine a good final segmentation and to discover the objects from the input image, the syntactic based sequence of volumetric segmentations, Sf, as defined by Eq. 10, can decomposed into several subsequences, each subsequence being determined by a modified form of the Boruvka's algorithm.

Let $i_1 < i_2 < \cdots < i_x < i_{x+1}$ be a sequence of indices, with $i_1 = t$ and $i_{x+1} = k$, that allows a decomposition of the sequence S_f as follows:

$$\begin{aligned} S_f = \langle & S^{i_1}, S^{i_1+1}, \ldots, S^{i_2-1}, S^{i_2}, \\ & S^{i_2+1}, S^{i_2+2}, \ldots, S^{i_3}, \\ & \cdots \\ & S^{i_x+1}, S^{i_x+2}, \ldots, S^{i_{x+1}} \rangle. \end{aligned} \tag{22}$$

As presented in Algorithm 3 the procedure *CREATESYNTACTICPARTITION* implements the syntactic based volumetric segmentation, while the function *GENERATEPARTITION* is used to generate the subsequences of segmentations, $S_{f_1}, \ldots, S_{f_x}$, each subsequence of the form,

$$S_{f_j} = \langle S^{i_j}, S^{i_j+1}, \ldots, S^{i_{j+1}-1}, S^{i_{j+1}} \rangle, \tag{23}$$

being determined by the function *GENERATEPARTITION* at the *jth* call. The last segmentation of the subsequence S_{f_j} generate by *GENERATEPARTITION* is also the input sequence of the $(j + 1)th$ call of *GENERATEPARTITION*. The first input

segmentation S^{i_1} is the final segmentation S^t of the color based segmentation algorithm. The function *DETERMINEWEIGHTS* determines the set A of weights.

Algorithm 3 Syntactic-based segmentation

```
1: procedure createSyntacticPartition(G, G', th_g^k)
2:     Input G, G', th_g^k
3:     Output G'
4:     A ← determineWeights(G')
5:     count ← 0
6:     repeat
7:         G' ← generatePartition(G, G', th_g^k, newPart)
8:         if newPart then
9:             count ← 0
10:            k ← [a_0 a_0 a_0 a_0]^T
11:        end if
12:        th_g^k ← modifyWeights(G', k)
13:        count ← count + 1
14:        nextKVector(k)
15:    until count = |A|^4
16: end procedure
```

More formally, the *jth* call of the function *GENERATEPARTITION*, for which the output parameter *newPart* has the value true, is associated to the non-empty subsequence S_{f_j} of volumetric segmentations and it generates a sequence of graphs,

$$G^{i_j} = \langle G^{i_j}_{i_j}, G^{i_j}_{i_j+1}, \ldots, G^{i_j}_{i_{j+1}-1}, G^{i_j}_{i_{j+1}} \rangle, \tag{24}$$

and a sequence of associated forests of minimum spanning trees,

$$F^{i_j} = \langle F^{i_j}_{i_j}, F^{i_j}_{i_j+1}, \ldots, F^{i_j}_{i_{j+1}-1}, F^{i_j}_{i_{j+1}} \rangle, \tag{25}$$

such that the last forest is empty, $F^{i_j}_{i_{j+1}} = \emptyset$. For each graph $G^{i_j}_l$ from the sequence G^{i_j}, $F^{i_j}_l$ represents the forest of minimum spanning trees of $G^{i_j}_l$, and $G^{i_j}_{l+1}$ is the contraction of $G^{i_j}_l$ over all the edges that appear in $F^{i_j}_l$, as presented in Algorithm 3.

Because the last graph, $G^{i_j}_{i_{j+1}}$, of the sequence G^{i_j} cannot be further contracted the dissimilarity vectors of functions associated to the edge weights, $d(C(v_i), C(v_j))$, are not modified, and thus the edge weights, $w(v_i, v_j)$, as defined by the function *GRAPHEXTRACTION* are not modified. In order to restart the process for determining the new subsequence,

$$S_{f_{j+1}} = \langle S^{i_{j+1}}, S^{i_{j+1}+1}, \ldots, S^{i_{j+2}} \rangle, \tag{26}$$

the first graph, $G^{i_{j+1}}_{i_{j+1}}$ of the sequence $G^{i_{j+1}}$ differs from the last graph, $G^{i_j}_{i_{j+1}}$, of the sequence G^{i_j} by modifying only the weighted vector $k \in \mathbb{K}$. The function *MODIFYWEIGHTS* of Algorithm 3 realizes this modification and recalculates the new global weighted threshold. In this case the values for the weighted vector k

are sequential determined in the lexicographic order, generated by the procedure *NEXTKVECTOR*.

The function *MODIFYWEIGHTS* realizes this modification and recalculates the new global weighted threshold. In this case the values for the weighted vector k are sequential determined in the lexicographic order, generated by the procedure *NEXTKVECTOR*.

This constraint is necessary in order to realize a stopping criterion for the algorithm: the last graph cannot be modified and for all distinct values of the weighted vectors $k \in \mathbb{K}$ and thus another partition cannot be determined. Each time when *GENERATEPARTITION* generates a non-empty sequence of segmentations, the output parameter *newPart* became true and the first vector of the set $\mathbb{K}$ is generated.

When *GENERATEPARTITION* generates an empty sequence of segmentations, *newPart* is false and the next vector in lexicographic order is generated by the procedure *NEXTKVECTOR*.

When sequentially for all distinct weighted vectors $k \in \mathbb{K}$ (e.g. $|A|^4$ distinct vectors, with the set A specified by the Relation 23) generated in lexicographic order the function *GENERATEPARTITION* generates a empty sequence of segmentations, the procedure *CREATESYNTACTICPARTITION* finishes.

Between the last graph, $G^{i_j}_{i_{j+1}}$, of the sequence G^{i_j} and the first graph, $G^{i_{j+1}}_{i_{j+1}}$ of the sequence $G^{i_{j+1}}$, there is a sequence of graphs that differ only by the edge weights,

$$\widehat{G}^{i_j} = \langle \widehat{G}^{i_j}_1, \widehat{G}^{i_j}_2, \ldots, \widehat{G}^{i_j}_{\widehat{n}^i_j} \rangle, \tag{27}$$

such that $\widehat{G}^{i_j}_1 = G^{i_j}_{i_j}$ and $\widehat{G}^{i_j}_{\widehat{n}^i_j} = G^{i_{j+1}}_{i_{j+1}}$. This sequence is obtained when the function *GENERATEPARTITION* generates an empty sequence of segmentations, with $\widehat{n}^i_j \leq |A|^4$.

6 Computational Complexity Analysis of the Color-Based Spatial Segmentation Algorithm

Let $m = |E|$ be the number of the tree-edges of the input spatial graph $G = (V, E)$ of the color-based algorithm, and $n = |V|$ the number of the vertices of G. The running time of the color-based spatial segmentation Algorithm 2 can be factored into four parts:

- The running time required to determinate the threshold τ, denoted by t_0 (line 4), where $t_0 = O(m)$ from relation

$$T(CREATEHEXAGONALSTRUCTURE) = O(n),$$

because $O(n) = O(n_p)$ (the assertion that the number of the resulted tree-hexagons is always less than $np/8$)

- The running time required to initialize the array *CC* at the lines 4–6, denoted by t_1,

$$t_1 = O(n). \tag{28}$$

- The running time required to sort the edges into non-increasing order of weights at the line 9, denoted by t_2.
- The running time of the main part of the algorithm at the lines 12–27, denoted by t_3.

Because $m \leq 3n - 6$ it follows that $O(m) = O(n)$, and thus the running time t_0 is

$$t_0 = O(n). \tag{29}$$

The running time required to sort the edges into non-increasing order of weights can be done in $O(m \log m)$ by using one of several sorting methods (e.g., the Quicksort method). It follows that $O(m \log m) = O(n \log n)$, and thus the running time t_2 is

$$t_2 = O(n \log n). \tag{30}$$

In the following we will discuss the running time t_3. The running time of the function *UNION* at the line 18 can be also factored into two parts:

- the running time for the operations concerning disjoint-set data structures, denoted by t_3^s,
- the running time of the additional operations for determining the values for the fields of the *Tree* objects when merging two components, denoted by $t_3{}^l$.

As a consequence the running time t_3 can be written as

$$t_3 = t_3^s + t_3^l, \tag{31}$$

where t_3^s is the part of t_3 by considering only the operations for disjoint-set data structures in the union function, and t_3^l is the part of t_3 by considering only the additional operations in *UNION*.

Because the function *FINDSET* performs standard operations on disjoint-set data structures and the operation at the line 17 is done in constant time it follows that

$$t_3^s = O(m * \alpha(n)), \tag{32}$$

where $\alpha(n)$ is a very slowly growing function, the inverse of the extremely quickly-growing Ackermann function $A(n, n)$ [19]. Because we have $m = 3n - 6$, and because

$$a(n, n) = O(\log^* n), \tag{33}$$

where

$$\log^* n = \min_{i \geq 0}(\log^{(i)} n, 1). \tag{34}$$

it follows that

$$t_3^s = O(n \log^* n). \tag{35}$$

The running time t_3^l for determining the values for the fields of the *Tree* objects when merging two components is factored as follows:

- the running time for determining the values for the fields *isRoot*, *parent*, *compIndex*, *surface*, and *color*, denoted by t_c, is $t_c = O(m)$, because at each iteration determining these values can be done in constant time,
- the running time for determining the value of the field frontier, denoted by t_f.

In order to determine t_f, let $sp(C^{'})$ and $sp(C^{''})$ be the two lists of tree-hexagons belonging to the frontier of the two components, $C^{'}$ and $C^{''}$, that are merged by the union function, and let $t_f(C)$ be the running time for determining the frontier of the merged component, C. Determining the value of the field frontier associated to the merged component require the traversal of the shortest list from the pair of lists $sp(C^{'})$ and $sp(C^{''})$. Because for every component C the number of the tree-hexagons contained in the region associated to C is less than the number of tree-hexagons from its frontier, the running time $t_f(C)$ verify the following condition:

$$t_f(C) = |C|/2, \tag{36}$$

where $|C|$ represents the number of the tree-hexagons contained in the region associated to C. For the sake of simplicity we assume that $n = 2^k$ for a some integer k. In the worst case the final segmentation S^t contains only one component, $S^t = \langle C^t \rangle$, with $|C^t| = n$, and at each merge operation, the two merged components have the same frontier length

$$\min(|sp(C')|, |sp(C'')|) = |sp(C')| = |sp(C'')|. \tag{37}$$

Thus the worst scenario is in the case when all pairs of merged components have the same frontier length and the same area: first are merged all components containing one hexagon, then are merged all components containing two tree-hexagons, etc. It follows that the running time for determining all the values frontier verify the following relation:

$$t_f = \frac{n}{2} + 2\frac{n}{2^2} + 2^2\frac{n}{2^3} + \cdots + 2^{k-1}\frac{n}{2^k}, \tag{38}$$

where for each term, $2^{i-1}\frac{n}{2^i}$, the factor $\frac{n}{2^i}$ represents the number of the tree-hexagons associated to a component, and 2^{i-1} represents the number of components with the same area. Because

$$\frac{n}{2} + 2\frac{n}{2^2} + 2^2\frac{n}{2^3} + \cdots + 2^{k-1}\frac{n}{2^k} = k\frac{n}{2} = \frac{n \log n}{2 \log 2}$$

it follows that $t_f = \frac{n \log n}{2 \log 2}$, and, in conclusion, $t_f = O(n \log n)$.

Because G is a spatial graph and $m \leq 3n - 6$ it follows that $t_c = O(m) = O(n)$ and thus the running time t_3^l is determined as

$$t_3^l = O(n \log n), \tag{39}$$

and from the relations 31, 35 and 39 it follows that

$$t_3 = O(n \log n). \tag{40}$$

Finally from the relations 28, 30 and 40 it follows the overall running time of Algorithm 2 is

$$T(CREATECOLORPARTITION) = O(n \log n). \tag{41}$$

7 Conclusions

Image segmentation plays a crucial role in effective understanding of digital images, planar or volumetric images. Past few decades saw hundreds of research contributions in this field. However, the research on the existence of general purpose segmentation algorithm that suits for variety of applications is still very much active. Among the many approaches in performing image segmentation, graph based approach is gaining popularity primarily due to its ability in reflecting global image properties. The current research in graph based methods orients towards producing approximate solution (or sub-optimal solution) for such graph matching problem to reduce processing time. Also, use of a priori information that include shape, topology and appearance model of the category of images to be segmented is getting more popularity [21].

The problems of volumetric image segmentation and grouping remain great challenges for computer vision. The problem of all segmentation methods is a well-studied one in literature and there are a wide variety of approaches that are used [6]. Different approaches are suited to different types of input images and the quality of output of a particular algorithm is difficult to measure quantitatively due to the fact that there may be many 'correct' segmentation method for a single image [13]. We plan to use a larger image database to confirm the quality of the obtained results, and do the evaluation with additional low level cues as well as different statistical measures.

Here, a graph-based theoretic framework is considered by modeling image segmentation as a graph partitioning and optimization problem using input spatial graph.

We are introducing new algorithm for volumetric segmentation based on Virtual Tree-Hexagonal Structure constructed on the image voxels [22, 23]. We have presented the original and efficient algorithm of volumetric segmentation methods and honeycomb cells used is the first run in volumetric segmentation algorithm. Then we can use the graph facilities and their related algorithms and computational complexity can be viewed as slow as the fundamental graph algorithms. The key to the whole algorithms of volumetric segmentation method is the honeycomb cells.

The major concept used in graph-based volumetric segmentation method is the concept of homogeneity of volumes and thus the edge weights are based on color distance. Our original algorithms for Color-based Segmentation and Syntactic-based Segmentation are linear. The proposed volumetric graph-based segmentation method is divided into two different steps: (a) a segmentation step that produces a maximum spatial spanning tree of the connected components of the tree-grid spatial graph constructed on the tree-hexagonal structure of the volumetric input image, and (b) the final volumetric segmentation step that produces a minimum spatial spanning tree of the connected components, representing the visual objects, by using dynamic weights based on the geometric features of the volumes.

Then the paper describes the Computational Complexity Analysis of the Color-Based Spatial Segmentation Algorithm.

Enhancement and generalization of this method is possible in several further directions. First, it could be modified to handle open curves for the purpose of medical diagnosis. Second, research direction is the using of composed shape indexing for both semantic and geometric image reasoning. Incorporation of the fuzzy set theory into graph based frameworks can achieve enhanced segmentation performances.

References

1. Janakiraman, T., Mouli, P.C.: Image segmentation using euler graphs. Int. J. Comput. Commun. Control **5**(3), 314–324 (2010)
2. Felzenszwalb, P., Huttenlocher, W.: Efficient graph-based image segmentation. Int. J. Comput. Vision **59**(2), 167–181 (2004)
3. Guigues, L., Herve, L., Cocquerez, L.P.: The hierarchy of the cocoons of a graph and its application to image segmentation. Pattern Recogn. Lett. **24**(8), 1059–1066 (2003)
4. Gdalyahu, Y., Weinshall, D., Werman, M.: Self-organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization. IEEE Trans. Pattern Anal. Mach. Intell. **3**(10), 1053–1074 (2001)
5. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 885–905 (2000)
6. Camilus, K.S., Govindan, V.: A review on graph based segmentation. Int. J. Image Graph. Sig. Proc. **5**, 1–13 (2012)
7. Jermyn, I., Ishikawa, H.: Globally optimal regions and boundaries as minimum ratio weight cycles. IEEE Trans. Pattern Anal. Mach. Intell. **23**(8), 1075–1088 (2001)
8. Cooper, M.: The tractibility of segmentation and scene analysis. Int. J. Comput. Vision **30**(1), 27–42 (1998)
9. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. Int. J. Comput. Vision **43**(1), 7–27 (2001)
10. Comaniciu, D., Meer, P.: Robust analysis of feature spaces: color image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
11. Brezovan, M., Burdescu, D., Ganea, E., Stanescu, L.: An adaptive method for efficient detection of salient visual object from color images. In: Proceedings of the 20th International Conference on Pattern Recognition, pp. 2345–2349. Istanbul, Turkey (2010)

12. Comaniciu, D., Meer, P.: Mean shift analysis and applications. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1197–1203. Madison, Wisconsin (1999)
13. Powers, D.M.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. J. Mach. Learn. Technol. **2**(1), 37–63 (2011)
14. Burdescu, D., Brezovan, M., Ganea, E., Stanescu, L.: A new method for segmentation of images represented in a HSV color space. In: Proceedings of the Advanced Concepts for Intelligent Vision Systems Conference, pp. 606–617 (2009)
15. Stanescu, L., Burdescu, D., Brezovan, M.: A comparative study of some methods for color medical images segmentation. EURASIP Journal on Advances in Signal Processing, 128 (2011)
16. Stanescu, L., Burdescu, D., Brezovan, M., Mihai, G.: Creating New Medical Ontologies for Image Annotation. Springer, Berlin (2011)
17. Gonzales, R., Wintz, P.: Digital Image Processing. Addison-Wesley, Reading (1987)
18. Middleton, L., Sivaswamy, J.: Hexagonal Image Processing; A Practical Approach. Advances in Pattern Recognition. Springer, Berlin (2005)
19. Cormen, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. MIT Press, Cambridge (1990)
20. Gijsenij, A., Gevers, T., Lucassen, M.P.: A perceptual comparison of distance measures for color constancy algorithms. In: Proceedings of the 10th European Conference on Computer Vision, pp. 208–221. Marseille, France (2008)
21. Sanfeliu, A., Alquézar, R., Andrade, J., Climent, J., Serratosa, F., Verges, J.: Graph-based representations and techniques for image processing and image analysis. Pattern Recogn. **35**(3), 639–650 (2001)
22. Burdescu, D.D., Brezovan, M., Stanescu, L., Spahiu, C.S.: A spatial segmentation method. Int. J. Comput. Sci. Appl. **1**(5), 75–100 (2014)
23. Burdescu, D.D., Stanescu, L., Brezovan, M., Spahiu, C.S.: Computational complexity analysis of the graph extraction algorithm for 3d segmentation. In: Proceedings of the IEEE Tenth World Congress on Services, pp. 462–470. Alaska, USA (2014)

A Hybrid Intelligent System in Cultural Intelligence

Zhao Xin Wu and Li Zhou

Abstract We live in an era of globalization where international activities between different cultures and intercultural communications and exchanges are becoming more common and are taking on much greater importance than ever before. Researches on cultural intelligence supply a new perspective and a promising way to reduce intercultural conflicts or obstacles. To date, no research on cultural intelligence has been empirically computerized. This research aims to invent a cultural intelligence computational model and to implement the model in an expert system in order to process cultural intelligence soft data through the use of hybrid artificial intelligence technology. This intelligent system represents a breakthrough in the cultural intelligence and AI domains. The purpose of this research is to support individuals and organizations in solving the intercultural adaptation problems that they face in various authentic situations.

Keywords Cultural intelligence · Fuzzy logic · Artificial neural networks · Expert system · Hybrid intelligent technologies

1 Introduction

The globalization has increased dramatically. Culture can play a significant role in the success or failure of face-to-face encounters [1], and because of cultural diversity, "Culture is more often a source of conflict than of synergy. Cultural

Z.X. Wu (✉)
Computer Science Department, University of Quebec in Montreal,
PO Box 8888, Downtown, Montreal, QC H3C 3P8, Canada
e-mail: zhao_xin_wu@hotamil.com

L. Zhou
School of Electronic and Control Engineering, Chang'an University, Xi'an,
People's Republic of China
e-mail: 47599053@qq.com

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_10

differences are a nuisance at best and often a disaster" (Dr. Geert Hofstede). Confronted with cultural diversity, some individuals and organizations successfully adapt themselves to a new cultural environment, but others do not. Why is the decisive factor for these completely different results? How a good decision can be made in different cultural environment? What skills can be improved for cultural adaptation? In recent years, cultural intelligence (CQ) has been presented as a new phenomenon to answer these questions in certain ways. However, current studies relative to CQ are currently treated at the manual level. Moreover, cultural knowledge is generally represented by natural language, in ambiguous terms, and it is difficult for traditional computing techniques to cope with these. In such a context, globalization and traditional computing techniques have encountered two major challenges: the first is, for human beings, how to adapt to cultural diversity, and the second is, for computers, the processing of "soft data" and the representation of human-like thinking.

The main focus of this research attempts to give effective solutions for the problems mentioned above. There are three goals of this study: (1) To help individuals and companies in decision-making processes that involve cultural affairs. (2) To improve use a specific form of intelligence based on an individual's capacity to understand, to reason correctly, and to adapt to culturally diversified situations [2]. (3) To facilitate the work of researchers and to better equip them in their CQ studies.

2 Cultural Intelligence and Its Dimensions

Cultural Intelligence has been referred to as the acronym CQ. Earler and Ang [3] present CQ as a reflection of people's ability to collect and process information, to form judgments, and to implement effective measures in order to adapt to a new cultural context. Earley and Mosakowski [4] later redefined is a complementary intelligence form which may explain the capacity to adapt and face diversity, as well as the ability to operate in a new cultural setting. Brisling and Worthley [5] define the CQ as the level of success that people have when adapting to another culture. Thomas and Inkson [6] describes CQ as the capability to interact efficiently with people who are culturally different. Johnson et al. [7] define CQ as the effectiveness of an individual to integrate a set of knowledge, skills and personal qualities so as to work successfully with people from different cultures and countries.

Different researchers have different dimensional structures to measure CQ. Earley and Ang [3] describe the first structure of CQ by three dimensions: cognition, motivation and behavior. While Thomas and Inkson advocate another tridimensional structure. They state that the structure of CQ should be based on the skills required for intercultural communication, that is to say, knowledge, vigilance and behavior [6]. In these three dimensions, vigilance, which is the key to CQ, acts as a bridge connecting knowledge and behavior. Tan [8] believes that CQ has three main

components: (1) cultural strategic thinking; (2) motivational; and (3) behavioral. CQ integrates these three components. Tan stressed the importance of behavior as being essential to CQ. If the actions in the first two parts are not converted into action, CQ is meaningless.

Ang and Van Dyne [9] suggest a CQ structure with four dimensions rather than three. This structure has been widely used in the following cultural researches and studies. The four dimensions of CQ are described as following:

- *Metacognition* refers to the cognitive ability of an individual to recognize and understand appropriate expectations in different cultural situations. It reflects the mental processes that an individual uses to acquire and understand cultural knowledge.
- *Cognition* is a person's knowledge of the standards, practices and conventions in different cultures which he/she acquired from education and personal experiences.
- *Motivation* refers to the motivation of an individual to adapt to different cultural situations. It demonstrates the individual's ability to focus his/her attention and energy on learning and practicing in culturally diverse situations.
- *Behavior* is defined as an individual's ability to communicate and behave with cultural sensitivity when interacting with people of different cultures. It represents a person's ability to act and speak appropriately (i.e., use suitable language, tones, gestures and facial expressions) in a given culture [9].

Although studies of CQ structures have made some progress in the three-dimensional and four-dimensional structures, they are not always conclusive. One of the most potentially contentious issues is whether the structure should or should not include a metacognitive CQ dimension. Moreover, apart from the three and four dimensions identified in the structures, are there any other dimensions or important elements to consider in CQ structures? To answer these questions, there is a need for further theoretical and empirical researches.

3 Cultural Intelligence Computational Model

3.1 Data and Knowledge Acquisition

Kon et al. [10], Ang et al. [2, 11] developed a self-assessment questionnaire which has 20 questions that measure CQ. This questionnaire was used to collect data for studies on the capabilities of the test subjects regarding their cultural adaptation capacity. This questionnaire is generally divided into four sections: metacognitive (four questions), cognitive (six questions), motivational (five questions) and behavioral (five questions). For example, one of the questions from metacognitive section is "I am conscious of the cultural knowledge I use when I interacting with people with different cultural backgrounds." Van Dyne et al. [12] developed a version of the questionnaire from the point of view of an observer. It is also based on

the 20 questions of Ang et al. [2, 11] in order to measure the CQ of individuals. The questionnaire was adapted from each question of the self-assessment questionnaire to reflect the assessment made by an observer rather than the user himself. For example, the question of the questionnaire shown above changes from: "I am conscious of the cultural knowledge I use when …" to "This person is conscious of cultural knowledge he/she uses when …." As explained by Van Dyne et al. [12], these questionnaires allow for the effective assessment of CQ in practical applications. We therefore adapted the self-assessment questionnaire of Ang and Van Dyne [2], along with the observer questionnaire by Van Dyne et al. [12] to measure CQ in order to integrate the evaluation functions offered by our system. Thus, the user can be evaluated and proper recommendations can be offered by the system.

3.2 Applying Hybrid Artificial Intelligent Technology to Computational Model

We used the hybrid neuro-fuzzy technology to design this model. This hybrid technology makes use of the advantages and power of fuzzy logic and Artificial Neural Network (ANN), which are complementary paradigms. (1) fuzzy logic technology is used for three reasons. First, the CQ variables, which are ambiguous and imprecise, such as "this person has low motivation" and "that action is highly risky because of this religion." Second, fuzzy logic is particularly well-suited for modeling human decision-making when dealing with "soft data," which come from common sense, as well as vague and ambiguous terms. Third, fuzzy logic provides a wide range of expressions that can be understood by computers. (2) ANN: Although the fuzzy logic technology has the ability and the means of understanding culturally natural language, it offers no mechanism for automatic rule acquisition and adjustment. The ANN is a good solution for processing incomplete cultural information. The ANN can incorporate new cultural data input with the generalization of acquired knowledge. The hybrid neuro-fuzzy technology which can process CQ "soft data" represents the essence of our computational model.

Modeling is an essential step. Our computational model describes in an abstract way the entity of the system and the problematic to solve in our research in order to understand better them. The model includes a highly detailed plan so as to take into consideration the general layout of the system. The model based on the four dimensional structure of Ang and Van Dyne [9] (see Sect. 2). The model is noteworthy because we use the four CQ dimensions as integrated and interdependent entities. This model represents a comprehensive overview of the various aspects of CQ researches. Our model 'filters' the non-essential details of information. The main three parts of our model are shown in Fig. 1.

- *Input unit* presents information (questionnaires) which expresses the answers of the user via the input of the user interface;
- *Filter and Classifier module* takes the inputted information, classifies it, and filters what is not useful for analysis in the next steps;

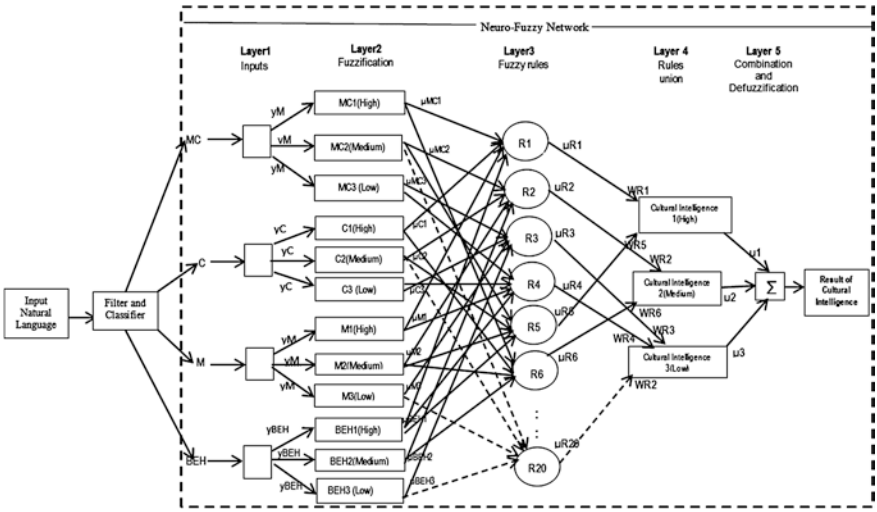


Fig. 1 CQ computational model

- *Neuro-Fuzzy Network* is a neural network with fuzzy inference model capabilities. The system can be trained to develop IF-THEN cultural fuzzy rules and determine membership functions for input and output variables. This unit has four inputs: metacognition (MC), cognition (C), motivation (M) and behavior (BEH), and it has one output: CQ.

Layer 1—*Inputs*: No calculation is made in this layer. Each neuron corresponds to an input variable. These input values are transmitted directly to the next layer.
Layer 2—*Fuzzification*: Each neuron corresponds to a linguistic label (e.g., high, medium and low) associated with one of the input variables in layer 1. In other words, the connection of the output, representing the inclusion value which specifies the degree to which the four input values belong to the neuron's fuzzy set, is calculated in this layer.
Layer 3—*Fuzzy Rule*: The output of a neuron at level 3 is the fuzzy rules of CQ. Each neuron corresponds to one fuzzy rule. The neuron receives as input from the Fuzzification neurons. Neuron R1 represents Rule 1 and receives input from the neurons MC1 (High) and C1 (High). The weights (WR1 to WRn) between layers 3 and 4 are the normalized degree of confidence of the corresponding fuzzy rules. These weights are adjusted when the model is trained.
Layer 4—*Rule Unions* (or consequence): This neuron has two main tasks: (1) to combine the new precedent of rules, and (2) to determine the output level (High, Medium and Low), which belongs to the CQ linguistic variables. For example, μR1, μR5 are the inputs of CQ1 (High), and $\mu 1^{(4)}$ is the output of neuron CQ1 (High).
Layer 5—*Combination and Defuzzification*: This neuron combines all the consequence rules and, lastly, computes the crisp output after Defuzzification. The composition method "sum-product" [13] is used. This method represents a shortcut of

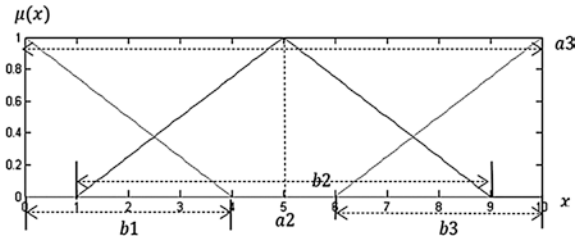


Fig. 2 General CQ Fuzzy sets

the Mamdani-style inference calculation. It computes the outputs of the membership functions defined by the weighted average of their centroids. The calculation formula of weighted average of the centroids of the clipped fuzzy sets CQ3 (Low), 2 (Medium) and 1 (High) are calculated as shown in Fig. 2.

$$y(CQ) = \frac{\frac{1}{3}b_1^2\mu 1 + a_2 b_2 \mu 2 + \left(a_3 - \frac{1}{3}b_3\right)b_3\mu 3}{b_1\mu 1 + b_2\mu 2 + b_3\mu 3} \tag{1}$$

where $a2$ is the center and $a3$ is the end of the triangle. $b1$, $b2$ and $b3$ are the widths of fuzzy sets which correspond with CQ 3, 2 and 1.

4 Supervised Learning

One of the main properties of the model is supervised learning, which has the ability to learn from cultural expert experiences and to improve performance by modifying the CQ rules through learning. Supervised learning involves cultural inputs and cultural outputs that are available to our multilayer neuro-fuzzy network. The task of the network is to predict or adjust inputs to the desired outputs.

This multilayer neuro-fuzzy network can apply standard learning algorithms, such as back-propagation, to train it. The network offers a mechanism for automatic IF-THEN rule acquisition and adjustment. This mechanism is very useful, especially in situations where cultural experts are unable to verbalize the knowledge or problem-solving strategy they use.

The principle of the back-propagation algorithm in supervised learning in our model is that we provide the model with the final external CQ data that supervised learning requires; these data represent the results of a user's CQ evaluation. Each case contains the original input cultural data and the output data offered by CQ human experts to be produced by the model. The model compares actual output with the CQ experts' data during the training process. If the actual output differs from the data given by experts in the training case, the model weights are modified. Figure 3 shows two parts (metacognitive and cognitive dimensions) of the Fig. 1 with three layers (*input layer*, *hidden layer* and *output layer*) as an example to illustrate how the neuro-fuzzy network learns by applying the back-propagation

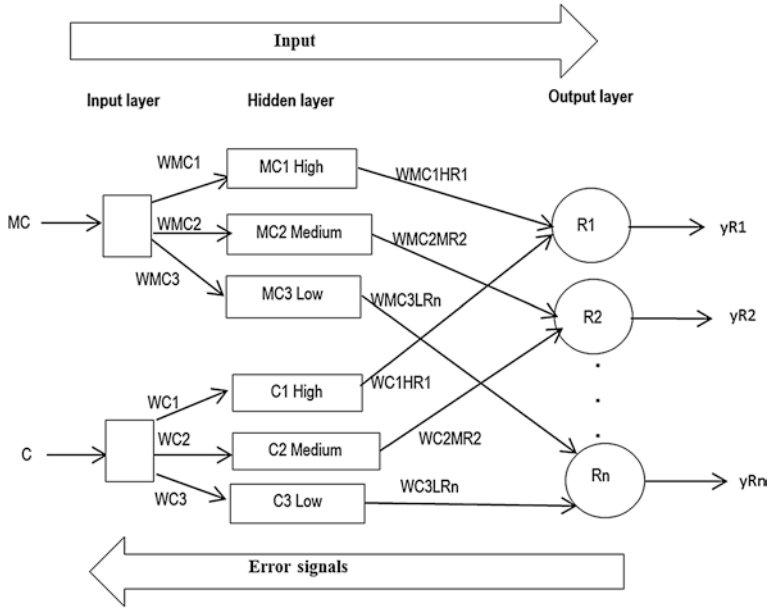


Fig. 3 Back-propagation in CQ computational model learning

algorithm. *MC* and *C* refer to neurons in the input layer; *MC1/C1 High*, *MC2/C2 Medium* and *MC3/C3 Low* refer to neurons in the hidden layer; and *R1, R2* and *Rn* refer to neurons in the output layer.

We explain our model's learning process theory in three steps as follows:

Step 1 *Input Signals*: we input signals from *MC* to *C* into the model; these signals are propagated through the neuro-fuzzy network from left to right, while the difference signals (or error signals) are propagated from right to left.

Step 2 *Weights Training*: to propagate difference signals, we start at the output layer and work backward to the hidden layer. The difference signal at the output of neuron R1 at sequence s is calculated as follows:

$$\mathrm{D_{R1}(s)} = \mathrm{y_{e,R1}(s)} - \mathrm{y_{R1}(s)} \tag{2}$$

where $y_{e,R1}(s)$ is the cultural experts' desired output data of neuron *R1* at iteration *S*. $D_{R1}(s)$ is the difference between the output $y_{R1}(s)$ and the experts' desired output data at iteration s. For example, we use a forward procedure method to update the CQ rules' weight W_{MC1HR1}(MC1 High) Rule *R1* for updating weight at the output layer at iteration *S* is defined as:

$$\mathrm{W_{MC1HR1}(s+1)} = \mathrm{W_{MC1HR1}(s)} + \Delta \mathrm{W_{MC1HR1}(s)} \tag{3}$$

Step 3 **Iteration**: We increase iteration *S* by one and repeat the process until the pre-set difference criterion is satisfied.

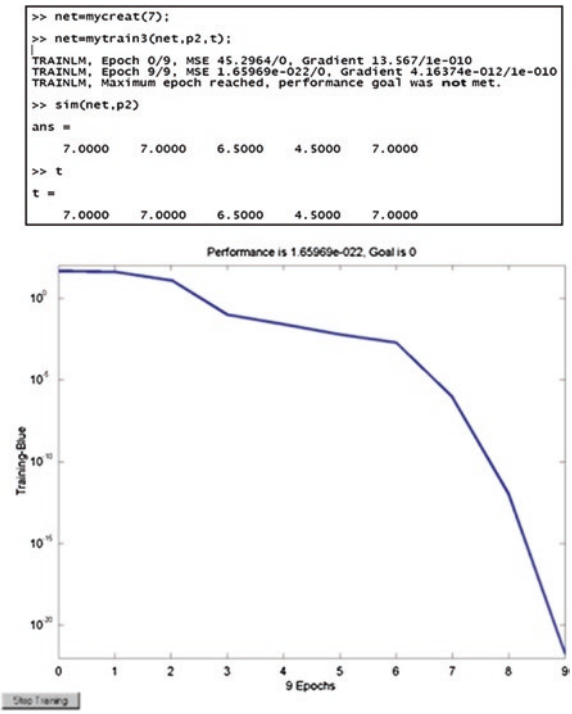


Fig. 4 Learning result in the computational model

Following the above three-step learning procedure, we give a concrete example to demonstrate how the model obtains the desired value after learning, shown in Fig. 4. Suppose we have collected five people's answers as input data, and get five corresponding CQ evaluation results from the output of the model as: y = [5, 6, 7, 3, 2]. For any reason, the cultural experts gave five desired CQ output values as: yd = [7, 7, 6.5, 4.5, 7]. We then used these five pairs of input data and the desired values to train the model. After nine epoch training processes, our new output from the model was: y = [7, 7, 6.5, 4.5, 7].

The model's output quite accurately resembles the desired CQ values from the cultural experts, that is to say, the model has the ability to learn new CQ knowledge.

5 Implementing the Model in an Intelligent System

We would like the system, first, to be capable of acquiring, extracting and analyzing the new CQ knowledge of experts, and second, to serve as an efficient team comprised of top CQ experts, able to provide both recommendations and explanations to users whenever required in culturally diverse settings. Hence, we implemented the computational model in an expert system, called Cultural Intelligence

Evaluation Expert System (CQEES). Figure 5 shows the structure of the CQEES. The CQEES structure includes four main modules:

- *The CQ Computational Model* contains CQ knowledge that is useful for solving CQ problems. The soft-computing technology used in this model enables the system to reason and learn in an uncertain and imprecise CQ setting. It supports all the evaluation steps in the system. This module connects with the *Training Data Database*. The *Training Data Database* are sets of training examples used for training the neuro-fuzzy network during the learning phase.
- *The Cultural Intelligence Rules* examine the CQ knowledge base, which is represented by the trained network, and produce rules which are implicitly built into and incorporated in the network.

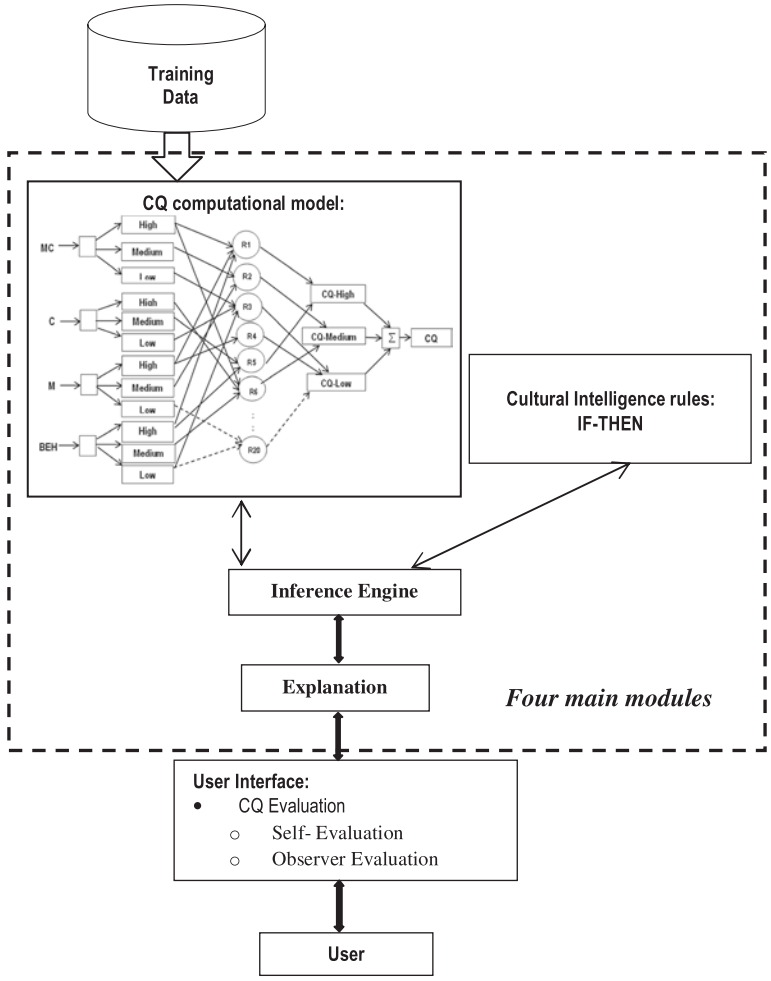


Fig. 5 Structure of CQEES

- *The Inference Engine* controls the flow of information in the system and initiates inference reasoning from the computational model. It also concludes when the system has reached a solution.
- *The Explanation module* explains to the user why and how the CQEES reached the specific CQ evaluation results. These explanations include the conclusion, advice and other facts required for deep reasoning.

The computational model and CQEES are validated and confirmed by evaluations conducted by several cultural experts. The experts simulated some real world problems. These validations ultimately reflect the consistency between the real world and the artificial intelligent system. Based on the results of the validation, users can get two evaluations (self- and observer evaluations) using the 20-item questionnaires (see the interface of the system prototype in Fig. 6).

The CQ evaluation process in the CQEES, first of all, receives the input data from the 20 items of the questionnaire. The system then analyzes and treats these data specifically by applying the strategies of CQ human experts. At the end, the system gives the result of the CQ evaluation and provides suggestions for users who want to follow the CQ training.

Figure 7 illustrates the CQEES as a black box where the input data corresponds to the answers to the 20 items. The output is the evaluation result with explanations to the users.

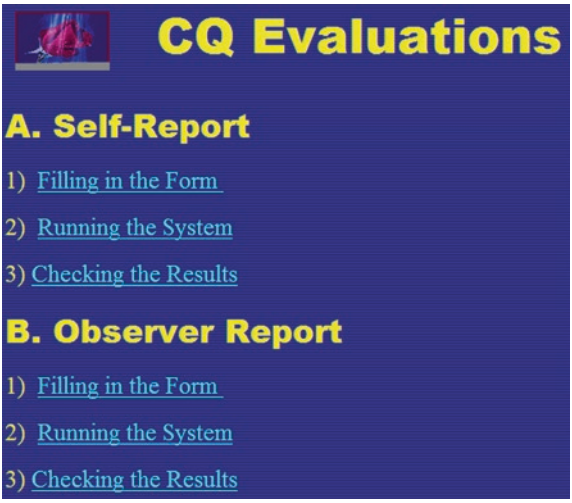


Fig. 6 Interface of CQEES prototype

Fig. 7 Input and output of CQEES

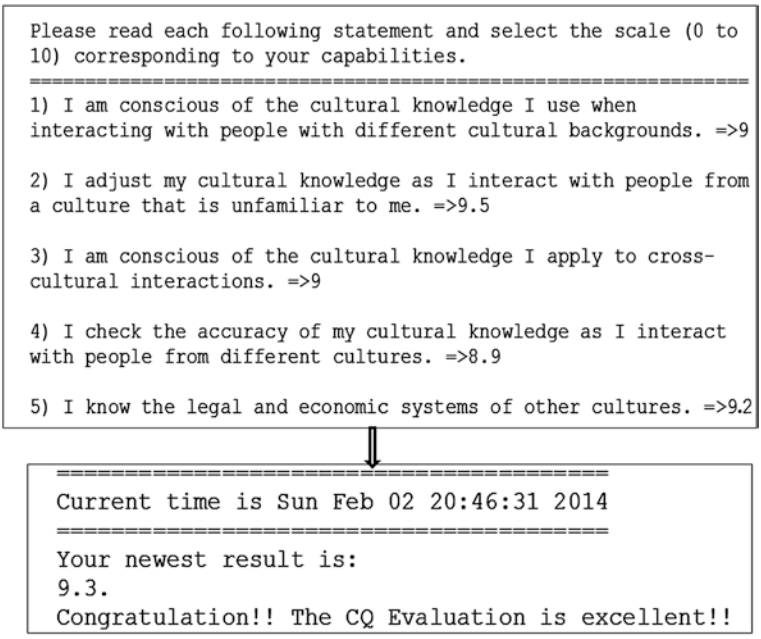


Fig. 8 Self-evaluation result in CQEES (scores higher than 8)

For example, two different results of the self-evaluation questionnaire that evaluate the user's CQ are presented in the CQEES as follows:

Result 1: After inputting the answers to the 20 items in the CQEES, the system provides the feedback. If a user's evaluation achieves a high score (e.g.: more than 8), the system shows the following message in Fig. 8

Result 2: When the evaluation results are below 6, the system accordingly gives useful suggestions for personal self-development as required. This process permits the system to evaluate users so as to identify their problems in the CQ domain and then offers several precise recommendations to users based on the results of the evaluation. Moreover, the system uses natural language to give users recommendations in order to provide them with a stress-free and friendly evaluation. The CQEES presents some recommendations in Fig. 9

The evaluation result shows that the CQEES allows for improved interactions and for more effective aid to users. The evaluation result clarifies and defines the exact problem of concern to the users; indeed, the CQEES could be used in self-awareness training programs. The system provides important insights on personal capabilities, as well as information on the user's own CQ in situations where cultural diversity is of primary importance. This point is particular importance in modern learning theories. Organizations could also use the CQEES (both self- and observer evaluations) to evaluate and train employees so that the latter may function more effectively in such situations.

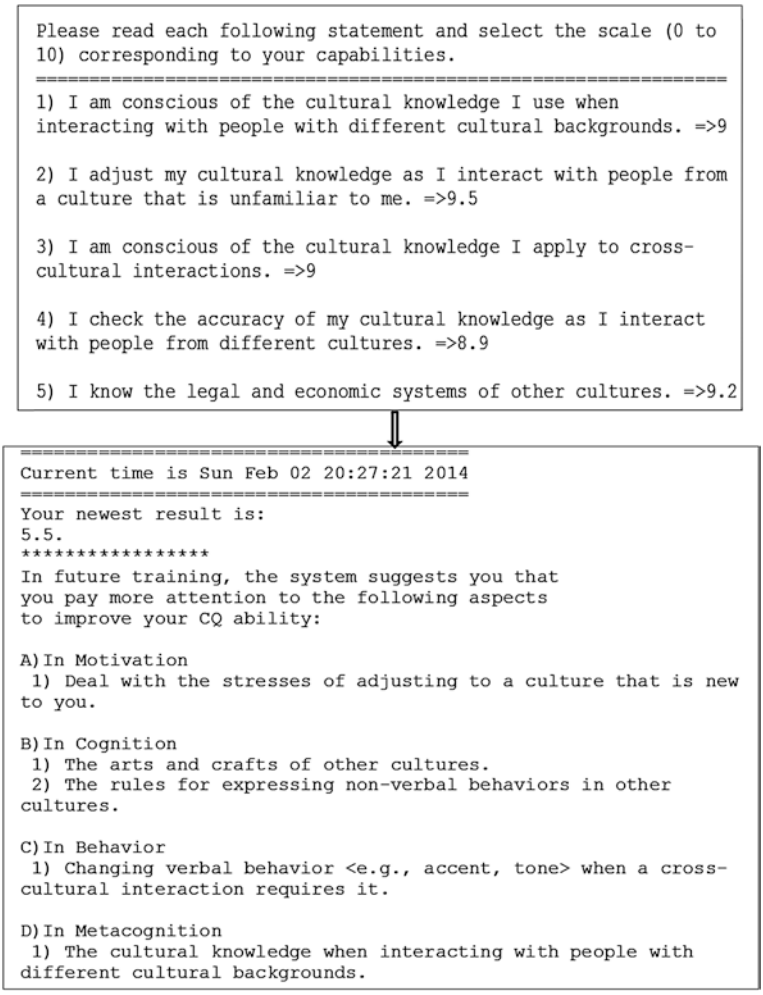


Fig. 9 Self-evaluation result in CQEES (scores lower than 6)

The CQEES serves as an efficient team comprised of top CQ experts who work continuously with individuals and organizations that wish to have an evaluation or insights on how to improve their effectiveness in culturally diverse settings.

6 Conclusion

CQ is defined as the capacity to function effectively in cultural diversity. The achievement of this research is noteworthy because in the CQ domain, this study effectively deals with linguistic variables, soft data and human decision making

based on a hybrid neuro-fuzzy technology, and it possesses parallel computation and the learning abilities of neural networks. From a practical perspective, first, the system is able to evaluate trainees and provide them specific recommendations. It is also able to dynamically adapt to the CQ capacity of trainees. Second, this system is open in the sense that it can provide a standard interface that can facilitate further development. Third, the CQEES is also extensible, both in terms of the system concept model and the system implementation. Fourth, this system has the potential to work as a training extension agent in order to integrate it into another existing intelligent system. Fifth, due to its powerfully designed functions, this system is very easy to extend to other application domains, such as Expatriation and Business Activities [14]. As a result of its high CQ capabilities, the system can not only use its knowledge to train people, but also to work as a CQ decision-making support system to help individuals and organizations take cultural decisions in cross cultural activities.

The contribution of our research, first, fills that gap between CQ and AI. Second, it improves the application of CQ theories in the cognitive domain. The research focuses on modeling four CQ dimensions as an integrated and interdependent body. As a result, the theories should be more complete, more efficient, and more precise in their applications. Third, we have made progress in the domain of AI by computerizing CQ. As a result, new research topics and directions have arisen, and the range of computational intelligence possibilities has been expanded. Fourth, our research is groundbreaking as it simplifies the work of the researchers by freeing them of heavy, complex, repetitive tasks, normally carried out manually in the process of CQ studies.

References

1. Lane, H.C., Hays, M.J.: Getting down to business: teaching cross-cultural social interaction skills in a serious game. Culturally-Aware Tutoring Systems, ITS 2008, Montreal, 23–27 June 2008
2. Ang, S., Van Dyne, L.: Handbook of Cultural Intelligence, 1st edn. M.E. Sharpe, Armonk (2010)
3. Earler, P.C., Ang, S.: Cultural Intelligence: Individual Interactions Across Cultures. Stanford University Press, Stanford (2003)
4. Earley, P.C., Mosakowski, E.: Cultural intelligence. Harv. Bus. Rev. **82**(10), 139–146 (2004)
5. Brisling, R., Worthley, R.M.: Cultural intelligence: understanding behaviors that serve people's goals. Group Org. Manage. **31**(1), 40–55 (2006)
6. Thomas, D.C., Inkson, K.: Cultural intelligence people skills for a global workforce. Consult. Manage. **16**(1), 5–9 (2005)
7. Johnson, J.P., Lenartowicz, T., Apud, S.: Cross-cultural competence in international business: toward a definition and a model. J. Int. Bus. Stud. **37**(4): 525–543 (2006)
8. Tan, J.S.: Cultural intelligence and the global economy. Leadersh. Action **24**(5), 19–21 (2004)
9. Ang, S., Van Dyne, L.: Conceptualization of cultural intelligence. In: Handbook on Cultural Intelligence: Theory, Measurement and Applications. Chapter I, pp. 1–15. M.E. Sharpe, Armonk (2008)
10. Kon, C., Damien, J., Ang, S.: Cultural Intelligence and the Global Information Technology Workforce. NanYang Technological University, Singapore (2010)

11. Ang, S., Van Dyne, L., Koh, S.K.: Personality correlates of the four-factor model of cultural intelligence. Group Org. Manage. **31**, 100–123 (2006)
12. Van Dyne, L., Ang S., Koh, C.: Development and validation of the CQS: The cultural intelligence scale. In: Handbook of Cultural Intelligence, 1st edn. M.E. Sharpe, Armonk (2008)
13. Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall, Englewood Cliffs (1997)
14. Wu, Z.X., Nkambou, R., Bourdeau, J.: Cultural intelligence decision support system for business activities. In: 2nd International Conference on Business Intelligence and Technology, BUSTECH 2012, Nice (2012)

Semantic-Based Recommender System with Human Feeling Relevance Measure

David Werner, Thomas Hassan, Aurelie Bertaux, Christophe Cruz and Nuno Silva

Abstract This work presents a recommender system of economic news articles. Its objectives are threefold: (i) managing the vocabulary of the economic news domain to improve the system based on the seamlessly intervention of the documentalist (ii) automatically multi-classify the economic new articles and users profiles based on the domain vocabulary, and (iii) recommend the articles by comparing the multi-classification of the articles and profiles of the users. While several solutions exist to recommend news, multi-classify document and compare representations of items and profiles. They are not automatically adaptable to provide a mutual answer to previous points. Even more, existing approaches lacks substantial correlation with the human and in particular with the documentalist perspective.

1 Introduction

The decision-making process in the economic field requires the centralization and intakes of a large amount of information. The aim is to keep abreast with current market trends. Thus, contractors, businessmen and salespersons need

D. Werner (✉) · T. Hassan · A. Bertaux · C. Cruz
Université de Bougogne, LE2I, CNRS, Dijon, France
e-mail: david.werner@u-bourgogne.fr

T. Hassan
e-mail: thomas.hassan@u-bourgogne.fr

A. Bertaux
e-mail: aurelie.bertaux@u-bourgogne.fr

C. Cruz
e-mail: christophe.cruzg@u-bourgogne.fr

N. Silva
GECAD and School of Engineering, Polytechnic of Porto, Porto, Portugal
e-mail: nps@isep.ipp.pt

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_11

to continuously be aware of the market conditions. This means to be up-to-date regarding ongoing information and projects undergoing development. With the help of economic monitoring, prospects can be easily identified, so as to establish new contracts. Our tool is specialized in the production and distribution of press reviews about French regional economic actors. The overload of news information is a particular case of information overload, which is a well-known problem, studied by Information Retrieval and Recommender Systems research fields. News recommender systems already exist [1], Athena [2], GroupLens [3] or News Dude [4]. Some of these systems use domain knowledge to improve the recommendation task [1, 2]. To achieve this goal, a content-based recommender system is being developed. A recommender system is necessary for the item ranking and a content-based approach is required to analyze the content of each article to structure and preserve information content. The results of the analysis enable linking the domain knowledge to the articles to improve the recommendation task [1, 2]. Content-based recommender systems typically follow a two-step process:

(i) the indexing of articles and users (also known as *profiling*), which allows to describe them.
(ii) the comparison process which consists in comparing the description of the articles and the profiles of the users. The latter computes the article relevance with regards to the user profile.

In order to capture the economical context, we are moving towards a customized review for each user and towards an opinion survey on magazine readers that cover a broad array of subjects, including news services.

As consequence of this effort, the complete production process of the review is redesigned to produce and to automatically distribute a customized review for each user. So, the aim of the overall system is to manage all news articles produced, and provide the most relevant article for each customer.

In this paper we focus on:

(i) Content-based recommender system, and on the indexing subtask of textual items, based on a controlled vocabularies. Ontologies are central to our proposal, as they are used to represent and manage the controlled vocabularies, to describe profiles and articles, and finally to automatically multi-classify them via inference process. Also, we adopt a machine learning approach for generating a prediction model for supporting the automatic classification. This paper presents a proposal for enriching the documentalist-oriented ontology with the model prediction rules, which provides the necessary capabilities to the DL reasoner for automatic multi-classification.
(ii) Moreover, we are interested in the distinction between the *relevance* and the *similarity*, two notions that are often mixed up. We propose a new measure, *Relevance-Measure*, that allows us to capture the relevance of an article for a given user profile, based on their ontological descriptions.

This paper is organized as follows. First we present the background research work and related work. In the second section we present the system architecture followed by, in Sect. 4 the description of our proposal to automate the indexing task and his valuation, in Sect. 5 the indexing task applied to the particular case profiles and in Sect. 6 the description of our proposal to compute relevance of articles according to profiles and his valuation. Finally, we summarize the contributions.

2 State of the Art

The large amount of information on the web, company information systems, digital libraries, selling websites and so on, is a well-known fact. Recommender systems aims at providing for each user the better items according to his/her needs. Items can be websites, news articles, books, video, music, washing machine, etc. In the literature two paradigms are distinguished. First, *Content-Based Recommender Systems* try to recommend items similar to those a given user has liked in the past. Second, *Collaborative Filtering Recommender Systems* identify users whose preferences are similar to those of the given user and recommend items they have liked [5]. Some subtasks should be performed and the first is named the *indexing task*. It is possible to distinguish two cases, the indexing of items, by content analysis, and the indexing of the profiles, via profiles learning (which generally includes a study of the behavior for implicit feedbacks or proposed to the user to giving explicit feedbacks as "*I like*" button). Both can be seen as a multi-classification task. The second task: *comparison*, consists in filtering each item relative to a given profile [6]. The survey of Nagewara Rao [7] proposes a general comparison of the main advantages and drawbacks of each kind of Recommender System (e.g. content-based or collaborative filtering).

Knowledge management for humans and machines handling. As it is presented in the survey about controlled vocabularies from [8], more and more companies use controlled vocabularies in their information system. Several kinds of structure are used to manage vocabularies, i.e. from the lowest to the richest semantic definition: glossary, taxonomy, thesaurus, ontology. A lot of companies plan to use ontologies in their applications [8]. While acquiring, managing and maintaining controlled vocabularies are important yet relatively easy tasks for the documentalist, the ontology approach to model the domain knowledge is hardly accessible for them due to the complexity of the logical structure, but it is easily used by the machine because it is formal.

Indexing by multilabel-classification: In order to make the indexing process automatic, the system has to associate a set of labels from the taxonomic thesaurus to each article and profile. This cannot be done without the two following process: (i) a text analysis process to extract keywords and other features from texts and (ii) a machine learning process to learn from examples. Feature extraction processes

range from simple term extraction process like tf-idf [9] to text-based semantic-aware processes, e.g. term extraction from text based on (i) information retrieval methods [10], or (ii) based on NLP works [11, 12]. It is possible to use several degrees of text processing tools (and preprocessing), to extract noun phrases (i.e. tokenizer, part of speech tagger, handmade patterns and even parsers). This process allows us to interlink a set of terms (i.e. *features*) from the article term extraction process (i.e. *content analysis*) and the set of taxonomic keywords (i.e. *labels*) for the articles indexing task.

Machine learning process has two primary goals: *Prediction* and *Description*. *Prediction* is concerned with using features of previously classified examples (e.g. documents or any other resources that can be analyzed and classified) to predict the unknown classification (i.e. labels). *Description*, on the other hand, focuses on finding human-interpretable patterns that describe the performed classification.

Two main categories of label-classification prediction can be enumerated: the single-label and the multi-label classification. Single-label classification aims to learn a prediction model from a set of examples that are related with a single label from a set of disjoint labels. In multi-label classification instead, the examples are associated with a set of labels [13]. Multi-label classification faced increased attention in the last decade, overcoming the single-label classification previous dominance, but it was only much recently that hierarchical multi-label classification (HMC) approaches received the desired attention [14–16]. Even so, some of the so-called HMC approaches do not follow a strict hierarchical semantics (in the sense of subsumption), but a clustering approach. This is the case of the state of the art "hierarchical" multi-label approach HOMER [17] and that of [18] that uses Predictive Clustering Tree (PCT) framework. However, unlike HOMER, the approach described in [18] is constrained by the taxonomy or Direct Acyclique Graph (DAG) underlying the training and testing datasets. This is also the case of other works, notably in the area of bioinformatics [14, 19].

Both [18, 19] are very interested in the description of the predictions to the user. In [20] the authors propose an iterative and interactive (between AI methods and domain experts) approach to achieve prediction and description (which are usually hard to fulfill), considering domain expert knowledge and feedback. Unlike us however, in [20] the authors do not aim to automatically multi-classify the items but only to improve the ontology, which means that the resulting ontology is not used for automatic classification of items.

In [15] the authors propose an approach to build ontologies using data mining results upon databases. The result is the enrichment of the ontology with new concepts and datatype properties, which is far from the required specification of classes. Our goal instead is to enrich the already existing taxonomic thesaurus with constraints capturing the prediction model allowing the user to perceive the taxonomic thesaurus, rules and the adopted features.

In [21] the authors are concerned with automatically creating an ontology from the text documents without any prior knowledge about their content. For that they

use an iterative and interactive 4-phases process. Unlike [15, 16] that constructs thesaurus from the learning examples, in this project/paper the thesaurus-based taxonomy already exists and should be applied both in the automatic classification and description.

The profiling is a particular case of indexing, most of existing approaches try to nd correlations based on reading time and eventually scrolling [22, 23] states that there is a lack of research literature about other methods. Reading time is generally correlated to the relevance of the resources, but this correlation is highly dependent on the reading condition (test protocols), and more importantly on the task complexity [24, 25]. Assuming it is possible to qualify these two characteristics in an application, it would be possible to determine a reading time threshold from which a document can be relevant. The use of scrolling, eyetracking and implicit methods in general are not sufficient, but these methods seem to improve the precision of the recommendations in conjunction with ratings [22, 23, 26, 27] concludes by saying that if other actions are taken into account, they must be weighted, as saving or printing a document is probably more important than copying a part of it for example.

Comparison and relevance: Vector Space Model (VSM) [28] involves the representation of items to be recommended (e.g. articles), and sometimes the users needs (e.g. queries, profiles) as vectors. This presentation allows the use of different metrics for comparison. In this article we use the Cosine distance similarities. Many recommender systems based on content use them to make comparisons, either between items or between item and profile [1, 2, 4, 29, 30]. This method of linear algebra has two main advantages: it not only provides a non-binary outcome, so for ordering the results of recommender systems, but also allows quick calculations and resistance to scalability.

Furthermore, methods of information retrieval can be used to take into account the knowledge from a knowledge base while using a vector modeling. The proposed approach [31] uses the WordNet lexical knowledge base [32] to improve the management of the heterogeneity of natural language and thus improve the understanding of user needs. The idea is to add information to user queries (*query expansion*). This method increases the *return*[1] with the aim of improving overall system performance. We adapt this method to the way which the domain knowledge in our ontology is modeled.

We found that the concepts of similarity and relevance are usually combined in systems using vector modeling. Determine suitability as a similarity does not take into account the different levels of specificity in the description of the requirement of users. This description is, however, made possible by the use of external knowledge system. We propose an evaluation of the relevance, *Relevance measure* using the concepts of similarity, but taking into account the perception of the relevance by the user.

[1] Number of items correctly considered as relevant to the actual number of relevant articles.

3 The System

Our system is an ontology-driven content-based recommender system (also named semantic-based recommender system). An ontology is created and populated with the help of company experts, in order to model their domain knowledge in a knowledge base. As a classic content-based recommender system, our system is composed of two main tasks (Fig. 1). The first one is *indexing*, to create a representation of each items content, and users needs. In our system the knowledge base is also populated during this task. The second task is *comparison*, to apply a comparison metric between items and profiles representations so as to measure each items degree of relevance for each profile. Items are ranked with the help of the relevance measure, before being provided to the user. These subjects are developed as follows.

3.1 The Knowledge Base

In order to capture the economical context, we move towards a customized review for each user and towards an opinion survey on magazine readers that covers a broad array of subjects, including news services. Criteria for a relevant customization of the review were identified as a result of this survey as well as expert

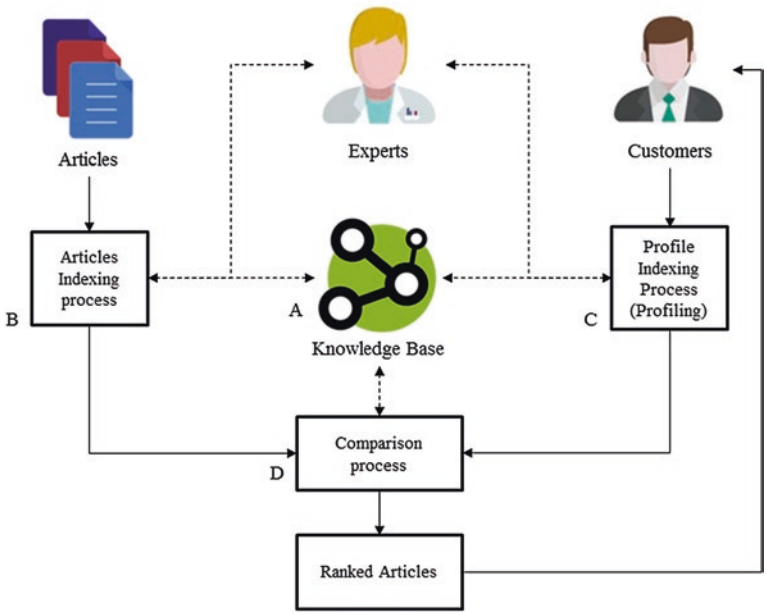


Fig. 1 Our ontology-based recommender system

domain knowledge. These criteria are economic themes (i.e. main economic events), economic sectors, temporal and localization information. Vocabularies associated to these criteria are defined and structured by experts as thesauri. These thesauri are used as facets for the description of the items and profiles. An hybrid domain knowledge representation has been developed, in which the skeleton of the domain knowledge is delivered by an ontology that captures the process and application needs, complemented by a set of thesauri that capture the domain knowledge of the experts. These thesauri has been captured in a very light, expert-oriented fashion, with minimal formal semantics and consistency obligations. Once integrated in the ontology the strict semantics of OWL DL is followed. The relations in thesauri are integrated into the ontology to be used in the recommendation process as input for the evaluation of the semantic distance between articles and profiles. Furthermore, these thesaurus relations are very important for the experts' tasks as allow them to better and faster understand the context of the word/label to use in an articles or profiles indexing.

3.2 Indexing

To achieve the recommendation of articles to customers, the system needs are a representation of the content of each article, and representation of the needs of each customer. The index used in our system is the same for articles and profiles: the knowledge base (Fig. 1a). Articles and profiles are represented by instances in our knowledge base. The ontology contains several roles to model articles' content, and users' interests. To avoid manual classification of articles, which is a time consuming process, we propose to adopt a machine learning approach for generating a prediction model for supporting the automatic indexing. The documentalist-oriented ontology is enriched with the model prediction rules, which provides the necessary capabilities to the DL reasoner for automatic multi-classification. To ensure the same quality as manual indexing, the documentalist who know the economic context, perform the supervision and correction. This is discussed in the Sects. 4 and 5.

3.3 Comparison

The recommendation task is mainly based on the comparison between the user profile and the items available for recommendation. The system uses the knowledge base as an index, and profiles and articles are presented through a set of instances and relations. For each article validated by writers, the full content is inserted in the database and a representation of the article is created in the knowledge base. An instance of the concept article and an instance of each *isAbout* relation are created to link the article to its criteria. For each profile made by the sellers, the associated

representation is created in the knowledge base. An instance of the concept profile is created, and each *isInterestedIn* relation between the profile instance and its criteria are instantiated. We present the comparison method used in the Sect. 6.

4 Automatic Indexing Task

The automatic indexing task (Fig. 1b), is the automatic association of a set of labels from the taxonomic thesaurus to each article. This task needs the following process: (i) a text analysis process to extract keyword and other features from texts and (ii) a machine learning process to learn the classification process from examples.

This paper does not aim at improving the state of the art in multi-classification, nor in ontology learning from text, but instead to propose a method to semantically enrich the ontology by adopting machine learning processes in order to both classify and describe classification, so the gap between the experts perspective and the classification rules representation is filled. First we present the method and then its evaluation.

4.1 Four Steps Method for Automatic Indexing

Our method is based on four following steps.

1. The *vectorization* phase allows generating the matrix of term frequencies from a learning set.
2. The *resolution* allows creating logical constraints (rules) associated to the keywords of the taxonomy (controlled vocabulary) using named entities extracted in the vectorization phase. This phase generates a flat ontology.
3. The *hierarchization* allows to generate a class hierarchy of subsumption of the ontology used to label documents.
4. The *realization* allows searching and deducing the most specific classes of documents to be classified which consists in generating the multi-classification.

Phase 1: uses the indexing work already done by librarians and a text analysis process to extract keywords, to generate a matrix that presents the frequency of each word for each label.

Phase 2: uses the matrix to define rules able to define whether an item should be associated with a label based on the terms contained. Two frequency thresholds are defined, α and β. The words whose frequency is greater than the threshold α are considered as reliable clues. The presence of one of these words is considered as sufficient to consider that the document must be associated with the label. The β threshold is a lower frequency. In this case we need a

combination of β-terms (having frequency greater then β) to confirm the label for a document. More information about the indexing rules can be found in [33].

Phases 3 and 4: are done using standard reasoner: FaCT++, HermiT and Pellet.

Phase 3: the classification provides two types of results. The first is the discovery of the most specific subsuming class. The second allows to infer equivalence classes when the logical constraints are equivalent. On one hand, this means that when a document is labeled with a class that has subsumers, this document will also be labeled by subsumant classes. On the other hand, when a document is labeled with a class that has the equivalence classes then this document is also labeled with equivalent classes. These two elements can achieve a multi-labeling, knowing that the terms of the taxonomy are hierarchical. Accordingly, this is a hierarchical multi-label classification (HMC) process.

Phase 4: the realization phase consists in finding all the most specific classes of individuals. This phase is carried out by the inference engine which enables to deduce all the more specific classes. It also allows to manage multi-labeling while adding subsuming and equivalent classes. As consequence, a document is multi-labeled according to a hierarchy.

4.2 First Evaluation of the Approach

In this section we present a preliminary evaluation of the indexing approach. Due to the lack of real data for our platform, the evaluation is based on the *delicious* dataset available on the Mulan[2] project web site already used in some multilabel-classification works [11]. It was extracted from the del.icio.us social bookmarking site on the 1st of April 2007 and contains textual features and tags of webpages. The characteristics of this dataset, used to train the classifier for tag recommendation, are illustrated in (Table 1).

With this dataset the (*phase 1)* manual multi-classification and the (*phase 2)* feature extraction tasks are not necessary: features and tags are already associated with documents and a sub-dataset is predefined for the (*phase 3*) learning of the prediction model. Our prediction model is the set of α and β-rules. The ontology is populated (*phase 4)*, and reasoners are used to perform the multi-classification task. The results produced by the reasoner are not only a multilabel-classification of documents, but also a hierarchical reorganization of tags based on the equivalence rules. Table 2 presents a benchmark of the three best reasoners evaluated on different hardware.

Table 2 shows that the β-type rules are much more time and memory consuming. We have only one result to show. Only FaCT++ produced a result with the best machine and an ontology without any instance (i.e. document). Within 2 h the reasoner infers a hierarchical reorganization of tags based on the equivalence rules. Yet the ontology populated with documents and equivalent class rules seems

[2] http://mulan.sourceforge.net/.

Table 1 The dataset in some numbers

Dataset	Exemples		Attributes		Labels		
	Train	Test	Numeric	Nominal	Count	Cardinality	Density
Delicious	12,920	3,185	0	500	983	19.020	0.019

Table 2 Reasoner time computation comparison with α *and* β-rules

α-rules	FaCT++	HermiT	Pellet
i7 4Go DDR3	50 s	n.e.m.[a]	n.e.m.
Xeon E3 24Go DDR3	–	8 h	18 h

α and β-rules	FaCT++	HermiT	Pellet
i7 4Go DDR3	n.e.m.	n.e.m.	n.e.m.
Xeon E3 24Go DDR3	n.e.m.	out[b]	out
Xeon E5 128Go DDR3	2 h/out[c]	out	out

[a]Not enough memory
[b]Too much time consumption (more than 3 days)
[c]Only the hierarchical reorganization of tags for the document less ontology

very time consuming even for FaCT++. The ontology with β-type rules is not evaluated in the following steps due to the lack of results provided by reasoners.

This precision-recall evaluation is only based on α-rules, because of the difficulty for reasoners to provide results with β-rules. Table 3 shows that the quality of the results are low. Another approach [17] with this dataset also shows low value for the F-measure.

One of the consequences of our method to create rules (i.e. with an average of 10 terms for all rules) is the creation of some ruleless classes. With this method, for our 983 classes, only 427 have rules. There are 556 classes without labeling rule (obviously, theses classes should have had β-rules). So there are classes that the predictive model can not affect. This impacts very negatively the Recall.

Our experiments allow to determine that only one of the selected terms for the rule is not a sufficient clue. So, α-rules are not convenient for every labelling case, because it requires other word(s) to confirm a label. The solution is β-rules, but the impact on the computation time and memory used is very important.

The realization phase, shows interesting results such as the detection of semantic proximity between the terms tags “blogger”, “blogging”, “wp” (i.e. acronym for wordpress) and “wordpress”. For instance, the semantic proximity is detected between the terms tags blogger, blogging, wp and wordpress, but these results will not be described further here.

Table 3 Evaluation and comparison with a similar work of multi-classification

Evaluation	Precision	Recall	F1-measure
Proposal with α-rules (%)	30	6	10
HOMER [17] (%)	–	–	25

5 Profiling and Profile Refinement

The previous section addresses the indexing of articles (Fig. 1b), profiling (Fig. 1c) (i.e. profiles indexing) is made similar way. Keywords are extracted from the articles that have been loved by users, as well as notes taken by the experts during the conversation with the customer. Manual profiling done by the experts can be used as a basis for the learning task. So automatic indexing is realized in the same way as it is described in the previous section. But manual indexing performed by the experts is not precise, it aims to launch the system in response to the cold start problem. A phase of refinement is necessary to identify more precisely the users needs.

To capture the user behavior, we integrate some sensors in the web application which presents the economic articles recommended. These sensors are of three types. The first one permits to capture the significance of an article due to the login of the user. When the user redirects an the information, his/her logs are updated and taken into account. The second sensor is the *time* spent to read an article. In order to read the complete article, the user has to click on a button with start a timer. The click on the button is itself an important information. The last sensor is a *news filter system* that allows to regenerate the web page regarding the most important information for the user. The navigation behavior is stored in our recommender system to qualify the profile refinement, using the three kinds of footprints: actions, reading time, and domain specific knowledge.

6 Automatic Recommendation Task

The recommendation task (Fig. 1d) is mainly based on the comparison between the user profile and the items available for recommendation. The semantic description of each article and profile is stored in an ontological knowledge base. In order to use the vector space model, it is necessary to transform these descriptions into vectors. This modelling is less expressive than an ontology because the dimensions being orthogonal in a vectorial model, every element of each vector is considered independently ti the other [31].

According to the previous section, the description vectors of the articles and profiles only consist of the instances of the criteria which are directly related to them in the knowledge base (i.e. terms of each facet with which the article instance or profile is related to in the knowledge base).

We develop a *vector expansion* method based on the work of [2]. It is adapted to preserve the ontology knowledge in the vectors. The instances of each criterion are organized in a hierarchy in the knowledge base. For each added instance, its parent instances are also added.

6.1 Similarity Versus Relevance

An article can more or less fit the needs of a user, that's why we use the vectorial model to estimate the relevance. Unlike classic approaches that mix up the concepts of similarity and relevance [28], we distinguish them.

Similarity: $Similarity(x, y) : I \times I \rightarrow [0, 1]$ is a function that evaluates the similarity degree between two objects x et y. In our case, x is an article and y is a profile. This function must satisfy the three properties of *positivity*, *reflexivity* et *symmetry*. The similarity evaluation in a vectorial space can be done by different measures, such as cosine similarity, Jaccard similarity or euclidean distance. In this article, we will illustrate our comment with cosine similarity because it is the most used in the literature. The cosine similarity between two vectors $\vec{a}$ and $\vec{p}$ is based on the measure of the angle Θ between the two vectors.

Relevance: $Relevance\,(x, y) : I \times I \rightarrow [0, 1]$ is a function that measures the relevance degree of an article x for a profile y. This relevance measure must respects the properties of *positivity* and *reflexivity*, but not *symmetry*. Relevance is a concept widely used in the information retrieval field. In our case, the relevance is not binary.

Hierarchical relations between two related topics is a good example to highlight the difference between similarity and relevance: if a given user is interested in the city of Paris, he may be interested in more specific information (monuments, history…), but not in more general information about France. To solve this issue, we use an intermediate vector. The sub-vector $\overrightarrow{s_c}$ consists of the shared instances between the vectors of the article $\overrightarrow{a_c}$ and the profile $\overrightarrow{p_c}$. Therefore we define the relevance for a given criterion c in the following way :

$$Relevance_c(\overrightarrow{a_c}, \overrightarrow{p_c}) = \frac{\omega'_{1,c} \times Similarity_c(\overrightarrow{a_c}, \overrightarrow{s_c}) + \omega'_{2,c} \times Similarity_c(\overrightarrow{p_c}, \overrightarrow{s_c})}{\omega'_{1,c} + \omega'_{2,c}} \quad (1)$$

With S_c the sub-set of shared elements of the set of instances in relation with both the profile $I'_{p,c}$ and the article $I'_{a,c}$; $S_c = I'_{p,c} \cap I'_{a,c}$. $\forall i_{x,c} \in S_c$, the vector $\overrightarrow{s_c}$ consists of the elements of the set S_c; $\overrightarrow{s_c} = \langle i_{1,c}, i_{2,c}, \ldots, i_{t,c} \rangle$. With this method, it is possible to balance the weight of the precision difference between profiles and articles. In our case, we use $\omega'_{1,c} = 1$ et $\omega'_{2,c} = 4$, considering that the precision loss of the profile compared to the article mustn't influence the result by more than 20 %. Moreover, the precision loss of the article compared to the profile must influence greatly the result, i.e. by 80 % in our case. The global $Relevance\,(\overrightarrow{a}, \overrightarrow{p})$ is the sum of the relevance measures for each criterion, eventually weighted. This measure is used in our prototype to sort the results (articles) proposed to the user, according to his profile:

$$Relevance\,(\overrightarrow{a}, \overrightarrow{p}) = \frac{\sum \omega_c * Relevance_c(\overrightarrow{a_c}, \overrightarrow{p_c})}{\sum \omega_c} \quad (2)$$

Table 4 Result of binary and rank evaluation measures for each algorithm

Algorithms	Precision	Recall	F1-measure	Kendall Tau rank	Spearman's rank
A	0.856	0.971	0.910	0.837	0.899
B	**0.916**	0.453	0.607	0.830	0.894
C	0.883	0.181	0.301	0.713	0.694

6.2 Experiments

We have compared two different aspects (binary and rank evaluation) of the results of the articles recommendation via cosine similarity (*C*), cosine similarity with extended vectors (*B*) and *Relevance Measure* with extended vectors (*A*), which handles the precision difference between profiles and articles. For our evaluations, the set consists of 10 profiles and 70 articles corresponding to one day of articles production. For the binary evaluation, a manual selection of the relevant articles has been conducted for each profile by experts. For the rank evaluation, a manual ranking of the relevant articles has been conducted for each profile by experts. In both cases, the results of the different algorithms are compared to the work of experts, considered as an ideal recommendation.

Binary evaluation: To evaluate the binary recommendation we use the classical measures used in information retrieval, i.e. *precision*, *recall* and *F1-measure*, [34]. Every article with a correlation to the profile superior to 0.5 is kept. The results of the binary evaluation of the recommendation presented in Table 4 confirm the results of [31], as for the interest of *vector expansion*, and show that our *Relevance Measure* method, with distinction of the relevance and similarity gives the best result.

Rank evaluation: To evaluate the rank evaluation of the recommendation, we use two most popular linear correlation measures: Spearman's rank and Kendall Tau rank. The output score of these method ranges from -1 to 1, 0 being the lack of similarity, 1 the complete similarity and -1 the opposite. The results of the evaluation are presented in Table 4. The interest of *vector expansion* is also confirmed here. Moreover, the results show that the use of our *Relevance Measure* enhances furthermore the performances of the system.

7 Conclusion

This article describes the architecture developed in the context of global project to develop a recommender system of economic news articles.

The first point addressed in this paper is automatic indexing article and profiles based on controlled vocabularies contained in an OWL-DL ontology using reasoners. We describe the process of using an HMC approach to enrich an already existing ontology to be used for automatic multi-classification of economic news

articles. We decided to capture the prediction model into the taxonomic thesaurus part of the ontology, thus transforming it into a more semantically rich ontology. Based on the early experiments, it was observed that the logical axioms/rules suggested the existence of several subsumption relations that were not present in the taxonomic thesaurus, giving rise to Direct Acyclic Graphs, i.e. a class can have more than one super-class. While this observation is potentially relevant for the refinement of the taxonomic thesaurus and therefore for the classification, a deeper and finer analysis and expert-based experiments have to be performed to better understand the advantages, disadvantages and potential applications. Moreover, our preliminary tests have highlighted the complexity of reasoning on ontology.

The second point addressed in this paper is comparing articles and profiles indexed with vocabularies from the knowledge base using an approach based on the VSM. We present the adaptation of a standard VSM recommender system to our specific method of indexing (e.g. articles and profiles are semantically defined in the knowledge base via relations with the domain knowledge already defined in it). We explain the specific task of comparison that we adapted to our case and finally we evaluate our algorithms using both binary and graded evaluation and show that the results are improved.

References

1. Middleton, S.E., Shadbolt, N.R., De Roure, D.C.: Ontological user profiling in recommender systems. ACM Trans. Inf. Syst. **22**(1), 54–88 (2004)
2. IJntema, W., Goossen, F., Frasincar, F., Hogenboom, F.: Ontology-based news recommendation. In: Proceedings of the 2010 EDBT/ICDT Workshops, EDBT '10, pp. 16:1–16:6. ACM, New York (2010)
3. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94, pp. 175–186. ACM, New York (1994)
4. Billsus, D., Pazzani, M.J.: A personal news agent that talks, learns and explains. In: Proceedings of the Third Annual Conference on Autonomous Agents, AGENTS '99, pp. 268–275. ACM, New York (1999)
5. Balabanović, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Commun. ACM **40**(3), 66–72 (1997)
6. Lops, P., de Gemmis, M., Semeraro, G.: Content-based recommender systems: state of the art and trends. In: Recommender Systems Handbook, pp. 73–105. Springer, Berlin (2011)
7. Rao, K., Talwar, V.: Application domain and functional classification of recommender systems—a survey. DESIDOC J Libr Inf Technol **28**(3), 17–35 (2008)
8. Kondert, F., Schandl, T., Blumauer, A.: Do controlled vocabularies matter? Survey results, pp. 17–35 (2011)
9. Cimiano, P.: Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer, Secaucus (2006)
10. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. **24**(5), 513–523 (1988)
11. Frantzi, K., Ananiadou, S., Mima, H.: Automatic recognition of multi-word terms: the c-value/nc-value method (2000)
12. Pantel, P., Lin, D.: A statistical corpus-based term extractor. In: Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence, AI '01, pp. 36–46. Springer, London (2001)

13. Tsoumakas, G., Katakis, I.: Multi-label classification: an overview. Int J Data Warehouse. Min. **2007**, 1–13 (2007)
14. Bi, W., Kwok, J.T.: Multilabel classification on tree- and dag-structured hierarchies. In: ICML, pp. 17–24 (2011)
15. Garrido, A.L., Gomez, O., Ilarri, S., Mena, E.: An experience developing a semantic annotation system in a media group. In: Lecture Notes in Computer Science, vol. 7337, pp. 333–338. Springer, Berlin (2012)
16. Vogrincic, S., Bosnic, Z.: Ontology-based multi-label classification of economic articles. Comput. Sci. Inf. Syst. **8**(1), 101–119 (2011)
17. Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and efficient multilabel classification in domains with large number of labels. In: ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD08), pp. 30–44 (2008)
18. Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Mach. Learn. **73**(2), 185–214 (2008)
19. Holden, N., Freitas, A.: Hierarchical classification of g-protein-coupled receptors with a pso/aco algorithm. In: IEEE Swarm Intelligence Symposium (SIS 06) (2006)
20. Johnson, I., Abcassis, J., Charnomordic, B., Destercke, S., Thomopoulos, R.: Making ontology-based knowledge and decision trees interact: an approach to enrich knowledge and increase expert confidence in data-driven models. In: Lecture Notes in Computer Science, vol. 6291, pp. 304–316. Springer, Berlin (2010)
21. Elsayed, A.E., El-Beltagy, S.R., Rafea, M., Hegazy, O.: Applying data mining for ontology building. 42nd Annual Conference on Statistics, Computer Science, and Operations Research (2007)
22. Cantador, I., Bellogn, A., Castells, P.: News@hand: a semantic web approach to recommending news. In: Adaptive Hypermedia and Adaptive Web-Based Systems, Lecture Notes in Computer Science, vol. 5149, pp. 279–283. Springer, Berlin (2008)
23. Kelly, D., Teevan, J.: Implicit feedback for inferring user preference: a bibliography. SIGIR Forum **37**(2), 18–28 (2003)
24. Kellar, M., Watters, C., Duffy, J., Shepherd, M.: Effect of task on time spent reading as an implicit measure of interest. Proc. Am. Soc. Info. Sci. Technol. **41**(1), 168–175 (2004)
25. Kelly, D., Belkin, N.J.: Reading time, scrolling and interaction: Exploring implicit sources of user preferences for relevance feedback. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '01, pp. 408–409. ACM, New York (2001)
26. Jawaheer, G., Szomszor, M., Kostkova, P.: Comparison of implicit and explicit feedback from an online music recommendation service. In: Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, HetRec '10, pp. 47–51. ACM, New York (2010)
27. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., Gay, G.: Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. ACM Trans. Inf. Syst. **25**(2), 7 (Apr 2007)
28. Salton, G.: The SMART retrieval system—experiments in automatic document processing (1971)
29. Ahn, J.W., Brusilovsky, P., Grady, J., He, D., Syn, S.Y.: Open user profiles for adaptive news systems: help or harm? p. 11. ACM Press, New York (2007)
30. Getahun, F., Tekli, J., Chbeir, R., Viviani, M., Yetongnon, K.: Relating RSS news/items. In: Web Engineering, No. 5648 in Lecture Notes in Computer Science, pp. 442–452. Springer, Berlin (Jan 2009)
31. Voorhees, E.M.: Query expansion using lexical-semantic relations. In: SIGIR 94, pp. 61–69. Springer, London (Jan 1994)
32. Fellbaum, C.: WordNet: an electronic lexical database. MIT Press, Cambridge (1998)
33. Werner, D., Silva, N., Bertaux, A., Cruz, C.: Using DL-reasoner for hierarchical multilabel classification applied to economical e-news. In: Accepted for Publication in Proceedings of the Science and Information (SAI) Conference, London (Aug 2014)
34. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: SIGIR 94, pp. 3–12. Springer, London (Jan 1994)

Alignment of Time Series for Subsequence-to-Subsequence Time Series Matching

Vineetha Bettaiah and Heggere S. Ranganath

Abstract The success of time series data mining applications, such as query by content, clustering, and classification, is greatly determined by the performance of the algorithm used for the determination of similarity between two time series. The previous research on time series matching has mainly focused on whole sequence matching and to limited extent on sequence-to-subsequence matching. Relatively, very little work has been done on subsequence-to-subsequence matching, where two time series are considered similar if they contain similar subsequences or patterns in the same time order. This paper presents an effective approach capable of handling whole sequence, sequence-to-subsequence and subsequence-to-subsequence matching. The proposed approach derives its strength from the novel two stage segmentation algorithm, which facilitates the alignment of the two time series by retaining perceptually important points of the two time series as break points.

Keywords Data mining · Dimensionality reduction · Piecewise linear representation · Time series representation

1 Introduction

Many areas of science, engineering and business are generating, archiving and processing vast amounts of time series data. Mathematically, a time series $T = \{T[1], T[2], \ldots, T[n]\}$ is a sequence of n real numbers in the increasing order

V. Bettaiah (✉) · H.S. Ranganath
Computer Science Department, The University of Alabama in Huntsville,
301 Sparkman Dr NW, Huntsville, AL 35803, USA
e-mail: vineetha.bettaiah@gmail.com

H.S. Ranganath
e-mail: ranganat@uah.edu

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_12

of time, where each value has a time stamp. The ability to match two time series, T1 of length m, and T2 of length n, to determine their similarity, is a fundamental and critical step in most time series data mining applications including query by content, clustering and classification [1]. Time series T1 and T2 may differ in length, time scale, amplitude scale, time shift, and amplitude shift. There may be considerable distortion due to time warp and noise, and the elements of T1 and T2 may not align. In such cases, the matching should be established on general shapes and local trends of T1 and T2. There are three time series matching scenarios.

1. Sequence-to-sequence matching
2. Sequence-to-subsequence matching
3. Subsequence-to-subsequence matching

The subsequence-to-subsequence matching, which is the task of establishing similarity based on sufficiently long similar subsequences of T1 and T2, has received very little attention and remains an open research problem. In the simplest case, where T1 and T2 are of the same time scale and basic Euclidean distance is used as the similarity measure, the order of computation for subsequence to subsequence matching is O(n3m). As time series are high dimension data, the computation of dissimilarity between two time series in their raw form is very expensive. The situation becomes even worse, if the data points of the two time series do not align.

This paper presents an approach capable of handling sequence-to-sequence, sequence-to-subsequence, and subsequence-to-subsequence matching effectively and efficiently. A segmentation algorithm that selects perceptually important points as primary break points, a time series alignment algorithm that suggests sequence-to-sequence, sequence-to-subsequence, or subsequence-to-subsequence based on the relational analysis of primary break points, and a hierarchical representation that supports coarse-fine matching are the primary contribution of this paper.

The remainder of this paper is organized five sections. Section 2 reviews briefly the related work. The HPLA based time series matching approach is described in Sect. 3. The approach consists of three major steps: Identification of perceptually important primary breakpoints, alignment of the two time series to identify pairs of corresponding candidate matching segments, and matching of segments in each suggested pair using their HPLA representations. The experimental results using a variety of time series data are analyzed and discussed in Sect. 4. Finally, the conclusion is given in Sect. 5.

2 Related Work

Several distance measures have been developed for the computation of the dissimilarity between T_1 and T_2 [2]. These distance measured are grouped into 4 categories: lock-step, elastic, edit and threshold based distance measures. The computationally efficient lock-step distances are not capable of handling even the slightest misalignment between T_1 and T_2. The elastic measures, such as

Dynamic Time Warping (DTW), allow time series to be stretched or compressed as needed to achieve good matching [3]. As time series are high dimension data and DTW uses dynamic programming requiring $O(mn)$ time, matching time series in their raw form is computationally expensive. Researchers have embraced two approaches for improving computational efficiency. They have developed techniques to speed-up DTW and other time series matching algorithms, and to represent time series compactly while preserving salient attributes.

Sakoe and Chiba [4] improved the efficiency of the DTW algorithm by defining a warp window, and by comparing each data point of T2 (query sequence) with only the data points of T1 (sequence in the data base) that are inside the warp window. The Fast Time Series Evaluation algorithm (FTSE) maps data points of T1 into a grid based on their values. The data point of T1 that matches T2[i] is determined by comparing T2[i] with only the data points of T1 that reside in the same grid cell as T2[i] [5]. The Embedding-Based Subsequence Matching (EBSM) algorithm converts each subsequence of T1 into a k-dimensional vector, where the ith component of the vector is the DTW distance between the subsequence and the ith embedding sequence. Thus, each time series T1 in the database becomes a sequence of vectors. The query sequence T2 is also converted to a vector using the same embedding sequences, and vector matching techniques are used for retrieval. The experimental results in [6] indicate one to two orders of magnitude faster retrieval than the brute force method. Though time series are converted to sequence of vectors offline, the approach generates a large number of vectors with high computational cost. Many widely used methods including DTW are natural for only sequence-to-sequence matching. There are variants of DTW algorithm, which are developed for sequence-to-subsequence matching [7]. Some methods, in order to handle this problem, cut the long time series into non-overlapping short segments, and match each segment with the query sequence [8]. Such approaches cannot retrieve any subsequence other than the stored segments. Faloutsos et al. [9] use a sliding window of size w to convert each time series in the database to a trail in a low-dimensional feature space. The window is placed at all possible position, features are extracted for the subsequence inside the window and used to map the subsequence to a point in the feature space. The trail is partitioned into sub-trails, and each sub-trail is enclosed in a minimum bounding rectangle for indexing purpose. Similarly, the query sequence T2 is mapped to the feature space to determine the sequences for retrieval.

The second approach obtains compact representations of the two time series, and matches them in the representation space. During the past two decades, several representations, such as Discrete Fourier Transform (DFT) [10], Discrete Wavelet Transform (DWT) [10], Singular Value Decomposition (SVD) [11], Piecewise Aggregation Approximation (PAA) [12], Adaptive Piecewise Constant Approximation (APCA) [13], Piecewise Linear Approximation (PLA) [14], Piecewise Polynomial Approximation (PPA) [16], Symbolic Representations (SAX) [15], etc. have been developed.

It is important to note that most of the time series matching research is in the context of query by content, where the focus is on whole sequence or

sequence-to-subsequence matching. The indexing centered algorithms and representations are not very beneficial to many other applications of time series matching. The speed-up techniques beneficial to indexing, for example, are not beneficial to clustering, where pairwise comparison of all time series in the dataset is needed. It is fair to say that, relatively, very little work has been done on subsequence-to-subsequence matching of time series.

Any representation that is suitable for subsequence-to-subsequence matching must segment and represent the matching subsequences in the two time series similarly, even if the two time series differ in translation, time and amplitude scale. Ideally, the segmentation should ensure a one-to-one mapping of segments of the two matching subsequences, and the corresponding segments must be similar. One approach is to partition time series at perceptually important points, and build the representation for each segment independently. The local maxima and minima, and points at which the slope changes abruptly may be taken as the perceptually important points. Bettaiah and Ranganath [17] have clearly shown that the segmentation algorithm used for the generation of PAA, APCA, PLA, and SAX representations do not segment the two time series to meet the stated requirement. Thus, they are not able to support subsequence-to-subsequence matching [19]. The DFT being a global transform is also not able to handle subsequence-to-subsequence matching [17]. The PLA representation has the potential to support the development of algorithms for all matching scenarios if each time series is segmented into identifiable segments by placing breakpoints at the perceptually important local maxima and minima [18]. However, well-known and frequently used segmentation methods (sliding window, top-down and bottom-up) do not guarantee the identification of PIPs as breakpoints.

3 The HPLA Based Time Series Matching Approach

An efficient and effective approach for subsequence-to-subsequence matching is given in this section. The approach consists of three major steps: Identification of perceptually important primary breakpoints, alignment of the two time series to identify pairs of corresponding candidate matching segments, and matching of segments in each suggested pair using their HPLA representations.

3.1 Identification of Perceptually Important Primary Breakpoints

Usually, a time series has many local maxima and minima due to the presence of noise. The goal is to develop an algorithm that ignores minor fluctuations and identifies prominent peaks and valleys that define the general shape of the time series. The following algorithm selects such prominent peaks and valleys as primary breakpoints.

```
Algorithm 1. Determine_Primary_BreakPoints (T)
//Find average raise of local maxima and average fall of
local minima
(MaxMin, MaxMinIndex) = FindMax&Min (T);
//Find all local maxima and minima
(avgRaise, avgFall) = FindAverageRaise&Fall (MaxMin);
//Select candidates for prominent maxima and minima
for i=0 to length(MaxMin)-1
   if MaxMin(i) > avgRaise OR MaxMin(i) < avgFall
      Prominent_MaxMin_I.add(MaxMin(i));
      Prominent_MaxMin_Index_I.add(MaxMinIndex(i));
end

//Select final prominent maxima and minima as primary
breakpoints
i=0;
while i < length(Prominent_MaxMin_I) - 1
   if Prominent_MaxMin_I (i) > 0
      Add index of the global maximum between Promi
      nent_MaxMin_Index_I(i) and
      Prominent_MaxMin_Index_I(i+1) to Promi
      nent_MaxMin_Index_F
   if Prominent_MaxMin_I (i) < 0
      Add index of the global minimum between Promi
      nent_MaxMin_Index_I(i) and
      Prominent_MaxMin_Index_I(i+1) to Promi
      nent_MaxMin_Index_F
   i++;
return (Prominent_MaxMin_Index_F);
```

The algorithm first identifies every maximum (minimum) with a raise (fall) greater than the average raise (average fall) as an initial perceptually important maximum (minimum), and stores its value in Prominent_MaxMin_I. The time index of each entry in Prominent_MaxMin_I is recorded in Prominent_MaxMinIndex_I. The adja-cent elements of Prominent_MaxMin_I are examined to obtain Prominent_MaxMin_F, the final list of perceptually important points.

3.2 Time Series Alignment Algorithm

Let, $\{p_1, p_2, p_3, \ldots, p_N\}$ be the set of breakpoints of T_1 identified by **Determine_Primary_BeakPoints**. Let, A be the $(N \times N \times N)$ relational array, where a_{ijk} is a r-dimensional vector that specifies the relationship between p_i, p_j, and p_k. In this paper, a 2-dimensional vector $a_{ijk} = \left[\left(t_j - t_i\right)/\left(t_k - t_j\right)\left(T_1\left[t_j\right] - T_1[t_i]\right)/(T_1[t_k] - T_1[t_j])\right]$

is used to specify the relationship among p_i, p_j, and p_k. Note, a_{ijk} is computed for all (i, j, k), where $1 \leq i \leq (N-2)$, $i+1 \leq j \leq (N-1)$, and $j+1 \leq k \leq N$. Similarly, B is a $(M \times M \times M)$ relational array, where b_{lmn} specifies the relationship among q_l, q_m, and q_n, and is computed as $[(t_m - t_l)/(t_n - t_m)\ (T_2[t_m] - T_2[t_l])/(T_2[t_n] - T_2[t_m])]$. Note that A and B are invariant to translation, time scale, amplitude shift, and amplitude scale. The primary breakpoint mapping matrix C is computed by matching elements of A and B as follows.

```
Algorithm 2. Align_TimeSeries (T1, T2)
(p1, p2, p3, . . ., pN) = Determine_Primary_BreakPoints
(T1);
(q1, q2, q3, . . ., qM) = Determine_Primary_BreakPoints
(T2);
A = RelationalArray (p1, p2, p3, . . ., pN);
B = RelationalArray (q1, q2, q3, . . ., qM);
cij = 0, for 1 ≤ i ≤ N and 1 ≤ j ≤ M
for i = 1 to N-2
  for j = i+1 to N-1
    for k = j+1 to N
      for l = 1 to M-2
        for m = l+1 to M-1
          for n = m+1 to M
            if (1 - ε1) aijk[r] = blmn[r]) = (1 + ε1) aijk[r],

                                                for r = 1, 2
                ci1++; cjm++; ckn++;
List_of_Matching_Primary_Breakpoints =
Align_Primary_BreakPoints (C);
return (List_of_Matching_Primary_Breakpoints);
```

The contents of matrix C suggest possible correspondences between the breakpoints of T_1 and T_2. For example, a high value of c_{il} suggests that p_i in T_1 is very likely to correspond to q_l in T_2. If c_{jm} is zero or close to zero then p_j is unlikely to correspond to q_m. The algorithm **Align_Primary_BreakPoints** identifies likely correspondences between the primary breakpoints of T_1 and T_2 by analyzing C. The following example illustrates the use of the above algorithm for aligning two time series.

The two time series T1 and T2 to be aligned are given in Figs. 1 and 2, respectively. The time scale of T1 and the time scale of T2 differ by a factor of 2, and the subsequence T1[100:524] is identical to T2 in shape. In other words, T1 and T2 differ in translation, time scale and length. The algorithm Determine_Primary_BreakPoints partitions time series T1 into 6 primary segments by identifying 7 primary breakpoints (including end points) labeled p1 through p7. The primary breakpoints of T1 are {(1, 0), (52, 0.1735), (191, −0.1979), (263, 0.2072), (431, 0.16), (504, 0.211), (635, −0.1375)}. The five 2-dimensional arrays that make $7 \times 7 \times 7$ relational array A are shown in Fig. 3.

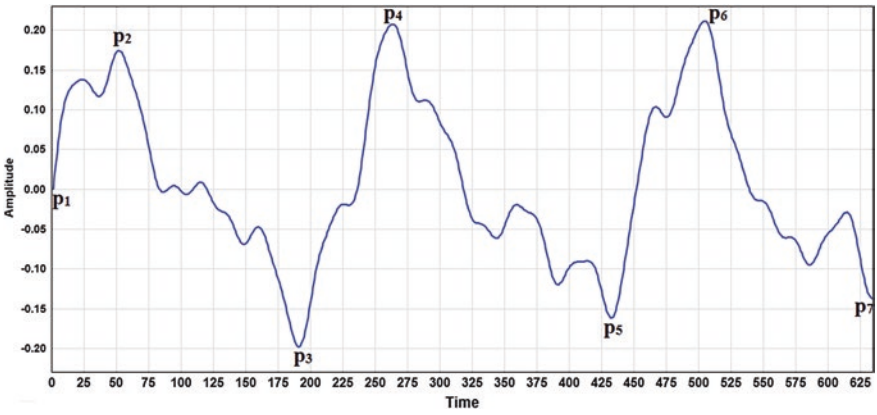
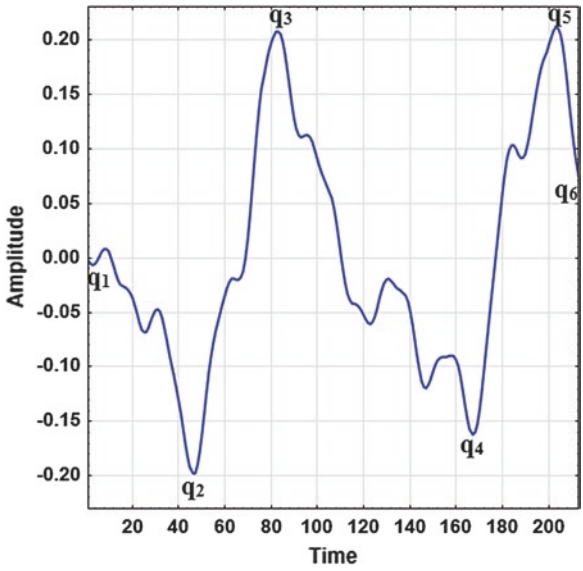


Fig. 1 The time series T_1 of length 635 with 7 primary breakpoints

Fig. 2 The time series T_2 of length 212 with 6 primary breakpoints



The time series T_2 is partitioned into 5 primary segments and the 6 primary break-points are labeled q_1 through q_7. The primary breakpoints of T_2 are {(1, −0.0023), (45, −0.1969), (82, 0.2061), (167,−0.1622), (203, 0.2112), (213, 0.075)}. The four 2-dimensional arrays that make 6 × 6 × 6 relational array B are shown in Fig. 4.

The vector aijk is taken as a match to blmn if corresponding values are within 15 % of each other. That means ε1 is set to 0.15. The 6 × 7 breakpoint mapping matrix C, where rows represent break points of T2 (q1 through q6), and columns

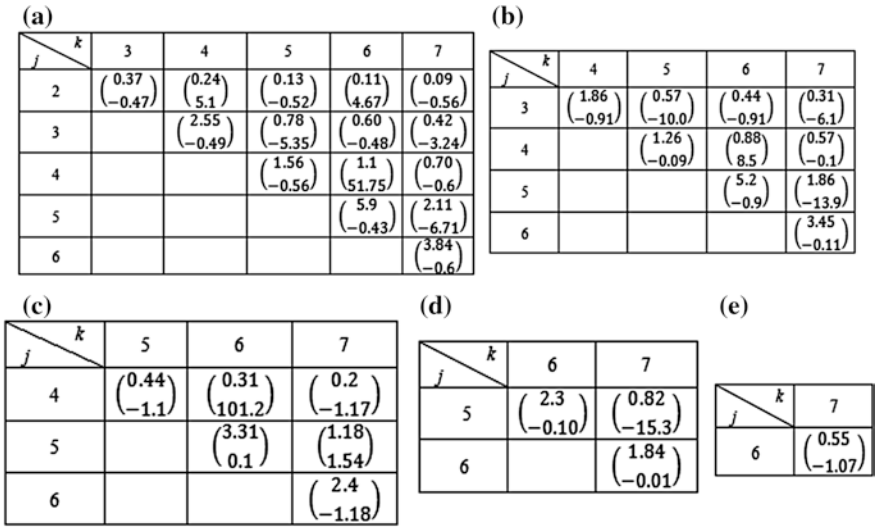


Fig. 3 The relational array for the time series in Fig. 1

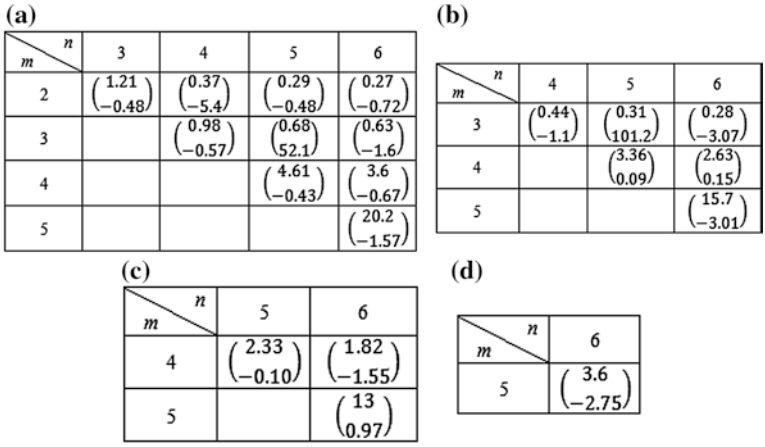


Fig. 4 The relational array for the time series in Fig. 2

represent breakpoints of T1 (p1 through p7) is given Fig. 5a. If the tolerance is increased to 20 % ($\varepsilon 1 = 0.2$), the C matrix changes as shown in Fig. 5b.

In both cases, it is clear that breakpoints p_3, p_4, p_5, and p_6 align with q_2, q_3, q_4, and q_5, respectively. Once again, with only one external input (tolerance ε_1), even when the two time series differ in scale and translation, the algorithm automatically suggested subsequence-to-subsequence matching, and correctly identified likely correspondence between the primary breakpoints of T_1 and T_2. The pairs

Fig. 5 Breakpoint mapping matrices for tolerances of 10 and 20 %

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 3 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

of corresponding candidate matching segments suggested by **Align_Primary_Breakpoints** for further matching are (p_3p_4, q_2q_3), (p_4p_5, q_3q_4), (p_5p_6, q_4q_5).

3.3 Matching of Segments Using the HPLA Representations

The alignment algorithm gives a list of pairs of segments to be matched to determine the similarity between the two given time series. It does not consider the shape of segments in the decision making process. In order to ascertain that the suggested pairs of segments indeed match, further processing is necessary. Depending on the application, one of the following two approaches may be taken.

1. Euclidean distance or DTW may be used to compute the similarity between the corresponding segments identified by the alignment algorithm. This method is suitable if the application requires the comparison of a specific time series with many time series in a dataset. As each time series in the dataset participates only in one comparison, there is no incentive to develop compact representations of segments for the purpose of matching.
2. The Hierarchical Piecewise Linear Approximation (HPLA), which supports coarse-fine matching of time series, may be used for the determination of similarity between two segments. In HPLA, any segment between adjacent primary breakpoints is called a primary segment. Each primary segment is partitioned recursively into two segments at the optimal point (secondary breakpoint) until the desired level of representation accuracy is achieved. A binary tree is used for recording the segmentation hierarchy of each primary segment (subsequence). The structure of the binary tree is simple. Each non-leaf node represents a subsequence $T[i{:}j]$, its left child represents $T[i{:}p]$, and right child represents $T[p{:}j]$, where p is the optimal break point that splits $T[i{:}j]$ into two sub-segments. Each non-leaf node includes a feature vector, components of which relate attributes of the two sub-segments represented by its two child nodes. The components of the feature vector are $(p-i)/(j-p)$ and $(T[p]-T[i])/(T[j]-T[p])$. The root node represents the subsequence by two line segments. The two non-leaf nodes in level-1 represent the subsequence by 4 line segments, and so on. The representation accuracy increases with the increasing level. Each leaf node specifies the starting point of the segment represented by the node. The HPLA representation of a time series is the time ordered sequence of binary trees of its primary segments.

The two segments in each suggested pair of candidate segments are matched by matching their binary trees. Two primary segments are considered similar if the feature vectors of the corresponding nodes of their binary trees are similar. The time series T_1 and T_2 are considered similar, if a sufficiently long continuous sequence of binary trees in the HPLA representation of T_1 matches a sequence of binary trees in the HPLA representation of T_2.

Therefore, for the example being considered, the six primary segments p3p4, p4p5, and p5p6 of T1, and their corresponding segments q2q3, q3q4, and q4q5 of T2 are further segmented to obtain their HPLA representations. As the components of the feature vectors of the corresponding non-leaf nodes of the corresponding segments are within 15 % of each other, the segments in all three suggested pairs are taken as similar. Therefore, T1 and T2 are taken as similar time series.

4 Experimental Results

Two datasets from UCR (Mallat and OliveOil) [20] and Pseudo Periodic Synthetic Time Series from UC Irvine archive (http://kdd.ics.uci.edu) are used as base time series to create a search set and a query set of time series. The three base time series are shown in Fig. 6. From each base time series, several translated, time scaled, amplitude scaled, and amplitude shifted versions are created. Some of these are corrupted with random Gaussian noise. Finally, the created time series are partitioned into two sets, a search set of 212 time series and query set of 40 time series. The search set includes 84 time series created from Mallat, 44 time series created from OliveOil, and 84 time series created from Pseudo Synthetic data. The query set includes 16 time series created from Mallat, 8 time series created from OliveOil, and 16 time series created from Pseudo Synthetic data. The data is carefully chosen to include several cases of sequence-to-sequence, sequence-to-subsequence, and sub-sequence-to-subsequence matching scenarios.

4.1 Matching Approach and Simulation Results

Each query time series Q in $\{Q_1, Q_2, \ldots, Q_{40}\}$, is matched with all 212 time series in the search set $S = \{T_1, T_2, \ldots, T_{212}\}$, and time series similar to the query time series are identified. Ideally, the matching algorithm should identify all time series in the search set that are created from the same base time series as the query time series, and others should not be selected. The details of the two stage matching approach used are given below.

1. As the 2×1 feature vectors used in the HPLA representation are invariant to amplitude scale and amplitude shift, the time series are not normalized. In fact, normalization is meaningful only in the case of whole sequence matching for

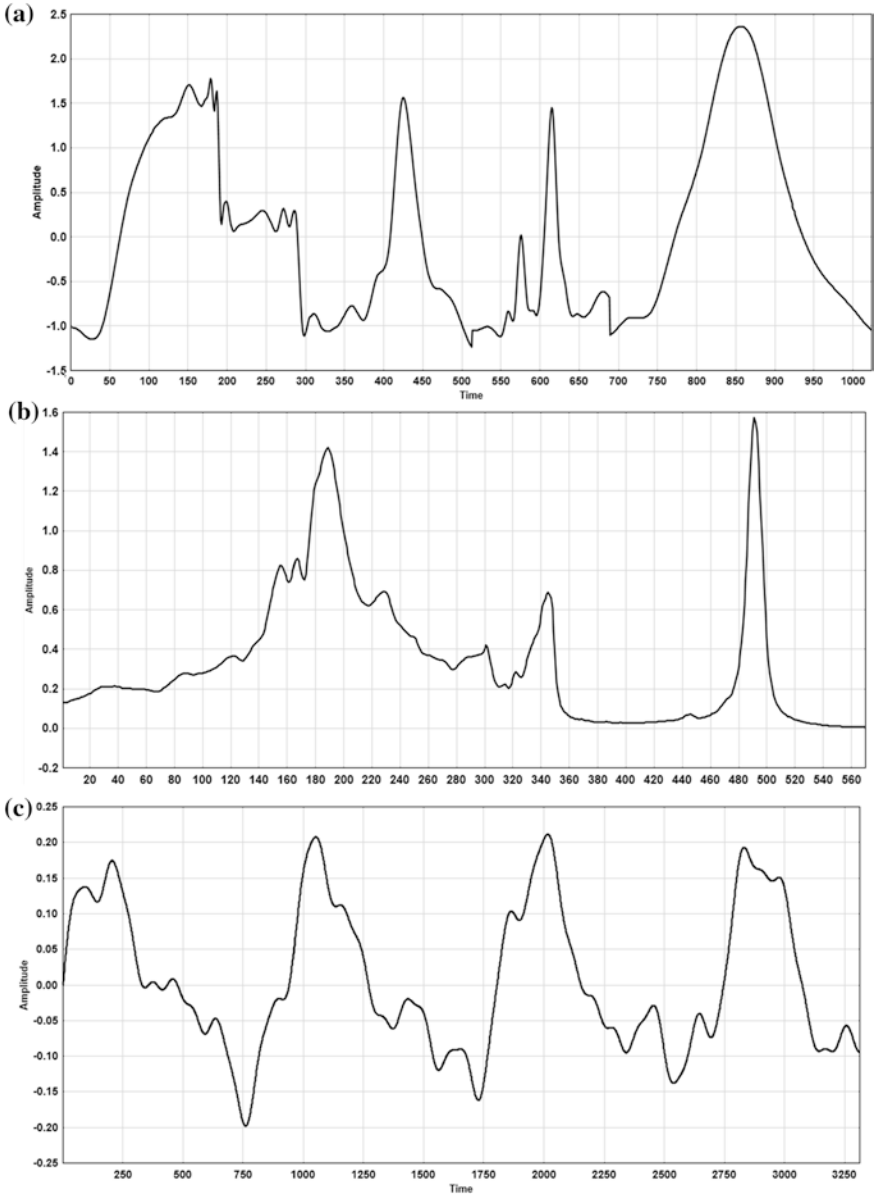


Fig. 6 The three time series used for the creation of the search and query sets. **a** Mallat time series of length 1,024. **b** OliveOil time series of length 570. **c** Pseudo Synthetic time series of length 3,313

representations that are not invariant to amplitude shift and amplitude scale. As the mean and standard deviation of a time series and its sub-series are usually not equal, the normalization is more likely to hurt the matching process than help when the similarity is based on sequence-to-subsequence or subsequence-to-subsequence matching. The optional smoothing step is also not used.

2. For each time series T in S, primary breakpoints are identified using the algorithm **Determine_Primary_Breakpoints**.
3. Each primary segment is partitioned recursively at the optimal point to obtain its binary tree representation [19]. For the sake of uniformity, each primary segment is represented by a complete binary tree with 8 leaf nodes (4 levels). The feature vector of each non-leaf node is computed.

Each query time series *Q* in the query set is matched with each *T* in *S* in two stages as described below.

Stage 1: *Selection of candidate time series from the search set*

The goal is to select a candidate subset of S for further matching in Stage 2. Ideally, the candidate set should include all time series in S that are similar to Q and none of the time series that are not similar to Q. In reality, the set will include a few time series not similar to Q (false positives) and not include a few time series similar to Q (false negatives).

In order to determine the candidate set, the primary breakpoints and the HPLA representation of Q are determined first. Algorithm Align_TimeSeries is used to obtain the list of pairs of potential matching primary breakpoints, which suggest pairs of candidate segments from Q and T for further matching. The value of the parameter ε1 used for comparing the corresponding elements of the relational arrays of T and Q is set at 0.15. If more than 50 % of the primary breakpoints of Q align with the primary breakpoints of T in the same time order, T is selected for further matching in Stage 2. Otherwise, T is not a candidate for further matching.

Stage 2: *Filtering the false positives*

The goal is to filter as many false positives as possible by matching the HPLA representations of Q and each T. The two segments (one of Q and one of T) in each suggested pair are matched by comparing their HPLA representations to determine if they are similar. The matching of the two binary trees begins with the matching of the feature vectors stored in their root nodes. If the two feature vectors are not similar, the two segments are considered as dissimilar. If the two feature vectors are similar, the matching is continued using non-leaf nodes in the next level. This process terminates when all corresponding non-leaf nodes are exhausted. The user may limit matching to root nodes, or consider the non-leaf nodes in other levels, depending on the level of accuracy desired. The time series T is considered similar to Q, if 75 % of the suggested segment pairs of T and Q are found similar. The simulation results, evaluated and discussed in the next section are tabulated in Table 1. Because of space constraints, results for 12 out of 40 query time series are shown.

Table 1 Simulation results of the HPLA representation based time series matching

Query time series	Number of candidates from stage 1	Number of time series selected in stage 2	Number of false positives	Number of false negatives	Recall	Precision	Pruning Power
Q_1	95	85	3	2	0.9761	0.9647	0.0859
Q_2	101	84	0	0	1	1	0.1328
Q_3	90	84	2	2	0.9761	0.9761	0.0468
Q_4	92	84	0	0	1	1	0.0625
Q_{17}	49	44	0	0	1	1	0.0297
Q_{18}	51	44	0	0	1	1	0.0416
Q_{19}	50	44	0	0	1	1	0.0357
Q_{20}	48	46	2	0	1	0.9565	0.0238
Q_{25}	94	85	1	0	1	0.98	0.0781
Q_{26}	90	84	0	0	1	1	0.0468
Q_{27}	89	85	1	0	1	0.98	0.0390
Q_{28}	88	84	0	0	1	1	0.0312

4.2 Evaluation of Simulation Results

The two metrics, recall and precision, commonly used for evaluating the performance of database retrieval methods are used for the evaluation of the performance of the HPLA based time series matching approach. Recall is defined as the ratio of the number of truly matching time series retrieved to the total number of matching time series in the search set. The value of recall is in the range [0, 1], where higher value indicates better performance. A recall value of 1 indicates that all matching time series in the database are retrieved without any false negatives. Precision is defined as the ratio of the truly matching time series retrieved to the total number of time series retrieved from the search set. The value of precision is also in the range [0, 1]. A value of 1 indicates that there are no false positives.

Pruning power is another frequently used metric which specifies the number of time series considered for matching. For the current situation, as an HPLA based indexing method is not developed, the number of time series ruled as non-matching by Align_TimeSeries may be used as a measure of the pruning power. In this paper, pruning power is defined as the ratio of the number of time series selected for matching by Align_TimeSeries minus the number of time series similar to Q in S to the total number of time series in S minus the number of time series similar to Q in S.

The simulation results in Table 1 (shown only for 12 query time series), which lists number of false positives, number of false negatives, recall, precision, and pruning power are very impressive. The important observations about the HPLA based matching approach, and results are discussed below.

1. Each time series in the query set is similar to a subset of time series in the search set. The similarity may be based on sequence-to-sequence, sequence-to-subsequence or

Table 2 The average pruning power, recall and precision for Mallat, OliveOil, and Pseudo Synthetic query sets

Query set	Average pruning power	Average recall	Average precision
Mallat	0.0469	0.9925	0.9921
OliveOil	0.0394	0.9943	0.9836
Pseudo Synthetic	0.0423	0.9937	0.9925

subsequence-to-subsequence matching. The matching scenario that establishes similarity between query and search time series is not known in advance. For example, consider Q_2 of length 512, which is created from Mallat of length 1,024 by adding 1.85 to each time scaled (factor of 2) value. In the search set there are 23 time series similar to Q_2 based on whole sequence matching, 31 time series similar to Q_2 based on sequence-to-subsequence matching, and 30 time series similar to Q_2 based on sub-sequence-to-subsequence matching. A total of 84 time series in S are similar to Q_2, and the remaining 128 time series in S are not similar. In Stage 1, the algorithm **Align_TimeSeries** has identified the correct matching scenario for Q_2, and for all other 39 query sequences.

2. The analysis of relative positions of primary breakpoints is able to prune most of the non-matching time series in S from further consideration. For example, in the worst case, 101 candidates are selected by **Align_TimeSeries** for further matching with Q_2 in Stage 2. As there are 84 time series in S that are similar to Q_2, even in the worst case, only 17 additional time series are selected for matching in Stage 2. On the average, for the Mallat query set, only 90 time series are selected as candidates for matching in Stage 2, giving an average pruning power of 0.0469 $((90 - 84)/(212 - 84))$. The average pruning powers are also given for OliveOil and Pseudo Synthetic query sets in Table 2.
3. The segment by segment matching of the HPLA representations of Q and T has filtered most of the false positives selected in Stage 1. For example, 101 candidates selected by **Align_TimeSeries** for further matching in Stage 2 include all 84 time series that are similar to Q_2, and 17 time series that are not similar to Q_2 (false positives). The HPLA based matching in Stage 2 filters all the false positives, and retains all 84 time series that are similar to Q_2. The ability of the method in eliminating or significantly reducing the number of false positives is consistently good for all 40 cases.
4. The effectiveness of the HPLA based time series matching is obvious from high recall and precision values, which are close to the ideal value of 1. The low values of pruning power (close to the ideal value of 0) indicate the potential for building an efficient HPLA based indexing approach. The average recall and precision for the three groups of query sequences (Mallat, OliveOil, and Pseudo Synthetic) are given in Table 2.
5. The approach requires the specification of only two tolerance parameters ε_1 and ε_2. The value of ε_1 directly affects the number of false negatives and false positives in Stage 1. As ε_1 increases, the number of false positives increases and the

number of false negatives decreases. In order to avoid the risk of losing matching time series in Stage 1, the value of ε_1 should be relatively high. Similarly, the value of ε_2 also affects the number of false positives and false negatives in Stage 2. As the value of ε_2 increases, the number of false positives increases and the number of false negatives decreases.

5 Conclusion

In summary, the HPLA based time series approach described in this paper handles all three matching scenarios uniformly by identifying the appropriate scenario to be used through the analysis of the relative positions of the perceptually important primary breakpoints in query and search sequences. There is no need for the user to specify the type of matching scenario needed. The approach identifies the matching scenario automatically, and also prunes most of the non-matching time series in the search set from further consideration. The HPLA based matching algorithm filters most of the false positive, and achieves high precision and recall. The approach is invariant to time and amplitude translation and scale differences between the two time series matched, and requires the specification of two simple tolerance parameters as external input.

References

1. Esling, P., Agon, C.: Time-series data mining. ACM Comput. Surv. **45**, 34 (2012)
2. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. Proc. Very Large Databases **1**, 1542–1552 (2008)
3. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD Workshop, pp. 359–370. AAAI Press, Palo Alto (1994)
4. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech Sig. Proc. **26**(1), 43–49 (1978)
5. Morse, M.D., Patel, J.M.: An efficient and accurate method for evaluating time series similarity. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 569–580 (2007)
6. Athitsos, V., Papapetrou, P., Potamias, M., Kollios, G., Gunopulos, D.: Approximate embedding-based subsequence matching of time series. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 365–378 (2008)
7. Park, S., Kim, S., Chu, W.W.: Segment-based approach for subsequence searches in sequence databases. In: Symposium on Applied Computing, pp. 248–252 (2001)
8. Zhu, Y., Shasha, D.: Warping indexes with envelope transforms for query by humming. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 181–192 (2003)
9. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. In: Snodgrass, R.T., Winslett, M. (eds.) Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 19–429 (1994)

10. Wu, Y., Agrawal, D., Abbadi, A.E.: A comparison of DFT and DWT based similarity search in time-series databases. In: Proceedings of the 9th International Conference on Information and Knowledge Management, pp. 488–495 (2000)
11. Hung, N.Q., Anh, D.T.: An improvement of PAA for dimensionality reduction in large time series databases. In: Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence, pp. 698–707 (2008)
12. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowl. Inf. Syst. **3**, 263–286 (2001)
13. Chakrabarti K, Keogh E, Mehrotra S, Pazzani M.: Locally adaptive dimensionality reduction for indexing large time series databases. ACM Trans. Database Syst. **27**, 118–228
14. Keogh, E., Pazzani, M.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In: 4th International Conference on Knowledge Discovery and Data Mining, pp. 239–241 (1998)
15. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing SAX: a novel symbolic representation of time series. Data Min. Knowl. Discov. **15**, 107–144 (2007)
16. Lemire, D.: A better alternative to piecewise linear time series segmentation. In: SIAM Data Mining (2007)
17. Bettaiah, V., Ranganath, H.S.: An analysis of time series representation methods. In: Proceedings of the 2014 ACM Southeast Regional Conference, Article 16, p 6 (2014)
18. Bettaiah, V., Ranganath, H.S: Two stage segmentation for efficient time series matching. In: 2nd International Conference on Research in Science, Engineering and Technology, pp. 29–35 (2014)
19. Bettaiah, V., Ranganath, H.S: An effective subsequence-to-subsequence time series matching approach. In: Science and Information (SAI) Conference, pp. 112–122 (2014)
20. The UCR Time Series Classification/Clustering. http://www.cs.ucr.edu/eamonn/time_series_data/. Retrieved Aug 2013

The Effects of Typing Demand on Emotional Stress, Mouse and Keystroke Behaviours

Yee Mei Lim, Aladdin Ayesh and Martin Stacey

Abstract Past research found that cognitive effort is related to emotion, which negative emotion may influence task performance. To enhance learning experience, it is important to have an effective technique to measure user's emotional and motivational affects for designing an adaptive e-learning system, rather than using a subjective method that is less reliable and accurate. Keystroke and mouse dynamics analyses shed light on a better automated emotion recognition method as compared to physiological methods, as they are cheaper, non-invasive and can be easily set up. This research shows that unification of mouse and keyboard dynamics analyses could be useful in detecting emotional stress, particularly stress induced by time pressure, text length and language familiarity. The changes of mouse and keystroke behaviours of the students are found cohere with their task performance and stress perception. However anomalies in mouse and keystroke behaviours present when the students are pushed beyond their capabilities.

Keywords Emotional stress · Keyboard dynamics · Mouse dynamics · Language familiarity · Text length

Y.M. Lim (✉)
Faculty of Applied Sciences and Computing, Tunku Abdul Rahman University College, Kuala Lumpur, Malaysia
e-mail: ymlim@acd.tarc.edu.my

A. Ayesh · M. Stacey
Faculty of Technology, De Montfort University, Leicester, UK
e-mail: aayesh@dmu.ac.uk

M. Stacey
e-mail: mstacey@dmu.ac.uk

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_13

1 Introduction

If mental or cognitive efficiency of a learner can be measured, or the learners' instruction condition can be diagnosed, it is believed that the effectiveness of learning can be enhanced, as adaptive learning materials and customized assessment can be given based on individual needs and performance. This research examines how mouse and keyboard dynamics analyses can be used in detecting emotional stress induced by high job demand, such as long typing task with time pressure and unfamiliar language. Cognitive load theory (CLT) emphasizes devising effective instructional procedures to enhance learning based on the understanding of human's cognitive process working with long-term and short-term memory [1, 2]. However, learner's performance could also be affected by motivational and emotional factors as suggested by Beilock and Ramirez [3]. Negative emotion could inhibit necessary resources being recruited for further cognitive process by human mental, which prevent optimal skill execution. Therefore, emotion and motivational factors should be considered when developing instructional procedures in a learning environment, in order to ensure that the students are always ready to accept and execute demanding learning tasks. There are a few measurement techniques that can be used to assess cognitive load of a learner, which include subjective methods, physiological tests and task performance-based measurement [4]. Subjective methods include traditional survey method, which usually requests the users to perform self-assessment on their mental effort or emotion, which could be erroneous and unreliable. Physiological measurements are most common as the accuracy of measuring mental activities and emotions using biological data is often higher and more reliable. However, the high cost of equipment and expertise, and the need of special setup of equipment become a hindrance in promoting ubiquitous computing. Task performance-based method commonly used in psychological tests. It is considered as an objective and standardized measure of individual's task performance, cognitive ability, aptitude, emotional functioning, etc. [5]. It is suggested that in a task-specific environment, user stress levels can be varied according to two factors: demand (e.g. excessive demand on worker production, especially to meet a deadline) and lack of control over work processes [6]. Besides, misfit between job demands and individual capabilities intensifies the stress effect [7]. Therefore, by changing the workloads and control of tasks the user stress level can be changed. However, the limitations of psychological tests are they are usually done based on social science approach and there is lack of automation in the assessment. Other emotion detection methods include facial expressions recognition. Although it provides promising accuracy rates, the assessments could be computationally intensive. To eliminate the downsides of the aforementioned methods, some research examines the possibilities of using mouse behaviour or keyboard dynamics analyses in emotion detection. The promising results shown by these research demonstrate great potential in developing a cost-effective, non-invasive, computational feasible and automated adaptive e-learning system that is able to provide customized learning environment based on learner's cognitive efficiency and affective state.

2 Related Work

2.1 Mouse and Keyboard Dynamics Analyses in Emotion Detection

Various studies have been carried out to examine the effects of stress on users' psychological and physiological reactions when using a mouse or keyboard. Research by Wahlström et al. [8] and Heiden et al. [9] showed that users would demonstrate increased psychological and physiological reactions using a computer mouse if they work under time pressure, verbal provocation and precision demand. Pusara and Brodley [10] used mouse dynamics to detect anomalous behaviour through user's mouse movements. Tsoulouhas et al. [11] found that mouse activities are affected by boredom induced by lengthy presentation of a course on computer screen. They reported that mouse movement speed, inactivity occurrences and durations, and movement directions would change significantly when the students claimed boredom. Despite lesser emotion detection research based on keyboard dynamics was done, the research by Vizer et al. [12] showed that different typing behaviours, such as changes of patterns in key latency, keystroke speed, delete keys, navigation keys and other keys (such as letter and number keys) are demonstrated when the users are induced by higher physical and cognitive stress. Although these studies demonstrate promising results but most of them have considered mouse and keyboard dynamics analyses in isolation. Many limitations would exist when using these methods alone. First, only small amount of information can be retrieved from mouse and keyboard. The information produced by these devices are unstructured and different from each other. Furthermore, some tasks require both devices to be used. When the users are working on one device, the other would be idle for long time. Lim et al. [13] investigated how keyboard and mouse dynamics are affected by mental arithmetic task with time pressure. Their research findings suggest that when task demand increased, task error, task duration, passive attempt, stress perception and mouse idle duration may increase, while mouse speed, left mouse click and keystroke speed decreased. The significant correlations between mouse and keystroke features suggests that it is better to unify both to complement the analyses.

2.2 Typing Task Demand

In order to induce emotional stress in the experiments, some methods that are widely adopted include mental arithmetic, N-back number recall, time pressure, reading aloud, viewing affective picture or video, emotive text reading and story telling (see [12, 14–16]). Among these methods, some are very useful to enable task demands to be objectively measured, for example mental arithmetic, N-back number recall and time pressure. There is little research done to examine the effect of typing task on

emotion since most of the tasks in an e-learning environment require text typing (e.g. post discussion). Besides, there is also little research carried out to study the influence of subject familiarity on task performance and physiological behaviour. Tobias and Abramson [17] suggest that lack of familiarity implies that the required cognitive resources or response needed for executing the task may not be available in the learner's repertory (memory). Therefore it would require a more overt response for optimal learning from content with unfamiliar subjects. Hulme et al. [18] also found that memory spans for unfamiliar words are lower than familiar words. Therefore we would like to examine the effects of text length and language familiarity on user behaviour (such as typing rhythms), even though the effect could be small.

However, there are a few issues to consider in typing task demand. The main issue is there are high variations of individual typing skills (such as typing speed), which are caused by individual expertise skills, experience, and environmental factors. According to Davidson and Sternberg [19], a typist's typing speed will increase if he or she is able to look far ahead. Far sight allows superior preparation and optimization of typing movement. Additionally, typing speed can be increased by 10–20 % if full concentration is exerted, and habitual typing behaviour could be broken when individuals engage in activities that are deliberately prescribed to increase their typing speed, such as setting time pressure, and this often leads to mistakes. The second issue in typing task demand is regarding text length. Most research limits the experiments to produce samples from structured and predefined text in order to analyze keyboard dynamics. Many researchers strived to work with relatively short sample phrases, such as username and password (for example [20–26]). Others used free and long text in their studies (see example [27, 28]). However, most of their studies show that both fixed text and free text are equally useful for keyboard dynamics analyses, regardless the length of the text.

3 Methodology

We would like to examine the effects of task demand on emotional stress (*SP*), student's task performance ($B(T)$), mouse behaviour ($B(M)$) and keystroke behaviour ($B(K)$). Task demand is varied by time pressure imposed (Timing), the size of the text to be typed (Text Length) and language familiarity (Familiarity). $B(T)$ consists of three variables: (1) task duration (*TD*) that records the duration to complete a typing task; (2) error (*Err*), which records the number of typing errors (such as missing words and punctuation marks, and spelling errors); and (3) passive attempt (*PA*), which records the number of attempt to give up or wait until the time is up. $B(M)$ is a dataset that consists of (1) mouse speed (*MS*), which is the average mouse speed (pixels per millisecond (ms)), (2) mouse idle duration (*MID*) (ms), (3) mouse idle occurences (*MIO*), and (4) mouse click rate (*MC*) (per ms), which includes left mouse click rate (*MCL*) and right mouse click rate[1] (*MCR*). $B(K)$ is a

[1] *MCR* was removed later due to no data.

dataset that consists of (1) average keystroke speed per key (per second)(*KS*), (2) average keystroke latency (Down-Down time) (ms), and (3) error key occurrences (*EK*), which recorded the number of error keys used in a task, such as backspace key (*BSK*) and delete key[2] (*DK*). Lastly, *SP* is obtained from user's self evaluation reports using 7-point Likert scale.

3.1 Experiment Setup

Data Collection Design To collect the primary data from mouse and keyboard, two programs are written in Java and VB.NET separately to acquire mouse raw data and the virtual-key codes generated by the Windows platform. The collection of mouse raw data include clicked mouse button, mouse location that is recorded every 10 ms, and their respective event time (ms). The collection of keyboard raw data include hit key code and its respective event time (ms). To protect user's privacy, the virtual-key codes were transformed into special codes automatically by the program. For instance, a number key or a letter key was recorded as 'k', delete key as '?' and backspace key as '*'. The actual hit key-codes are not stored. An imitation of the online-assessment system is built to imitate the students' e-learning environment. Six different typing tasks were set based on different text length and language familiarity. Three questions are set in English (as familiar language) and 3 in German (as unfamiliar language). The requirements of the typing tasks are shown in Table 1. The reason to set much longer text for Question 5 and Question 6 is to push the participant's performance beyond limit especially under time pressure. Longer text is also believed to lead to boredom, tiredness and fatigue (Selye [29] relates boredom to stress, as understress). In order to reduce invariabilities of mouse movements and typing behaviours that would affect the results, the students must use normal, external and common mouse and keyboard devices during the experiments. All the computers used in the experiments are equipped with Windows 7, 3.10 GHz CPU, 4 GB RAM, 17″ monitor with the resolution of 1,024 × 768 pixels, external standard QWERTY HID keyboard and external HID-compliant mouse.

To determine the time limit to be given to the participants, we conducted a pilot test with 13 samples. The average duration to complete Question 3 is 26,730 ms, Question 4 is 30,602 ms, Question 5 is 30,247 ms (100 % of them made more than 40 typing errors), and Question 6 is 24,952 ms (76.92 % of them made more than 40 typing errors). Therefore we set 30 s time limit for the experimental group to complete each task. The pilot results show that it is impossible to complete 63 words within 30 s without any error.

Procedures All participants are given the same set of typing tasks. Each question is displayed on individual page and they must type the given text into a

[2] *DK* was removed later due to no data.

Table 1 Typing task requirements

Question	Characteristics		Text length		Actual text to type
	Length	Familiarity	Words	Letters (without space)	
1	Short	Familiar	5	21	Time flies like an arrow
2	Short	Unfamiliar	5	25	Ich bringe Sie zum Flughafen
3	Medium	Familiar	20	94	Study by Lazar (2003) has shown that about one third of the time on computer is spent on frustrating experiences
4	Medium	Unfamiliar	20	99	Was denken Sie daruber? Ich fahre morgen nach Dresden. Wann isst du zu Mittag? Das schmeckt! Schonen Tag noch, Tschau
5	Long	Familiar	63	459	Vizer stated that cognitive-stress tasks such as mental-multiplication and number-recall are widely used to induce cognitive-stress. Their results show that those keystroke-features that can be changed by cognitive-stress include keystroke-pause-length, keystroke-time, deletion-keys, navigation-keys and other keys (such as letter-keys and number-keys). However, we are more interested to examine the user-interface factors that may cause cognitive-stress in the e-learning environment, which include navigation designs
6	Long	Unfamiliar	63	451	Jeder hat das Recht auf Bildung. Die Bildung ist unentgeltlich, zum mindesten der Grundschulunterricht und die grundlegende Bildung. Der Grundschulunterricht ist obligatorisch. Fach- und Berufsschulunterricht mussen allgemein verfugbar gemacht werden, und der Hochschulunterricht muB allen gleichermaBen entsprechend ihren Fahigkeiten offenstehen. Die Bildung muB auf die volle Entfaltung der menschlichen Personlichkeit und auf die Starkung der Achtung vor den Menschenrechten und Grundfreiheiten gerichtet sein

To enable the students to use the standard keyboard to type the text, the umlauted letters of German is replaced with English alphabet (for example, ‘ö’ is replaced with ‘o’)

designated textbox. Each task allows the students to click on "give up" button if they do not wish to continue. When the first question is revealed, the start time (in milliseconds) is recorded by the system. Then the participants must use the mouse device to submit the task once finish typing, and the end time (in millisecond) would be recorded. A survey form would then be displayed. They must assess and indicate whether they felt stressed when typing the text (1 for strongly disagree and 7 for strongly agree). At the same time $B(T)$, $B(M)$ and $B(K)$ are computed. The second question would be displayed next, the start time would be recorded and the subsequent process would be repeated until the last question.

The Control Group and Experimental Group Seventy-seven year-2 students from Bachelor Degree in Computer Science and Bachelor Degree in Information Technology were recruited based on voluntarily basis without any incentive. All of them passed the English test in Malaysian Certificate of Education, but none of them knows German language. Unfortunately only 60 participants provide valid data (aged between 18 and 24 years old, 90 % male). Thirty students in the control group are required to type the predefined texts of all 6 questions without any time constraint. The 30 students in the experimental group are given 30 s time limit for each question. If they could not complete the task on time, then the page would be submitted automatically.

4 Results

To observe the effects of task demand with time pressure on emotional stress (*SP*), task performance ($B(T)$), mouse behaviour ($B(M)$) and keystroke behaviour ($B(K)$), we conducted some statistical tests. To ensure homogeneity between the two subject groups in our experiments, we transformed *TD* and *MID* using *Log*10 function. The following subsections discuss the statistical results.

4.1 The Effects of Demand by Question and Time Pressure on Emotional Stress

We first tested the main effects of task demand by question (Question) and time pressure (Timing) on student's stress perception (*SP*) using Multivariate Analysis of Variance (MANOVA) [30]. *SP* increased when Question increased for both groups of students (see Fig. 1). However, Timing provides no effect on *SP* ($p = 0.663$), and it has no interaction with Question ($p = 0.446$), which indicates that even the students are given time pressure, they do not have different stress perceptions compared to those without time pressure.

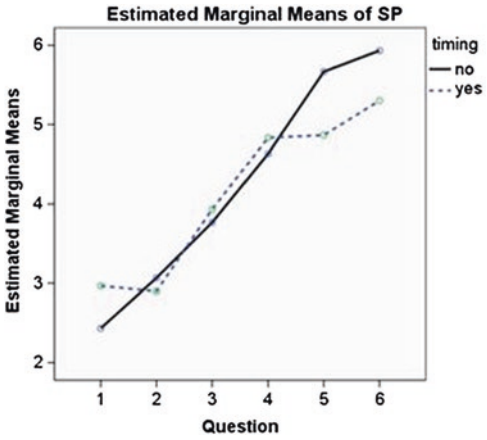


Fig. 1 Stress perception (*SP*) increased according to question. The differences between questions are significant at $p < 5e^{-18}$ level, but the effect of timing is not significant ($p = 0.633$)

4.2 The Effects of Question and Timing on Task Performance, Mouse Behaviour and Keystroke Behaviour

Task Demand and Time Pressure affect all behaviours, except *MCL* (see Table 2). However, *MCL* is affected by the interaction between Question and Timing. To further analyse the variations between Question and Timing effects on the three behaviours, we performed Tukey Post Hoc Tests. The results are illustrated in Figs. 2, 3 and 4. The arrow markers in each graph indicate significant differences or changeover points between classes. To observe the effects of Question and Timing on *B*(*T*), we first observe *TD*. In Fig. 2, *TD* increased significantly after Question 2, and then it decreased significantly after Question 4 or Question 5.

Table 2 MANOVA tests of the between-subjects effects

Question			Timing			Question x timing		
Behaviour	Feature	*p*-value	Behaviour	Feature	*p*-value	Behaviour	Feature	*p*-value
B(T)	TD	*0.0000*	B(T)	TD	*0.0272*	B(T)	TD	0.6918
	Err	*0.0000*		Err	*0.0000*		Err	*0.0000*
	PA	*0.0002*		PA	*0.0010*		PA	*0.0002*
B(M)	MS	*0.0488*	B(M)	MS	*0.0247*	B(M)	MS	0.6795
	MID	*0.0000*		MID	0.6982		MID	0.5710
	MIO	*0.0000*		MIO	*0.0000*		MIO	0.0536
	MCL	0.3396		MCL	0.4537		MCL	*0.0300*
B(K)	KS	*0.0005*	B(K)	KS	*0.0143*	B(K)	KS	*0.0304*
	KL	*0.0023*		KL	*0.0000*		KL	0.8718
	EK	*0.0387*		EK	0.1708		EK	0.3855

Emphasised cell indicates that the effect is significant at the level of $p < 0.05$

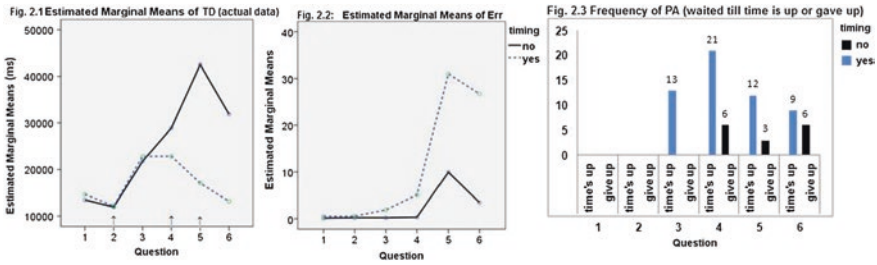


Fig. 2 Mean plots of task performance according to question and timing factors

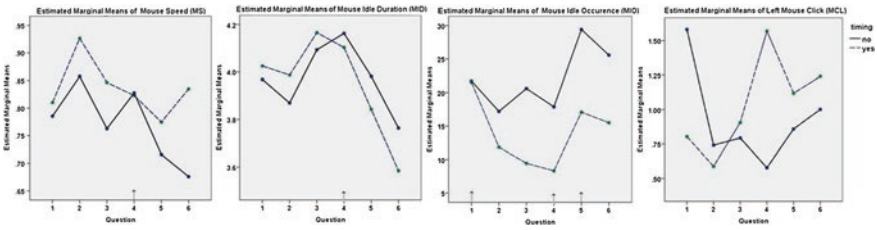


Fig. 3 Mean plots of mouse behaviour features according to question and timing

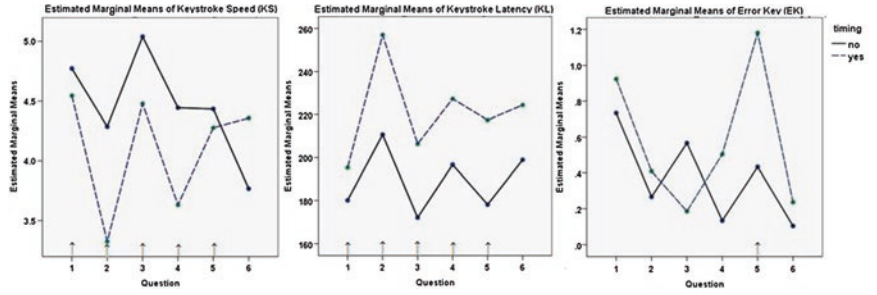


Fig. 4 Mean plots of keystroke behaviour features according to question and timing

The students in the experimental group obviously spent lesser duration to type in long text than medium-length text, while the students in control group spent shorter time for Question 6 than Question 5. In terms of Timing, the experimental group has initially spent indifferent duration with the control group students when short text length is introduced. However at Question 4, the students who are given time pressure obviously spend shorter time to finish the task. We then observe the number of students who made errors in answering the questions. Figure 2 shows that the students in experimental group generally made more errors than those in control group. *Err* is significantly different according to Timing and Question (both with $p < 0.05e^{-7}$). Besides, there is high number of students who scored

Err = 0 for Question 5 (total students = 30) and Question 6 (total students = 31) although we predicted that no one should be able to complete these two questions in the experimental group. In terms of *PA*, the number of students who gave up the task does not increase until Question 5; while the number of students who waited until the time is up has dropped after Question 4 (which indicates that more students submitted the task before the time is up). The anomalies of *Err* and *PA* after Question 4 show that the students have started to "cheat" from Question 5 onwards, where they copied-and-pasted the text directly to the text box instead of typing. Based on the users who obtained perfect score (*Err* = 0) for Question 5 and Question 6, 95.08 % of them used less than 20 s to complete the task (and 100 % spent less than 30 s to type 63 words), including those who were not given any time pressure. This phenomenon shows that when the users are given long text, even they are not pressured by time, the job will still be considered too demanding (*SP* is highest at Question 5 and 6). Such behaviours demonstrate that the students may possibly have lost motivation, overstressed or under-stressed.

4.3 The Effects of Text Length and Language Familiarity on Emotional Stress and the Three Behaviours

We examine how Text Length and Familiarity affect *SP*, *B*(*T*), *B*(*M*) and *B*(*K*). Table 3 shows that Text Length significantly changes *SP*, *B*(*T*) and *B*(*M*), except *MCL*. However Text Length does not affect *B*(*K*) at all. This indicates that user's typing behaviour is not affected by the Text Length. On the other hand for Familiarity, it affects *SP*, and it changes all the features of *B*(*K*) but not *B*(*M*) (except *MIO*). However for *B*(*T*), it only affects *TD*. To sum, Familiarity mainly affects S*P*, *TD* and *B*(*K*). To understand the behavioural changes of *B*(*T*), *B*(*M*) and *B*(*K*), Figs. 5, 6 and 7 demonstrate the responses of the three behaviours

Table 3 MANOVA tests of the between-subjects effects

Factor	Behaviour	Feature	*p*-value	Factor	Behaviour	Feature	*p*-value
Text length	SP		*0.0000*	Familiarity	SP		*0.0162*
	B(T)	TD	*0.0000*		B(T)	TD	*0.0221*
		Err	*0.0000*			Err	0.4684
		PA	*0.0001*			PA	0.2869
	B(M)	MS	*0.0267*		B(M)	MS	0.1461
		MID	*0.0000*			MID	0.0729
		MIO	*0.0000*			MIO	*0.0032*
		MCL	0.7752			MCL	0.7083
	B(K)	KS	0.5236		B(K)	KS	*0.0000*
		KL	0.5353			KL	*0.0003*
		EK	0.4164			EK	*0.0074*

Emphasised cell indicates that the effect is significant at the level of $p < 0.05$

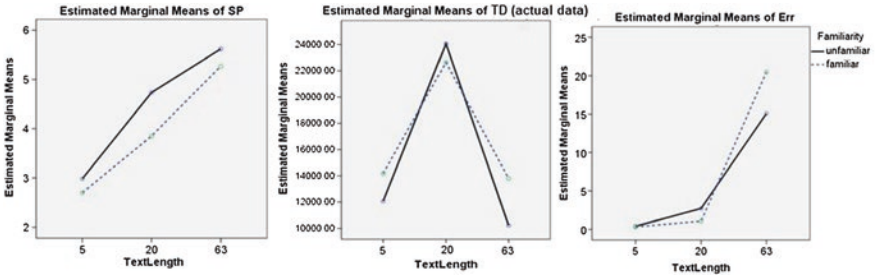


Fig. 5 Mean plots of task performance according to text length and familiarity

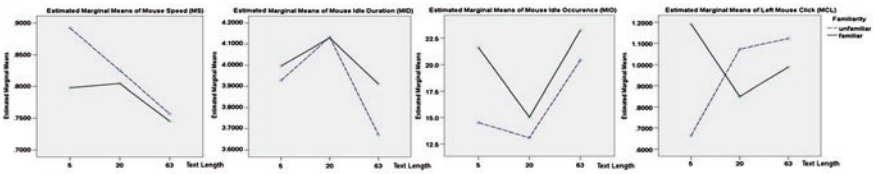


Fig. 6 Mean plots of mouse behaviour features according to text length (number of words) and familiarity. Note that *MCL* is not significantly affected by text length and familiarity

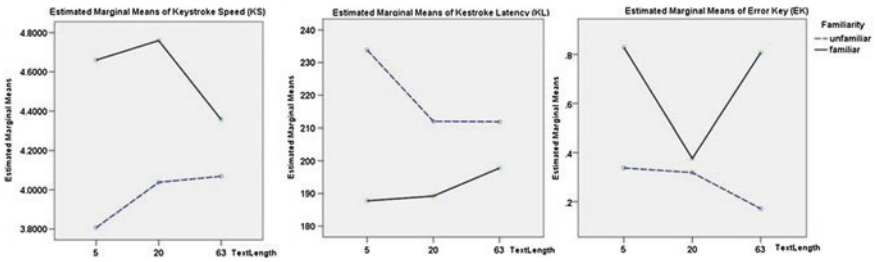


Fig. 7 Mean plots of keystroke behaviour features according to text length (number of words) and familiarity

according to Text Length and Familiarity. Note that *TD*, *MID*, and *MIO* demonstrate anomalous pattern with long text length. Nevertheless, by focusing only the short and medium length, we could observe that when Text Length increased, *TD*, *SP* and *MID* increased, but *MS* and *MIO* decreased (see Figs. 5 and 6). In Fig. 7, no significant changes of $B(K)$ can be observed when Text Length increased. However in terms of Familiarity, *KS* and *EK* are significantly lower but *KL* is higher when the student typed unfamiliar language. This explains that when users type the text written in unfamiliar language, their average *SP* would be higher (see Fig. 5), and *KL* would become higher while *KS* and *EK* become lower (see

Table 4 Correlation between features

	SP	TD	Err	MS	MID	MIO	MCL	KS	KL	EK
Question	0.0000	0.0001	0.0000		0.0151	*0.0002*				*0.0255*
Timing			0.0000			*0.0000*		*0.0004*	0.0003	
Length	0.0000	0.0000	0.0000		0.0025	*0.0090*				
Familiar.	*0.0294*		*0.0072*			0.0031		0.0000	*0.0001*	
SP		0.0000	0.0000		0.0000	*0.0025*		*0.0239*	0.0027	
TD	*0.0000*		0.0000		0.0000	*0.0216*	*0.0084*	*0.0469*	0.0000	0.0488
Err	0.0000	0.0000			0.0001	*0.0000*		*0.0032*	0.0042	
MS						*0.0020*				
MID	0.0000	0.0000	*0.0001*			*0.0011*	*0.0093*	*0.0497*	0.0000	0.0354
MIO	*0.0025*	*0.0216*	*0.0000*	*0.0020*	*0.0011*				*0.0300*	
MCL		*0.0084*			*0.0093*					
KS	*0.0239*	*0.0469*	*0.0032*		*0.0497*				*0.0000*	
KL	0.0027	0.0000	0.0042		0.0000	*0.0300*		*0.0000*		
EK		0.0488			0.0354					

Box without value indicates no significant correlation. Significant correlation exists between two features at $p < 0.05$ (2-tailed) level. Emphasised cell indicates negative correlation coefficient. *PA* is excluded due to insufficient data

Fig. 7). In terms of typing skill, the average completion duration per word dropped 62.35 % from Question 2 to Question 3, as well as an increase of 25.03 % of keystroke speed from Question 2 to Question 3, both signify increase of typing performance when Text Length increased (Fig. 3).

To further confirm the relationships among task demand, *SP*, *B*(*T*), *B*(*M*) and *B*(*K*), we perform Pearson Correlation Test. Due to the outliers given in Question 5 and Question 6, only samples for Question 1 to Question 4 are used. The results in Table 4 show that Text Length is correlated to *SP* and *B*(*M*), and Familiarity is correlated to *SP*, *B*(*T*) and *B*(*K*). Besides, there are correlations between *B*(*M*) and *B*(*K*) too.

5 Discussions

Our research studies the effects of task demand, which is varied by the combination of Text Length and language Familiarity (we refer it as Question) with time pressure (Timing) on student's stress perception (*SP*), task performance (*B*(*T*)), mouse behaviour (*B*(*M*)) and keystroke behaviour (*B*(*K*)). The following sections discuss the outcomes of the research.

5.1 The Effects of Task Demand on SP

Both Text Length and Familiarity produces significant effects on *SP*. The increment of Text Length or low Familiarity of the language may leads to higher *SP*. The students score highest *SP* when the Question is combined with long text and unfamiliar language. Timing has no effect on *SP* and it has no interaction effect with Question, which indicates that even the users are given time pressure, both groups of students have similar stress perceptions.

5.2 The Effects of Task Demand on B(T), B(M) and B(K)

Question The students demonstrate changes of behaviours in *B*(*T*), *B*(*M*) and *B*(*K*) from Question 1 to Question 6 except *MCL* (but *MCL* is affected by the interaction between Question and Timing). It is very interesting to note that the students started to cheat from Question 5 onwards, and this was further supported by evidence shown by anomalies in *B*(*T*), *B*(*M*) and *B*(*K*). Therefore Question 5 is considered a changeover point where the users have started losing motivation to continue the task. This can be explained by a few reasons, which include (1) high demand (of text length) that exceeds their estimated effort to complete the task, (2) time constraint and projected high *TD*, which reduce their estimated probability of

success, (3) the task is beyond their acceptable effort to invest, and (4) the aversiveness of the task cause fatigue and tiredness to the students at the end of the experiments.

Timing Timing has significant effects on all three behaviours. The number of students who made mistakes is higher when there is time pressure given. This suggests that more users will make more errors when job demand is higher, and it is worse if they are given time pressure. Interestingly the participants who are given time pressure type slower than those without time pressure. One possible reason is the results may be affected by other uncontrolled factors. Examples include (1) the students in the control group may experience other kind of time pressure incurred by external environment; (2) the students in the experimental group probably were not taking the experiments seriously; or (3) the students in the control group possessed better typing skills than those in the experimental group.

Text Length Text Length mainly affects $B(T)$ and $B(M)$ but not $B(K)$. Although it has no significant effect on $B(K)$, but the average completion duration per word by the users dropped and *KS* increased when the number of words increased from 5 to 20. Both suggest that the performance of task completion increased when Text Length increased. Generally, when Text Length increased, mouse speed and mouse idle occurrences decreased but mouse idle duration increased. This suggests that the users would move the mouse slower and less frequently.

Familiarity Familiarity mainly affects $B(T)$ and $B(K)$ but not $B(M)$. When the users are unfamiliar with the language, they tend to type slower. This shows that the users are not able to anticipate the upcoming words in unfamiliar language, such as German, and this affects their associated key presses. Despite that, there is no significant difference in terms of task duration. The possible reason is when the users are familiar with the language, the attempt to correct their typing errors before submission would increase (as use of error key increased). One of the reasons is the users can identify more errors when they are more familiar with the language. Another possible reason is that the capability of the browser (such as Google Chrome) that enables English spelling checking helps the users to spot spelling errors. Therefore it is important to switch off the capabilities of spelling and grammar checking before the experiment is conducted.

5.3 The Correlations Among Task Demand, SP, B(T), B(M) and B(K)

The correlation test results show that higher complexity of task (Question), such as longer Text Length and lower Familiarity would lead to higher *SP*. Some related changes in the behavioural patterns of $B(M)$ and $B(K)$ are observed. From the results, we suggest that when the students feel stressed due to higher task demand, it may lead to higher *TD*, *Err*, *MID* and *KL*, but lower *MIO*, *KS* and *EK*. In other words, this indicates that when task demand becomes higher, the students not only feel more stressed, but they use more time to complete the task (increased *TD*),

they move the mouse less frequently (increased MIO), type slower (increased KL and decreased KS), reduce the correction of error made (*EK* decreased), and therefore increase the number of task error (*Err* increased). Time pressure has no effects on *SP*, but those students who are given time pressure make more typing errors, move the mouse more frequently (lower *MIO*) and type slower (lower *KS* and higher *KL*). The correlation between *MCL* and *TD* also suggests that when task duration becomes longer, the use of mouse click has reduced. There are some correlations between *B*(*M*) and *B*(*K*), suggest the potential usefulness of unifying both methods in emotion detection. For instance, *KS* is correlated to *MID*, shows that when the users busy typing the text with higher speed, the mouse would be idle for shorter time. While *KL* is correlated to *MID* and *MIO*, suggest that when the users type the text slower (longer *KL*), mouse idle duration would increase, but mouse idle occurrences would reduce.

6 Conclusion

Generally, this research shows some useful information in stress detection. First, longer text length and lower familiarity of task increase stress perception. Higher stress due to high task demand generally results in longer task duration, higher error rate, slower mouse and keystroke speeds, longer mouse idle duration, and lower mouse idle occurrences and use of error key. This is consistent with the research by Lim et al. [13], which they also found that when mental arithmetic task demand increased, task error, task duration, passive attempt, stress perception and mouse idle duration increased, but mouse speed, left mouse click and keystroke speed decreased. Second, time pressure does not necessarily affect how users perceive stress but it affects task performance, mouse behaviour and keystroke behaviour. Time pressure leads to shorter task completion duration but higher error rate, and faster mouse movements but slower typing speed. Third, the correlations between mouse behaviour and keystroke behaviours suggest that the unification of both methods in emotion detection analysis could be potentially more useful than utilizing a single method alone. Fourth, language familiarity does affect task performance and keystroke behaviour. Text length changes mouse behaviour but not keystroke behaviour. This suggests that we should mainly look into task performance and mouse behaviour features if the typing tasks involve changes in length. For such, if task duration increased, and at the same time mouse idle duration increased but mouse speed and mouse idle occurrences decreased, while there are no changes in keystroke behaviour, then we could infer that the typing task demand has been increased. To assess how much a user is familiar with the task, we should look into task performance and keystroke behaviour features. If the user is familiar with the typing task (such as language), this should show increments of keystroke speed and use of error key, drop of key latency but no significant differences for other mouse features. Lastly, the projection of the emotional stress level based on the aforementioned behavioural responses according

to task demand is only valid as long as the students are still motivated to continue the task. Once the students have reached an ultimate stress point or have lost motivation, anomalous behaviours could occur. For instance, task duration and mouse idle duration started to drop while mouse idle occurrences increased at Question 5, although the users perceived even higher stress for Question 5 and Question 6. Therefore, once anomalous behaviours are detected, an adaptive content of e-learning could be activated, so that the students can be motivated to continue the tasks. The findings will be used as the basis of a computational algorithm for detecting user's emotional stress in our future research.

However, our research is not without limitations. First, the sample size is small, which only consists of 30 students in each group, therefore we may not be able to generalize the findings. More rigorous experiments need to be conducted to verify the stress model. Homogeneity cannot be assumed between questions due to outliers and anomalies in Question 5 and Question 6. Different individuals have high variations in typing skills, which could have affected the results. Lastly, the results may also be affected by external environmental factors, such as external time pressure, participants' mood, and the motivation of the participants to continue the experiments.

References

1. Sweller, J., Ayres, P., Kalyuga, S.: Cognitive Load Theory. Springer, Berlin (2011)
2. Paas, F., Renkl, A., Sweller, J.: Cognitive load theory: instructional implications of the interaction between information structures and cognitive architecture. Instr. Sci. **32**, 1–8 (2004)
3. Beilock, S.L., Ramirez, G.: On the interplay of emotion and cognitive control: implications for enhancing academic achievement. Psychol. Learn. Motiv. Res. Theor. **55**, 137 (2011)
4. Kirschner, P.A.: Cognitive load theory: implications of cognitive load theory on the design of learning. Learn. Instr. **12**, 1–10 (2002)
5. Anastasi, A.: Psychological Testing. Macmillan, UK (1954)
6. Karasek, R.A.: Job demands, job decision latitude and mental strain: implications for job design. Adm. Sci. Q. **24**, 285–308 (1979)
7. Rijk, A.E., Le Blanc, P.M., Schaufeli, W.B., Jonge, J.: Active coping and need for control as moderators of the job demand–control model: effects on burnout. J. Occup. Organ. Psychol. **71**, 1–18 (1998)
8. Wahlström, J., Hagberg, M., Johnson, P.W., Svensson, J., Rempel, D.: Influence of time pressure and verbal provocation on physiological and psychological reactions during work with a computer mouse. Eur. J. Appl. Physiol. **87**(3), 257–263 (2002)
9. Heiden, M., Lyskov, E., Djupsjbacka, M., Hellstrm, F., Crenshaw, A.G.: Effects of time pressure and precision demands during computer mouse work on muscle oxygenation and position sense. Eur. J. Appl. Physiol. **94**, 97–106 (2005)
10. Pusara, M., Brodley, C.E.: User re-authentication via mouse movements. In: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, pp. 1–8. ACM, New York (2004)
11. Tsoulouhas, G., Georgiou, D., Karakos, A.: Detection of learner's affective state based on mouse movements. J. Comput. **3**, 9–18 (2011)
12. Vizer, L.M.: Detecting cognitive and physical stress through typing behavior. In: Proceedings of 27th International Conference Extended Abstracts Human Factors Computer System CHI EA 09, p. 3113 (2009)

13. Lim, Y.M., Ayesh, A., Stacey, M.: Detecting cognitive stress from keyboard and mouse dynamics during mental arithmetic. In: Science and Information Conference 2014. pp. 146–152. IEEE Xplore, London (2014)
14. Setz, C., Arnrich, B., Schumm, J., La Marca, R., Troster, G., Ehlert, U.: Discriminating stress from cognitive load using a wearable EDA device. Inf. Technol. Biomed. IEEE Trans. **14**, 410–417 (2010)
15. Sloan, R.P., Korten, J.B., Myers, M.M.: Components of heart rate reactivity during mental arithmetic with and without speaking. Physiol. Behav. **50**, 1039–1045 (1991)
16. Khan, M.M., Ward, R.D., Ingleby, M.: Classifying pretended and evoked facial expressions of positive and negative affective states using infrared measurement of skin temperature. ACM Trans. Appl. Percept. **6**, 1–22 (2009)
17. Tobias, S., Abramson, T.: Interaction among anxiety, stress, response mode, and familiarity of subject matter on achievement from programmed instruction. J. Educ. Psychol. **62**, 357 (1971)
18. Hulme, C., Maughan, S., Brown, G.D.A.: Memory for familiar and unfamiliar words: evidence for a long-term memory contribution to short-term memory span. J. Mem. Lang. **30**, 685–701 (1991)
19. Davidson, J.E., Sternberg, R.J.: The Psychology of Problem Solving. Cambridge University press, Cambridge (2003)
20. Robinson, J.A., Liang, V.M.: Computer user verification using login string keystroke dynamics. IEEE Trans. Syst. Man Cybern. A Syst. Hum. **28**, 236–241 (1998)
21. Boechat, G.C., Ferreira, J.C., Carvalho Filho, E.: Authentication personal. In: International Conference on Intelligent and Advanced Systems, 2007 (ICIAS 2007), pp. 254–256 (2007)
22. Lv, H.-R., Lin, Z.-L., Yin, W.-J., Dong, J.: Emotion recognition based on pressure sensor keyboards. In: IEEE International Conference on Multimedia and Exposure 2008, pp. 1089–1092 (2008)
23. Teh, P.S., Yue, S., Teoh, A.B.J.: Improving keystroke dynamics authentication system via multiple feature fusion scheme. In: International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec). pp. 277–282 (2012)
24. Eswari, N., Sundarapandiyan, S., Vennila, P., Umamaheswari, R., Jothilakshmi, G.: Keystroke biometrics with number-pad input using hybridization of adaboost with random forest. In: International Conference on Advances in Engineering, Science and Management (ICAESM). pp. 105–109 (2012)
25. Giot, R., Rosenberger, C., Dorizz, B.: Can chronological information be used as a soft biometric in keystroke dynamics? In: Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), pp. 7–10 (2012)
26. Giot, R., El-Abed, M., Rosenberger, C.: Web-based benchmark for keystroke dynamics biometric systems: a statistical analysis. In: Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP). pp. 11–15 (2012)
27. Gunetti, D., Picardi, C.: Keystroke analysis of free text. ACM Trans. Inf. Syst. Secur. **8**, 312–347 (2005)
28. Shimshon, T., Moskovitch, R., Rokach, L., Elovici, Y.: Clustering di-graphs for continuously verifying users according to their typing patterns. 2010 IEEE 26th Convention of Electrical and Electronics Engineers in Israel (IEEEI), pp. 445–449 (2010)
29. Selye, H.: The Stress in Life. McGraw-Hill, New York (1956)
30. IBM: Multivariate General Linear Modeling. http://pic.dhe.ibm.com/infocenter/spssstat/v21r0m0/index.jsp?topic=/com.ibm.spss.statistics.cs/glmm_intro.htm

Friend Recommendation in a Social Bookmarking System: Design and Architecture Guidelines

Matteo Manca, Ludovico Boratto and Salvatore Carta

Abstract Social media systems allow users to share resources with the people connected to them. In order to handle the exponential growth of the content in these systems and of the amount of users that populate them, recommender systems have been introduced. As social media systems with different purposes arose, also different types of social recommender systems were developed in order to filter the specific information that each domain handles. A form of social media, known as *social bookmarking system*, allows to share bookmarks in a social network. A user adds as a friend or follows another user and receives updates on the bookmarks added by that user. In this paper, we present an analysis of the state-of-the-art on user recommendation in social environments and of the structure of a social bookmarking system, in order to derive design guidelines and an architecture of a friend recommender system in the social bookmarking domain. This study can be useful for future research, by highlighting the aspects that characterize this domain and the features that this type of recommender system has to offer.

Keywords Social bookmarking · Friend recommendation · User behavior · Tagging system

This work is partially funded by Regione Sardegna under project SocialGlue, through PIA—Pacchetti Integrati di Agevolazione "Industria Artigianato e Servizi" (annualità 2010).

M. Manca · L. Boratto (✉) · S. Carta
Dipartimento di Matematica e Informatica, Università di Cagliari,
Via Ospedale 72, 09124 Cagliari, Italy
e-mail: ludovico.boratto@unica.it

M. Manca
e-mail: matteo.manca@unica.it

S. Carta
e-mail: salvatore@unica.it

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_14

1 Introduction

Social media systems are web-based services that allow users to build a public or semi-public profile, create a list of other users with whom they share a connection, and view and traverse their list of connections and those made by others within the system [5]. The widely-known and studied information overload problem, in these system took the name of "social interaction overload" [13, 27], which means that each user has to interact with an excessive amount of users and items. This leads to a scarcity of attention, which does not allow a user to focus on users or items that might be interesting for her/him. In order to face the social information overload problem, recommender systems have been adopted to filter the large amount of information available in the social domain; the class of recommender systems that operate in the social domain is known as *social recommender systems* [25]. These systems face the social interaction overload problem, by suggesting users or items that a target user might be interested in. In particular, user recommendation in a social domain aims at suggesting *friends* (i.e., recommendations are built for pairs of users that are likely to be interested in each other's content) or *people to follow* (i.e., recommendations are built for a user, in order to suggest users that might be interesting for her/him) [13].

User recommender systems that operate in the social media domain can be classified into three categories, based on the source of data used to build the recommendations:

1. Systems based on the analysis of social graphs, which explore the set of people connected to the target user in order to produce recommendations. These systems recommend either the closest users in the graph, like friends of friends and followees of followees (the "People you may know" feature offered by Facebook [24] is the most widely known example of this approach), or recommend the users that have the highest probability to be crossed in a random walk of the social graph (the main reference for this type of systems is the "Who to follow" recommendation in Twitter [12]).
2. Systems that analyze the interactions of the users with the content of the system (tags, likes, shares, posts, etc.). In order to exploit the user interests, these systems usually build a user profile by giving a structured form to content, thanks to the use of metrics like TF-IDF (Term Frequency—Inverse Document Frequency). Recommendations are produced by identifying users with similar profiles. An example of this class of systems is presented in Chen et al. [10].
3. Hybrid systems, that consider both the social graph and the interactions of the users with the content (an example is represented by [15]).

A *social bookmarking system* is a form of social media, which allows users to use keywords (*tags*) to describe resources that are of interest for them, helping to organize and share these resources with other users in the network [11]. The most widely-known examples of social bookmarking systems are Delicious,[1] where the

[1] http://www.delicious.com.

bookmarked resources are web pages, CiteULike,[2] where users bookmark academic papers, and Flickr,[3] where each picture can be annotated with tags.

Even if the use of these systems is widespread (in 2014, one million photos per day have been shared on Flickr[4]), to the best of the authors' knowledge, no approach in the literature recommended friends in a social bookmarking system prior to our recent works [20, 21].

In this paper we present a study that proposes the design and the definition of an architecture of a friend recommender system in a social bookmarking system. By analyzing the state-of-the-art on user recommendation in the social domain and how social bookmarking systems work, we design a friend recommender system that operates in this context and present its architecture.

The scientific contributions coming from this paper are the following:

- we analyze the state-of-the-art on user recommendation in social bookmarking systems, in order to highlight the weaknesses of the existing systems and derive the characteristics and features that a friend recommender system that operates in this domain has to offer;
- given the structure of a social bookmarking system and the analysis of the state-of-the-art, we present a design of a friend recommender system;
- we propose a novel architecture of a system to build friend recommendations in a social bookmarking system.

This paper extends the work presented in Manca et al. [20] in the following ways:

- a deeper contextualization with the state-of-the-art is going to be presented;
- the motivation to our study is going to be improved, by presenting an analysis of how our design guidelines relate to a real-world scenario. This will help us validate our study and introduce the architecture;
- an extension to the proposed architecture is provided, by presenting it at different granularities and by providing more details on each component. Moreover, we are going to analyze possible approaches to implement it in a real-world system and present possible extensions to it.

This study can be useful for any future research in this area, by presenting design guidelines and an architecture, which can be adopted in the development of a friend recommender system in the social bookmarking domain.

The rest of the paper is structured as follows: Sect. 2 presents the state-of-the-art on user recommendation in social environments; Sect. 3 illustrates how a social bookmarking system is structured and how it works; Sect. 4 presents the aspects related to the design of a friend recommender system in a social bookmarking system and presents guidelines, useful in the development of a system; Sect. 5 proposes an architecture of the system; Sect. 6 presents conclusions and future work.

[2] http://www.citeulike.org/.

[3] http://www.flickr.com/.

[4] http://techcrunch.com/2014/02/10/flickr-at-10-1m-photos-shared-per-day-170-increase-since-making-1tb-free/.

2 Related Work

In the last years, social bookmarking systems have been studied from different points of view. This section presents related work on user recommendation in this research area. This study of the state-of-the-art will be deepened in Sect. 4, in order to analyze the aspects that characterize a recommender system that operates in this domain and the weaknesses of the existing approaches.

2.1 Systems Based on the Analysis of Social Graphs

In [12] authors present Twitter's user recommendation service, which allows to daily create a huge amount of connections between users that share common interests, connections and other factors. In order to perform the recommendations, the authors build a Twitter graph in which vertices represent users and the directed edges represent the "follow" relationship. The graph is stored in a graph database called FlockDB, and then data are processed with Cassovary (an open source in-memory graph processing engine). The system builds the recommendations by means of a user recommendation algorithm for directed graphs based on SALSA. In the next section, we are going to analyze this system, in order to design our proposal.

In [17] the authors model the user recommendation problem as a link prediction problem. They develop several approaches, that analyze the proximity of nodes in the graph of a social network, in order to infer the probability of new connections among users. Experiments show that the network topology is a good tool to predict future interactions.

In [2], Arru et al. propose a user recommender system for Twitter, based on signal processing techniques. The considered approach defines a pattern-based similarity function among users and makes use of a time dimension in the representation of the users profile. Our system is different, because we aim at suggesting friends while on Twitter there is no notion of "friend" but it works with "people to follow".

2.2 Systems Based on the Interactions with the Content

Quercia et al. [23] describe a user recommender system based on collocation. The proposed framework, called FriendSensing, recommends friends by analyzing collocation data. In order to produce the recommendations, the system uses geographical proximity and link prediction theories. In our domain we do not have such a type of information, so we cannot compare with this algorithm.

In [8], researchers present a study that considers different features in a user profile, behavior and network in order to explore the effect of *homophily* on user recommendations. They use the Dice coefficient on two users sets of tags and they find that similar tags do not represent a useful source of information for link prediction, while mutual followers are more useful for this purpose. As previously

highlighted, the presented friend recommender system focuses on producing friend recommendation based on users' content (tag, bookmarks, etc.).

2.3 Hybrid Systems

In [29] authors propose a framework of user recommendation, based on users' interests and tested on Yahoo! Delicious. The proposed framework operates in two main steps: first, it models the users' interests by means of tag graph based community detection and represents them with a discrete topic distribution; then, it uses the Kullback-Leibler divergence function to compute the similarity between users' topic distribution and the similarity values are used to produce interest based user recommendation. Differently from this framework, the aim of the approach proposed in this paper is to produce friend recommendations (i.e., bidirectional connections) and not unidirectional user recommendations.

Chen et al. [10] present a people recommender system in an enterprise social network called Beehive, designed to help users to find known, offline contacts and discover new friends on social networking sites. With the proposed study, the authors demonstrate that algorithms that use similarity of user-created content were stronger in discovering new friends, while algorithms based on social network information were able to produce better recommendations.

In [15], the authors propose a user recommender system (called *Twittomender*) that, for each user, builds a user profile based on user's recent Twitter activity and user's social graph. The proposed system operates in two different manners; in the former mode the user puts a query and the system retrieves a ranking list of users, while in the latter mode the query is automatically generated by the system and it is mined by the user profile of the target user (the target user is the user that receives the recommendations). Our proposal does not use the social graph and, furthermore, in building recommendations it considers the friendship relationship and not the "user to follow" relationship.

In [14] authors present a recommender system for the IBM Fringe social network, based on aggregated enterprise information (like org chart relationships, paper and patent co-authorship, project co-membership, etc.) retrieved using SONAR, which is a system that allows to collect and aggregate these kinds of information. The authors deployed the people recommender system as a feature of the social network site and the results showed a highly significant impact on the number of connections on the site, as well as on the number of users who invite others to connect.

3 Social Bookmarking Systems

This section presents how a social bookmarking system is structured and how it works. This definition is based on the ones previously given in the literature (in particular we refer to [11, 16, 26]).

A social bookmarking system is composed by:

- a set of *users*;
- a set of *resources*. These resources characterize the type of social bookmarking system and, as mentioned in the introduction, they might be of different types (e.g., web pages);
- a set of *tags*, which are the keywords used to describe the resources;
- a set of *bookmarks*, which are represented as triplets (*user*, *resource*, *tag*); these triplets are known either as *tag assignments*, or as *tag applications*;
- a set of *connections* among users, which are represented as couples (*user*, *user*). Depending on the type of connection among two users, a couple might be ordered (i.e., users are connected by a *follow* relation), or not (i.e., users are *friends* and mutually follow each other). These connections form a graph, known either as *social graph* or *interest graph*.

Once a user decides to bookmark a resource by adding tags to it, these bookmarks are shown to the users who are friends with or follow this user.

Social bookmarking systems also offer privacy options, which allow to keep a bookmark private, or to share it only with a limited amount of users.

Features that allow to explore the tags and to facilitate the management of the bookmarks, like their export from browsers [19] and the possibility to add a bookmark to the profile by email, are often offered.

4 Designing a Friend Recommender System

The first objective of our proposal is to design a friend recommender system in a social bookmarking system. This section presents an analysis of the aspects that characterize both the state-of-the-art and social bookmarking systems, according to what was presented in the previous sections.

4.1 Analysis

In our analysis, we considered the following aspects:

(a) In [12], authors highlight that Twitter is an "interest graph", rather than a "social graph". A problem highlighted by the authors is that the analysis of such a graph suffers from scalability issues and, in order to contain the complexity of the recommender system, no user profile information could be used to build the recommendations. The definition of interest graph can also be extended to social bookmarking systems, since a user can add as a friend or follow another user, in order to receive her/his newly added bookmarks.

(b) Social media systems grow rapidly. This means that both the amount of content added to a social media system and the user population increase at a fast

rate. A recommender system that operates in this context needs to build accurate user profiles, which are up-to-date with the constantly evolving preferences of the users.

(c) Resources usually have an unstructured form so, when building a content-based recommender systems, they are given a structured form, by introducing a *Content Analyzer* in the system [18].

(d) In the architecture of a content-based system, a *Feedback* component, which allows to update the user profile according to the recommended items that the user liked or did not like, is usually implemented [18].

(e) As [29] highlights, the tagging activity of the users reflects their interests. Therefore, the tags used by a user can be considered as an important source of information to exploit her/his interests.

Taking into account all these aspects, we drew the following conclusions.

Regarding point (a), in order to avoid the limitations related to the graph analysis in this domain, we aim at designing a system that only analyzes the content of the users (i.e., the tagged resources). So, we are going to design a system that belongs to the second class presented in the Introduction, i.e., the one that analyzes the interactions of the users with the content of the system.

Regarding points (b) and (c), given the rapid growth of information in social media systems, in order to efficiently and quickly update user profiles we decided to exploit the set of resources used by each user and the tags used to classify those resources, without using a *Content Analyzer* component, but analyzing only the *behavior of the users in the system*.

Regarding point (d), since the system we are designing deals with friend recommendations and we do not consider the connection between the users, the feedback of a user has no impact in her/his profile. On the contrary, when items are recommended in a content-based system, the feedbacks contain information about the preferences of the users, which help updating the user profiles.

Regarding point (e), we embraced the theory that user interests are reflected by the tagging activity and extended it, by following the intuition that users with similar interests use similar tags and the same resources.

4.2 Design Guidelines

Starting from the previous analysis, here we recap the features that a friend recommender system in the social bookmarking domain has to offer:

1. the resources saved by a user and the tags used to classify them represent a valuable source of information about a user. By monitoring them, we can constantly be updated on the interests of the users. Therefore, a friend recommender system in a social bookmarking system has to consider the tagged resources bookmarked by the users. Using only graph analysis to build the recommendations presents limitations, and building recommendations by

analyzing both the content the users interacts with and the interest graph would increase the complexity of the system (this might lead to the learning of user profiles that are not up-to-date with the current interests of the users);
2. the algorithms and metrics used by a system should be quickly computed, in order to keep the user profiles up-to-date. Therefore, we believe that a friend recommender system should mine *user behavior* (i.e., the interaction of the users with the content), more than the content itself. In fact, the introduction of a *Content Analyzer*, in order to give a structured form to the resources, would significantly increase the complexity of the system. In other words, it is harder to make an analysis of the content of each resource tagged by a user, instead of considering only the fact that a user is interested in that resource. Since social bookmarking systems grow at a fast rate, content analysis would lead to have outdated profiles and this component is discarded by our design and architecture;
3. in order to reduce the complexity of the system, and given the type of recommendations produced, the typical *Feedback* component of a Content-Based recommender system is removed when designing such a type of system. This choice was made since the accepted or rejected friends do not update the user profiles, which are built considering the tag assignments of the users;
4. in order to capture the interaction of the users on multiple levels and improve the capability to accurately recommend friends, a system has to be able to exploit all the sources of information coming from the tag assignments. Therefore, a friend recommender system has to analyze both the tags used by a user and the resources she/he bookmarked.

4.3 Design Guidelines Evaluation in a Real-World Scenario

In the following, an analysis of the user behavior in a social bookmarking system from a friend recommendation point of view is presented. In particular, how the bookmarking activity of a user is related to that of the others has been studied by analyzing a Delicious dataset, distributed for the HetRec 2011 workshop [9]. The dataset contains:

- 1,867 users;
- 69,226 URLs;
- 53,388 tags;
- 7,668 bi-directional user relations;
- 437,593 tag assignments [i.e., tuples (user, tag, URL)];
- 104,799 bookmarks [i.e., distinct pairs (user, URL)].

By analyzing user profiles, it emerges that users had an average of 123.697 tags, and an average of 56.132 bookmarked resources.

In order to be able to infer the possible connections among users, which might lead to friend recommendations, the number of common tags and resources

between the users of the dataset have been computed, obtaining the following results: the average number of common tags among two users is 7.807, while the average number of common resources among two users is 0.042. In particular, considering only the users who have at least a common tag, the average number of common tags for a couple of users increases to 10.417; while considering only the users who have at least a common bookmarked resource, the average number of common resources for each couple of users increases to 1.673.

From the conducted analysis is possible to infer some properties related to the user behavior in a social bookmarking system, recapped below:

- the behavior of two users in a social bookmarking system is related both to the use of the tags and to the use of the resources;
- the use of tags represents a stronger form of connection (as also proved in the literature), with respect to the amount of common resources between two users. This happens because the probability that two users use the same tags is higher than the one to bookmark the same resource, since a user classifies a resource with more tags;
- by comparing the number of common tags and resources with respect to the number of all tags and resources, it emerges that the number of common tags and common resources is much smaller than the number of tags and resources used by each user (more precisely, 10.4 out of 123.7 tags, and 1.7 out of 56.1 resources).

This means that the behavioral analysis of a user, which characterized the design of the system, can be exploited in order to recommend friends in this domain. Therefore, following these guidelines, in the next section we are going to present a novel architecture to build friend recommendations by exploiting the behavior of the users in a social bookmarking system.

5 Architecture

In order to build an architecture for a friend recommender system in the social bookmarking domain, we are going to follow the design guidelines presented in the previous section. Figure 1 illustrates the high level view of the architecture.

While designing the system, in the first point of the guidelines we highlighted that we would only analyze the content of the system (i.e., the tag assignments). Therefore, the architecture does not have components that analyze the connections among users (i.e., who they follow or they are friends with).

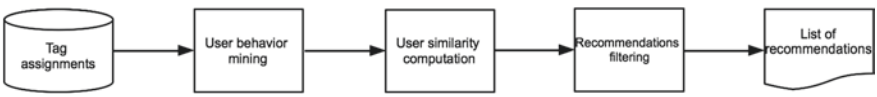


Fig. 1 High level architecture of the friend recommender system

The first task that the system has to compute is the mining of the user behavior by exploiting the tag assignments of each user (i.e., which resources a user tagged and with which tags). The *User behavior mining* component will allow to create a profile with the preferences of each user. Once the behavior of the user has been mined, a *User similarity computation* component will measure the similarity between the users. These similarities will then be inspected by the *Recommendation filtering* component, which will select the users most similar to each user, in order to recommend them.

The rest of the Section will provide the details of each high level component previously presented.

5.1 User Behavior Mining

As Fig. 2 shows, user behavior can be mined by two different components (i.e., the *Tag-based profile learner* and the *Resource-based profile learner*), following the considerations done on the fourth point of the design guidelines, which suggested to consider both the tags and the resources available in the bookmarks.

Taken as input the *Tag assignments* available for each user, two profile learner components will analyze the behavior of the user of her/his use of the tags (*Tag-based profile learner*) and on the bookmarked resources (*Resource-based profile learner*). Each component will now be presented in detail.

5.1.1 Tag-Based Profile Learner

Each time a user classifies a resource with a tag, her/his profile should be updated in order to capture the tagging behavior and build an accurate user profile. Taken as input the *Tag assignments* available for each user, this component builds a user profile, by considering the tags used by a user. Since in the design guidelines we highlighted the need to build profiles quickly, in order for them to be updated, this component might build profiles as binary vectors of the tags considered by users, or by considering the frequency of each tag used by a user. The output produced is a *Tag-based user profile*.

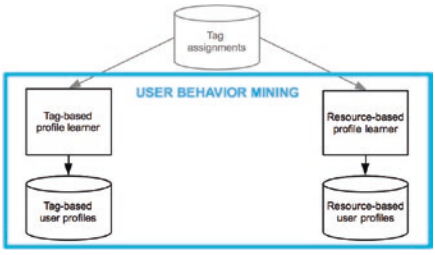


Fig. 2 Part of the architecture that mines user behavior to build the user profiles

5.1.2 Resource-Based Profile Learner

Given the *Tag assignments*, this component builds a second user profile, by analyzing the resources bookmarked by a user. Also this profile might be built as a vector, similarly to the possible implementations of the tag-based component. In case a binary vector is produced, it could highlight which resources have been bookmarked by the user and which not. Another possible implementation of this component would be by building a vector that contains in each element associated to a resource how many tags have been used to classify that resource (i.e., the relevance of a resource for a user could be measured by her/his effort to classify it, and a counter would keep track of this type of behavior). The output produced by this component is a *Resource-based user profile*.

5.2 User Similarity Computation

Figure 3 shows the two components that compute the similarities between the users, by comparing the tag-based user profiles (*Tag-based similarity computation* component) and the resource-based user profiles (*Resource-based similarity computation* component). This part of the architecture will now be described in detail.

5.2.1 Tag-Based Profile Similarity Computation

Given the *Tag-based user profiles* previously computed, this step estimates the association among each couple of tag-based user profiles, in order to derive how similar two users are. In case the similarity between binary vectors has to be computed, the Jaccard index would represent a standard measure to capture this similarity and efficient algorithms with low computational complexity have been proposed in the literature (e.g., the MinHash scheme [7], or the Signature scheme [1]). In case a vector with positive values is used to represent the profile, Pearson's correlation coefficient [22] as proved to be the most effective for the similarity assessment

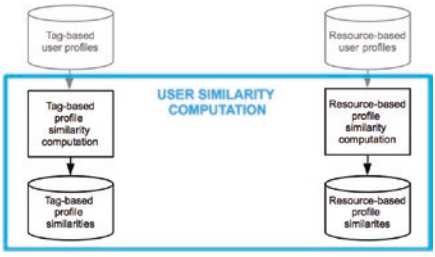


Fig. 3 Part of the architecture that computes the user similarities

between users [6]. Moreover, an efficient algorithm that exploits a support-based upper bound exists [28]. The output produced is a *Tag-based profile similarity*.

5.2.2 Resource-Based Profile Similarity Computation

Given the *Resource-based user profiles* previously computed, this step estimates the association among each couple of resource-based user profiles, in order to derive how similar two users are. According to the representation of the resource-based profile (i.e., binary vector or vector with positive values), the same algorithms used by the tag-based association component can be exploited. The output produced by this component is a *Resource-based profile similarity*.

5.3 Recommendations Filtering

This part of the architecture (shown in Fig. 4) solves the task of combining the tag-based and resource-based similarities between the users and filter them in order to produce the friend recommendations. Given the *Tag-based profile similarities* and *Resource-based profile similarities* previously built, the *Filtering component* selects the most similar users to recommend to the target user. For example, a threshold value might be used, in order to select only the users with high similarities with the target user. The output is a ranked *List of recommendations*, which contains the users to recommend to the target user.

5.4 System Architecture and Discussion

The full architecture of the system is presented in Fig. 5. Based on this structure and on the possible implementations previously presented, an efficient friend recommender system in the social bookmarking domain can be built.

Considering the social environment in which the recommendations have to be produced, an interesting aspect to notice in this architecture is that it lends itself

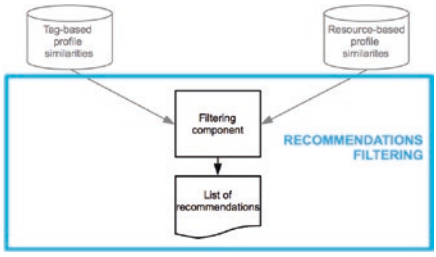


Fig. 4 Part of the architecture that produces the recommendations

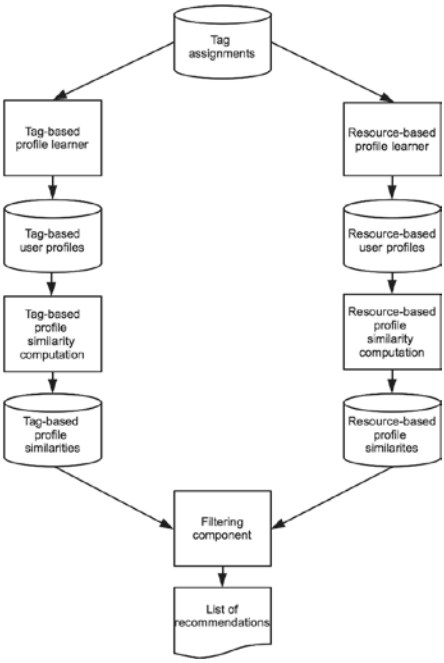


Fig. 5 Architecture of the friend recommender system

well to a parallelization on multiple machines. In fact, the two branches computed by the system (i.e., the one that operates with the tags and with the resources) can be independently computed.

Moreover, our architecture can be easily extended in case a new type of user behavior has to be mined. Suppose for example that we want to estimate the interest of the user on the topics of the resources.[5] A third branch could be added to this architecture, and this confirms that the proposed architecture can be adopted to build scalable systems. Given the possibility to extend our architecture to different types of behaviors to mine, this architecture can be used also to produce friend recommendations in different types of social media systems, by following the same pattern.

[5] Given that traditional techniques to manually categorize data cannot be applied in social environments [4] and that clustering techniques represent a good form to extract information for recommendation purposes [3], the resources could be clustered based on the tags used to classify them, in order to extract some meta-information about a group of resources related to a specific topic.

6 Conclusions and Future Work

This paper illustrated a study related to the design and the architecture of a friend recommender system in the social bookmarking domain. We analyzed the existing state-of-the-art works that recommend users in social domain and illustrated the structure of a social bookmarking system. This led to the design of a system that recommends friends in this context. After giving the design guidelines, the architecture of the system was presented. Following these design guidelines and this architecture, we built an efficient and very accurate friend recommender system [21], tested on the Delicious dataset previously illustrated. As future work we will implement the extension to the architecture proposed in the Discussion (i.e., the analysis of the topics) and the test it in our system.

References

1. Arasu, A., Ganti, V., Kaushik, R.: Efficient exact set-similarity joins. In: Proceedings of the 32nd International Conference on Very Large Data Bases. pp. 918–929. VLDB' 06, VLDB Endowment (2006). http://dl.acm.org/citation.cfm?id=1182635.1164206
2. Arru, G., Gurini, D.F., Gasparetti, F., Micarelli, A., Sansonetti, G.: Signal-based user recommendation on twitter. In: Carr, L., Laender, A.H.F., Lóscio, B.F., King, I., Fontoura, M., Vrandecic, D., Aroyo, L., de Oliveira, J.P.M., Lima, F., Wilde, E. (eds.) 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, 13–17 May 2013, Companion Volume. pp. 941–944. International World Wide Web Conferences Steering Committee/ACM (2013)
3. Boratto, L., Carta, S., Manca, M., Mulas, F., Pilloni, P., Pinna, G., Vargiu, E.: A clustering approach for tag recommendation in social environments. Int. J. e-Bus. Dev. **3**, 126–136 (2013)
4. Boratto, L., Carta, S., Vargiu, E.: Ratc: A robust automated tag clustering technique. In: Noia, T.D., Buccafurri, F. (eds.) e-Commerce and Web Technologies. In: 10th International Conference, EC-Web 2009, Linz, Austria, 1–4 Sept 2009. Proceedings of Lecture Notes in Computer Science, vol. 5692, pp. 324–335. Springer (2009)
5. Boyd, D.M., Ellison, N.B.: Social network sites: Definition, history, and scholarship. J. Comput.-Mediated Commun. **13**(1), 210–230 (2007)
6. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. pp. 43–52. UAI'98, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998). http://dl.acm.org/citation.cfm?id=2074094.2074100
7. Broder, A.: On the resemblance and containment of documents. In: Proceedings of the Compression and Complexity of Sequences 1997. pp. 21 SEQUENCES '97, IEEE Computer Society, Washington, DC, USA (1997). http://dl.acm.org/citation.cfm?id=829502.830043
8. Brzozowski, M.J., Romero, D.M.: Who should i follow? Recommending people in directed social networks. In: Adamic, L.A., Baeza-Yates, R.A., Counts, S. (eds.) Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, 17–21 July 2011. The AAAI Press (2011)
9. Cantador, I., Brusilovsky, P., Kuflik, T.: Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011). In: Mobasher, B., Burke, R.D., Jannach, D., Adomavicius, G. (eds.) Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, 23–27 Oct 2011. pp. 387–388. ACM (2011)

10. Chen, J., Geyer, W., Dugan, C., Muller, M.J., Guy, I.: Make new friends, but keep the old: recommending people on social networking sites. In: Jr., DRO, Arthur, R.B., Hinckley, K., Morris, M.R., Hudson, S.E., Greenberg, S. (eds.) Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, 4–9 April 2009. pp. 201–210. ACM (2009)
11. Farooq, U., Kannampallil, T.G., Song, Y., Ganoe, C.H., Carroll, J.M., Giles, C.L.: Evaluating tagging behavior in social bookmarking systems: metrics and design heuristics. In: Gross, T., Inkpen, K. (eds.) Proceedings of the 2007 International ACM SIGGROUP Conference on Supporting Group Work, GROUP 2007, Sanibel Island, Florida, USA, 4–7 Nov 2007. pp. 351–360. ACM (2007)
12. Gupta, P., Goel, A., Lin, J., Sharma, A., Wang, D., Zadeh, R.: Wtf: the who to follow service at twitter. In: Schwabe, D., Almeida, V.A.F., Glaser, H., Baeza-Yates, R.A., Moon, S.B. (eds.) 22nd International World Wide Web Conference, WWW'13, Rio de Janeiro, Brazil, 13–17 May 2013. pp. 505–514. International World Wide Web Conferences Steering Committee/ACM (2013)
13. Guy, I., Chen, L., Zhou, M.X.: Introduction to the special section on social recommender systems. ACM TIST **4**(1), 7 (2013)
14. Guy, I., Ronen, I., Wilcox, E.: Do you know?: recommending people to invite into your social network. In: Conati, C., Bauer, M., Oliver, N., Weld, D.S. (eds.) In: Proceedings of the 2009 International Conference on Intelligent User Interfaces, 8–11 Feb 2009, Sanibel Island, Florida, USA. pp. 77–86. ACM (2009)
15. Hannon, J., Bennett, M., Smyth, B.: Recommending twitter users to follow using content and collaborative filtering approaches. In: Amatriain, X., Torrens, M., Resnick, P., Zanker, M. (eds.) Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, 26–30 Sept 2010. pp. 199–206. ACM (2010)
16. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Bibsonomy: a social bookmark and publication sharing system. In: Proceedings of the First Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures. pp. 87–102 (2006)
17. Liben-Nowell, D., Kleinberg, J.M.: The link prediction problem for social networks. In: Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management, New Orleans, Louisiana, USA, 2–8 Nov 2003. pp. 556–559. ACM (2003)
18. Lops, P., de Gemmis, M., Semeraro, G.: Content-based recommender systems: state of the art and trends. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 73–105. Springer (2011)
19. Lund, B., Hammond, T., Hannay, T., Flack, M.: Social bookmarking tools (ii): a case study—connotea. D-Lib Magazine 11(4) (2005)
20. Manca, M., Boratto, L., Carta, S.: Design and architecture of a friend recommender system in the social bookmarking domain. In: Proceedings of the Science and Information Conference 2014. pp. 838–842 (2014)
21. Manca, M., Boratto, L., Carta, S.: Mining user behavior in a social bookmarking system—a delicious friend recommender system. In: Proceedings of the 3rd International Conference on Data Management Technologies and Applications (DATA 2014). pp. 331–338 (2014)
22. Pearson, K.: Mathematical contributions to the theory of evolution. iii. Regression, heredity and panmixia. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Math. or Phys. Character (1896–1934) 187, 253–318 (1896)
23. Quercia, D., Capra, L.: Friendsensing: recommending friends using mobile phones. In: Bergman, L.D., Tuzhilin, A., Burke, R.D., Felfernig, A., Schmidt-Thieme, L. (eds.) Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, 23–25 Oct 2009. pp. 273–276. ACM (2009)
24. Ratiu, F.: Facebook: people you may know (May 2008), https://blog.facebook.com/blog.php?post=15610312130

25. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 1–35. Springer, Berlin (2011)
26. Sen, S., Lam, S.K., Rashid, A.M., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F.M., Riedl, J.: Tagging, communities, vocabulary, evolution. In: Hinds, P.J., Martin, D. (eds.) Proceedings of the 2006 ACM Conference on Computer Supported Cooperative Work, CSCW 2006, Banff, Alberta, Canada, 4–8 Nov 2006. pp. 181–190. ACM (2006)
27. Simon, H.A.: Designing organizations for an information rich world. In: Greenberger, M. (ed.) Computers, Communications, and the Public Interest, pp. 37–72. Johns Hopkins Press, Baltimore (1971)
28. Xiong, H., Shekhar, S., Tan, P.N., Kumar, V.: Exploiting a support-based upper bound of pearson's correlation coefficient for efficiently identifying strongly correlated pairs. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 334–343. KDD '04, ACM, New York, NY, USA (2004). http://doi.acm.org/10.1145/1014052.1014090
29. Zhou, T.C., Ma, H., Lyu, M.R., King, I.: Userrec: a user recommendation framework in social tagging systems. In: Fox, M., Poole, D. (eds.) Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, 11–15 July 2010. AAAI Press (2010)

Quantum Behaved Genetic Algorithm: Constraints-Handling and GPU Computing

Amgad M. Mohammed, N.A. Elhefnawy, Mahmoud M. El-Sherbiny and Mohiy M. Hadhoud

Abstract Quantum-inspired evolutionary algorithm is a new evolutionary algorithm using concepts and principles of quantum computing to work on *classical computer* rather than quantum mechanical hardware. This article introduces main concepts behind the intersection between evolutionary algorithms and quantum computing, such as quantum-bit, superposition feature, quantum gate, quantum measurement and quantum interference. These behaviors of quantum concepts offer computational power and computational intelligence that must be harnessed and used. Intelligence is the main focus to design novel constraint-handling technique with quantum behaved genetic algorithm (QBGA) to solve well known constrained benchmark problems. Single quantum chromosome represents multiple solutions at the same time, so the same infeasible solutions based on quantum features are also feasible ones. Finally GPU (Graphics Processing Unit) will be discussed with (QBGA) to achieve parallel processing and speed up execution time, especially to solve high dimensional real world optimization problems requiring intensive computing resources.

Keywords Constraint-Handling techniques · Quantum inspired evolutionary algorithms · Quantum computing · GPU computing · Nonlinear optimization · Parallel processing

A.M. Mohammed (✉) · N.A. Elhefnawy
Department of Operations Research, Menofia University, Menofia, Egypt
e-mail: amgad.elsayed@ci.menofia.edu.eg

N.A. Elhefnawy
e-mail: nancyabbas_1@hotmail.com

M.M. El-Sherbiny
Department of Operations Research, Institute of Statistical Studies and Research, Giza, Egypt
e-mail: msherbiny@cu.edu.eg

M.M. Hadhoud
Department of Information Technology, Menofia University, Menofia, Egypt
e-mail: mmhadhoud@yahoo.com

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_15

1 Introduction

A quantum computer is a computation device that makes direct use of quantum-mechanical phenomena, such as superposition and interference, to perform operations on data [1, 2]. Quantum computer harness the power of atoms and molecules to be the computer's processor and memory. It is not easy to build quantum computer and process data on it, but all concepts of quantum can be simulated on classical computers to benefit from probabilistic and power computing.

The interpolation between quantum computing and evolutionary algorithms can be classified to main categories.

- *Quantum evolutionary algorithms*: related to implementing evolutionary algorithms over quantum computation environment [3–5].
- *Quantum-inspired evolutionary algorithms* (*QIEA*): focus on developing new evolutionary algorithms based on concepts and principles of quantum computing to work on *classical computer* paradigm [6–8].

1.1 Quantum-Bit (Q-Bit)

In contrast to classical computers that use binary digits (bits), quantum computers use q-bits. A single q-bit $|\psi\rangle$ can be represented as a linear sum of the basis states.

$$|\boldsymbol{\psi}\rangle = \boldsymbol{\alpha}|0 + \boldsymbol{\beta}|1\rangle$$

Where α & β are the probability amplitudes of the corresponding states. The values $|\alpha|^2$ & $|\beta|^2$ denote the probabilities that the q-bit will be found in "0" or "1" state respectively. The normalization condition for every q-bit must be satisfied, where

$$|\boldsymbol{\alpha}|^2 + |\boldsymbol{\beta}|^2 = \mathbf{1}$$

Figure 1 shows how unit circle represents two different q-bits.

Where $\alpha = \cos\theta$, $\beta = \sin\theta$ & $\theta = [0{:}2\pi]$. All data must be represented in the form of q-bits to be processed.

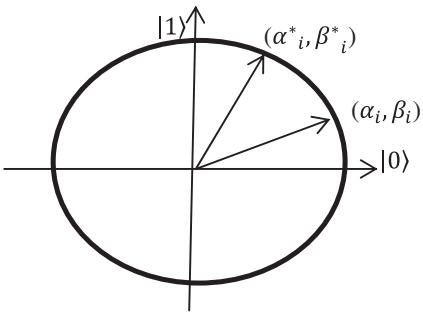


Fig. 1 Polar plot for two different q-bits

1.2 Quantum Superposition

Quantum system $|\psi_n\rangle$ with n q-bits can be found in 2^n states simultaneously [9, 10], but will represent single state after collapsing (observing).

$$|\psi_n\rangle = \sum_{j=1}^{2^n} C_j |S_j\rangle \tag{1}$$

Where C_j is the probability amplitude of the jth state S_j. Example consider a three q-bit system $|\psi_3\rangle$

$$|\psi_3\rangle = \left[\begin{array}{c|c|c} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \end{array}\right] = \left[\begin{array}{c|c|c} \frac{\sqrt{3}}{3} & \frac{\sqrt{1}}{3} & \frac{\sqrt{2}}{3} \\ \frac{\sqrt{6}}{3} & \frac{\sqrt{8}}{3} & \frac{-\sqrt{7}}{3} \end{array}\right]$$

Where $|\alpha_i|^2 + |\beta_i|^2 = 1, \quad i = 1,\ 2,\ 3$. This represents a linear probabilistic superposition of $2^3 = 8$ states.

$$\frac{\sqrt{6}}{27}|000\rangle - \frac{\sqrt{21}}{27}|001\rangle + \frac{\sqrt{48}}{27}|010\rangle - \frac{\sqrt{168}}{27}|011\rangle$$
$$+\frac{\sqrt{12}}{27}|100\rangle - \frac{\sqrt{42}}{27}|101\rangle + \frac{\sqrt{96}}{27}|110\rangle - \frac{\sqrt{336}}{27}|111\rangle$$

The above system can output states 000, 001,010, 011,100,101,110,111 with probability of 6/729, 21/729, 48/729, 168/729, 12/729, 42/729, 96/729 and 336/729 respectively. Every time we measure the system, we obtain new state based on its probability.

1.3 Quantum Gate

Various quantum gates such as the NOT gate, AND gate, OR gate, NAND gate and rotation gate can be used to modify the q-bit state [11]. As shown in Fig. 1, single q-bit can be represented by unit circle, and the quantum rotation gate $\cup(\theta i)$ is the most famous gate in previous work [6, 7].

$$\cup(\theta i) = \begin{bmatrix} \cos\theta i & -\sin\theta i \\ \sin\theta i & \cos\theta i \end{bmatrix} \tag{2}$$

Where θi is rotation angle in Eq. (3).

$$\theta i = s\ (\alpha_i,\ \beta_i) * \Delta\theta i \tag{3}$$

$s(\alpha_i, \beta_i)$ is the sign of θi that determines the direction, $\Delta\theta i$ is the magnitude of rotation gate (called rotational angle).

1.4 Quantum Measurement

The process of probabilistic observation collapse or measure each q-bit, to be rendered into "0" or "1". This process showed in Fig. 2.

x is the observed value of q-bit. Each time we observe the same q-bit we can obtain different value, since the observation process depends on random value. From here superposition feature can be achieved to produce all different states of quantum system, since it is probabilistic measurable.

2 Problem Definition

A constraint optimization problem to be solved, usually written as the following

Find $\vec{x}$ which minimizes
f($\vec{x}$)
subject to

$$g_i(\vec{x}) \leq 0, \quad i = 1, \ldots, I$$
$$h_j(\vec{x}) = 0, \quad j = 1, \ldots, J$$

Where $\vec{x} \in R^n$ is the vector of solution $\vec{x} = [x_1, x_2, \ldots, x_n]^T$, I is the number of inequality constraints and J is the number of equality constraints. Each x_k, k = 1,…,n is bounded by lower and upper limits $L_k \leq x_k \leq U_k$ which defines the search space. Both the objective function and the constraints can be linear or nonlinear. The equality constraints usually transformed to inequality ones [12] as follows

$$\left|h_j(\vec{x})\right| - \varepsilon \leq 0$$

Where $\boldsymbol{\varepsilon}$ is the tolerance allowed (a very small value). As showed in [13] constraint handling methods in classical optimization can be classified into

- *Generic methods*: where the mathematical structure of the problem is not considered during the solution (linear or nonlinear).
- *Specific methods*: only relevant to a special type of constraints.

Fig. 2 Binary observation of q-bit

```
Begin
   If random (0,1) < |α_i|^2

   Then  x ← 0
   Else   x ← 1
End
```

Generic methods, such as penalty function method and Lagrange multiplier method [14] each one of them might be connected to any issue without much change, but they are not efficient with all types of problems. Specific methods, such as cutting plane method and the reduced gradient method [14], applicable to convex feasible regions or to problems with fewer dimensions.

3 Penalty Function

Based on the mathematical programming, where the constrained problem transformed into unconstrained one. The performance of penalty function is not satisfactory since it is generic method. First, all equality constraints transformed to inequality ones, the fitness function $F(\vec{x})$ designed as the sum of the objective function $f(\vec{x})$ and the penalty term.

$$F(\vec{x}) = f(\vec{x}) + \sum_{i=1}^{I} R_i |g_i(\vec{x})|^2 \tag{4}$$

Where $|g_i(\vec{x})|$ denotes the absolute value of the constraint violation, the parameter R_i is the penalty parameter for the ith inequality constraint. As showed in [13] there are two problems when using penalty functions.

1. The optimal solution of $F(\vec{x})$ depends on the penalty parameter R_i. Every value for R_i produces different optimal solution, users have to run more experiments for the same problem to reach the optimal setting of R_i.
2. Choosing the R_i randomly make distortion to the objective function and produces artificial local optimal solutions.

3.1 Novel Penalty Function

In this section we explain a novel penalty function proposed by Kalyanmoy Deb [13], where no penalty parameters needed. This method as explained by him can be general method for solving linear and non-linear constraints. The author presented tournament selection operator where the following criteria enforced in his work

- Any feasible solution is preferred to any unfeasible one.
- Among two feasible solutions, the one having better objective function value is preferred.
- Among two unfeasible solutions, the one with less constraint violation is preferred.

K. Deb [13] modified the evaluation function to measure individual solutions. He clarifies why it is not reasonable to use Eq. (4), since it make no sense to calculate objective function value for unfeasible solution. He devised the following fitness function, where unfeasible solutions are compared based on only their constraint

violation. Before working all equality constraints transformed to inequality ones, so he assumed to have m inequality constraints.

$$F(\vec{x}) = \begin{cases} f(\vec{x}) & \text{if } g_i(\vec{x}) \leq 0 \quad \forall i = 1, 2, 3, \ldots, m \\ f_{max} + \sum_{i=1}^{m} |g_i(\vec{x})| & \text{otherwise} \end{cases}$$

Where f_{max} is the objective function value of the worst feasible solution in the current population. Thus the objective function for feasible solutions is the main objective function, while the objective function for unfeasible solutions depends on the constraint violation in addition to the worst feasible solution in the population at hand. If no feasible solution exists in a population the value of f_{max} will be zero. K. Deb [13] explained with figures how the new modified objective function forces unfeasible solutions to come closer and inside the feasible region. The author used the niching methods [15] and parameter-based mutation operator [16] to keep diversity between feasible solutions. K. Deb [13] implemented a normalized Euclidean distance as a niching strategy under the tournament selection operator. If the distance dij between two feasible parents (i and j) less than threshold value they are compared, otherwise, they are not compared and another solution j is checked. The normalized Euclidean distance calculated as follows:

$$d_{ij} = \sqrt{\frac{1}{n} \sum_{k=1}^{n} \left(\frac{x_k^{(i)} - x_k^{(j)}}{x_k^u - x_k^l} \right)^2} \tag{5}$$

This strategy allows feasible solutions that are far away from each other to be not compared and thus keep diversity.

4 The Proposed Method

This section uses the QGAXM discussed in [17] to work with the previous novel penalty function. There is no need to use any penalty parameters.

- The superposition feature of quantum individuals automatically gives benefits to unfeasible solutions to become feasible.
- The proposed method doesn't use any niching strategy, since diversity of population kept by quantum crossover and quantum mutation operators.
- Quantum rotational gate forces unfeasible solution to be closer or inside the feasible region if feasible solution exists.
- If the best solution doesn't change during 10 generations we consider it is trapped into local minimum, then the best 5 feasible solutions reserved and the remaining individuals generated randomly.

5 Experimental Results

In this section the performance of the proposed method will be tested on well-studied non-linear constrained optimization problems found in [13, 18]. The parameter setting of QBGA will be selected from Table 1, and will be modified from problem to another to be fair during comparison.

5.1 Test Problem 1

$$\begin{aligned} \text{Minimize} \quad & f_1(\vec{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 \\ \text{S .t} \quad & h(\vec{x}) = x_1 - 2x_2 + 1 = 0, \\ & g(\vec{x}) = -(x_1^2/4) - x_2^2 + 1 \geq 0, \\ & -10 \leq x_i \leq 10, i = 1, 2 \end{aligned}$$

The optimum solution recorded in previous literature is $\vec{x}^* = [0.82288,\ 0.91144]$ with $\boldsymbol{f}(\overrightarrow{\boldsymbol{x}^*}) = 1.393454$. Table 2 represents different comparisons of the best

Table 1 Different parameters affect quality of QBGA

Parameters (Factors)	**Levels (Values)**		
A. Mutation Operator	1. Quantum inversion mutation		
	2. Left & Right quantum swap mutation		
	3. Quantum boundary mutation		
	4. Quantum neighbor-swap mutation		
	5. Quantum shift mutation		
	6. Quantum inverse-swap mutation		
	7. Quantum displacement mutation		
	8. Quantum makinen,periaux and toivanen mutation		
	9. Quantum non-uniform mutation		
	10. Quantum power mutation		
B. Crossover Operator	1. Single-point quantum crossover		
	2. Two-point quantum crossover		
	3. Multi-point quantum crossover		
	4. Quantum interference crossover		
	5. Arithmetic quantum crossover		
C. Mutation Probability	(1) 0.05	(2) 0.2	(3) 0.3
	(4) 0.4	(5) 0.5	
D. Crossover Probability	(1) 0.05	(2) 0.1	(3) 0.3
	(4) 0.5	(5) 0.7	
E. Population Size	(1) 10	(2) 20	(3) 50
	(4) 80	(5) 100	
F. Rotational Angle	(1) 0.01π	(2) 0.05π	(3) 0.08π
	(4) 0.1π	(5) Eq. 13 [17]	

Table 2 Comparison of results from different methods for test problem 1

Method	X1	X2	h(x)	g(x)	f(x)
EP	0.835	0.9125	0.01	−0.0069625	1.36488125
GA	0.808	0.8854	0.0372	0.05285084	1.43399716
HS	0.8343	0.9121	0.0101	−0.00594053	1.3665829
QBGA	0.822884518	0.9112964024	2.92E−04	2.54E−04	1.3934691
MBA	0.822875	0.911437	1.00E−06	1.78E−06	1.39346667
Optimal	0.82288	0.91144	0	−5.75E−06	1.393454368

solution obtained by GA [19], EP [20], Harmony search (HS) [21], Mine blast algorithm (MBA) [18]. From Table 2 the proposed method (QBGA) gives near optimal solution. The parameter setting of the QBGA to solve test problem 1 was (4 1 5 2 2 3) from Table 1, with 500 iteration. Each variable encoded with 25 q-bits, 15 q-bits for integer part and 10 q-bits for fraction part.

5.2 Test Problem 2

$$\begin{aligned} \text{Minimize} \quad & f_2(\vec{x}) = x_1^2 + (x_2 - 1)^2 \\ \text{S .t} \quad & h(\vec{x}) = x_2 - x_1^2 = 0, \\ & -1 \le x_i \le 1,\ i = 1, 2 \end{aligned}$$

Table 3 represents the comparisons between optimal solution and the related design variables with cultured differential evolution (CULDE) [22] and MBA [18]. The parameter setting of the QBGA to solve test problem 2 was (4 1 4 2 4 1) from Table 1, with 150 iteration. Each variable encoded with 15 q-bits, 10 q-bits for integer part and 5 q-bits for fraction part.

5.3 Test Problem 3

$$\begin{aligned} \text{Minimize} \quad & f_3(\vec{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \\ \text{S .t} \quad & g_1(\vec{x}) = 4.84 - (x_1 - .05)^2 - (x_2 - 2.5)^2 \ge 0, \\ & g_2(\vec{x}) = x_1^2 + (x_2 - 2.5)^2 - 4.84 \ge 0, \\ & 0 \le x_i \le 6, i = 1, 2 \end{aligned}$$

Table 3 Comparison of results from different methods for test problem 2

Method	X1	X2	h(x)	f(x)
CLUDE	−0.707036	0.500000	1.94E−04	0.749899905296
QBGA	*−0.705339687194*	*0.4975562072336*	*5.213E−05*	*0.7499538392211*
	−0.71346529814	*0.50910312805*	*7.0396E−05*	*0.7500124705395*
MBA	−0.706958	0.499790	3.862E−07	0.749999657864
Optimal	−0.70711	0.500000	−4.55E−06	0.7500045521

The constrained optimum solution is $\vec{x}^* = [2.246826, 2.381865]$ with a function value equal to $f(\overrightarrow{x^*}) = 13.59085$. As described in [13] the feasible region is a narrow crescent-shaped region (approximately .7 % of the total search space). The parameter setting of the QBGA to solve test problem 3 was (4 1 4 2 4 4) from Table 1, each variable encoded with 36 q-bits,16 q-bits for integer part and 20 q-bits for fraction part. Although the feasible region is very small, the proposed method found 50 feasible solutions during 50 run. TS-R in Table 4 means the novel penalty function in [13] implemented under real GA. Table 5 shows statistical comparisons with Deb [13]. TS-B means the novel penalty function implemented under binary GA. The QBGA recorded optimal solution [2.246836837922230, 2.382068209169000] with objective value equals (13.5908430969596). From Table 5 the worst optimum from 50 run was 43.87163 which is better than worst values obtained by Deb [13]. Median is the 25th value after sorting the 50 objective value in ascending order.

5.4 Test Problem 4

$$\begin{aligned} \text{Maximize} \quad & f_4(\vec{x}) = \frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1+x_2)} \\ \text{S .t} \quad & g_1(\vec{x}) = x_1^2 - x_2 + 1 \le 0, \\ & g_2(\vec{x}) = 1 - x_1 + (x_2 - 4)^2 \le 0, \\ & 0 \le x_i \le 10, i = 1, 2 \end{aligned}$$

This maximization problem previously solved by HM [21], PSO-DE, PSO, GA [23]. Table 6 shows statistical comparison results for this problem. The

Table 4 Comparison of results from different methods for test problem 3

Method	X1	X2	g1(x)	g2(x)	f(x)
TS-R	2.246826	2.381865	−3.52E−07	0.22218295	13.59085
HS	2.246840	2.382136	2.09E−06	0.2221819	13.5908585
QBGA	2.2468368379	2.382068209	2.59E−010	0.22218368	13.5908430
MBA	2.246833	2.381997	6.21E−08	0.22218324	13.5908427
Optimal	2.246826	2.381865	−3.52E−07	0.22218295	13.59085

Table 5 Number of runs = 50, results for test problem 3

	Best	Median	Worst
Deb [13] (**Maximum generation = 50**) **& population size = 50**			
TS-B	13.59658	37.90495	244.11616
TS-R	13.59085	13.61673	117.02971
QBGA (Maximum generation = 30) & population size = 80			
	13.590843	13.70942	43.87163

Table 6 Comparison of results from different methods for test problem 4

Method	Worst	Mean	Best	SD
HM	−0.0291438	−0.0891568	−0.0958250	N.A
PSO	−0.02914408	−0.09449230	−0.09582594	9.4E−03
PSO-DE	−0.0958259	−0.0958259	−0.0958259	1.3E−12
GA	−0.0958250	−0.0958250	−0.0958250	2.70E−09
QBGA	−0.09577295	−0.09580933	−0.09582503	2.04 E−05
MBA	−0.0958250	−0.0958250	−0.0958250	0

parameter setting of the QBGA to solve test problem 4 was (4 1 4 2 2 3) from Table 1, each variable encoded with 25 q-bits, 15 q-bits for integer part and 10 q-bits for fraction part. The optimum solution recorded by the proposed method is $[1.227979286038545,\ 4.245321369441816]^T$ with objective function value equals (-0.095825035944661).

5.5 Test Problem 5

This problem has five variables and six inequality constraints [13].

$$
\begin{aligned}
\text{Minimize} \quad & f_5(\vec{x}) = 5.3578547x_3^2 + .8356891x_1x_5 + 37.293239x_1 - 40792.141 \\
\text{S .t} \quad & g_1(\vec{x}) = 85.334407 + .0056858x_2x_5 + .0006262x_1x_4 - .0022053x_3x_5 \geq 0, \\
& g_2(\vec{x}) = 85.334407 + .0056858x_2x_5 + .0006262x_1x_4 - .0022053x_3x_5 \leq 92, \\
& g_3(\vec{x}) = 80.51249 + .0071317x_2x_5 + .0029955x_1x_2 + .0021813x_3^2 \geq 90, \\
& g_4(\vec{x}) = 80.51249 + .0071317x_2x_5 + .0029955x_1x_2 + .0021813x_3^2 \leq 110, \\
& g_5(\vec{x}) = 9.300961 + .0047026x_3x_5 + .0012547x_1x_3 + .0019085x_3x_4 \geq 20, \\
& g_6(\vec{x}) = 9.300961 + .0047026x_3x_5 + .0012547x_1x_3 + .0019085x_3x_4 \leq 25, \\
& 78 \leq x_1 \leq 102, \\
& 33 \leq x_2 \leq 45, \\
& 27 \leq x_i \leq 45, i = 3, 4, 5
\end{aligned}
$$

From Table 7, we can see that the worst value obtained by QBGA is better than most of the median values and is the best of all worst values in K. Deb [13]. Only we needed to run QBGA with 1,000 generation each with 50 individuals to prove superiority over 5,000 generation on the other side. The best obtained objective function value via the proposed method was (−30649.5178) and the optimum solution was $\left[\ 78,\ 33.063059,\ 30.0967741,\ 45,\ 36.520161\ \right]^T$. Deb [13] obtained best optimal value with (−30665.537) when he used mutation and niching under 5,000 generation. The parameter setting of the QBGA to solve test problem 5 was (4 1 4 2 3 3) from Table 1, each variable encoded with 10 q-bits, 8 q-bits for integer part and 2 q-bits for fraction part.

Table 7 Number of runs = 50, results for test problem 5

Mutation	Niching	Best	Median	Worst
Deb [13] (**Maximum generation = 1,000**) **& population size = 50**				
No	No	−30614.814	−30196.404	−29606.451
No	Yes	−30646.469	−30279.744	−29794.441
Deb [13] (**Maximum generation = 5,000**) **& population size = 50**				
No	No	−30614.814	−30196.404	−29606.596
No	Yes	−30651.865	−30376.906	−29913.635
Yes	Yes	−30665.537	−30665.535	−29846.654
QBGA (Maximum generation = 1,000) & population size = 50				
		−30649.5178	−30600.47308	−30396.111328

5.6 Test Problem 6

This problem having 13 variables and 9 inequality constraints [13].

$$
\begin{aligned}
\text{Minimize} \quad & f_6(\vec{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i \\
\text{S .t} \quad & g_1(\vec{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} \leq 10, \\
& g_2(\vec{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} \leq 10, \\
& g_3(\vec{x}) = 2x_2 + 2x_3 + x_{11} + x_{12} \leq 10, \\
& g_4(\vec{x}) = -8x_1 + x_{10} \leq 0, \\
& g_5(\vec{x}) = -8x_2 + x_{11} \leq 0, \\
& g_6(\vec{x}) = -8x_3 + x_{12} \leq 0, \\
& g_7(\vec{x}) = -2x_4 - x_5 + x_{10} \leq 0, \\
& g_8(\vec{x}) = -2x_6 - x_7 + x_{11} \leq 0, \\
& g_9(\vec{x}) = -2x_8 - x_9 + x_{12} \leq 0, \\
& 0 \leq x_i \leq 1, \; i = 1, \ldots, 9 \\
& 0 \leq x_i \leq 100, \; i = 10, 11, 12 \\
& 0 \leq x_{13} \leq 1
\end{aligned}
$$

The optimal solution to this problem is $\vec{x}^* = \left[\, 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 3\ 3\ 3\ 1 \,\right]$ with $f(\overrightarrow{x}) = -15$. This is easy problem with the objective function and constraints being linear or quadratic as described in [24]. The parameter setting of the QBGA to solve test problem 6 was (4 1 4 2 4 3) from Table 1, each variable encoded with 25 q-bits, 15 q-bits for integer part and 10 q-bits for fraction part. Table 8 shows the best, median and the worst of 50 run. The best obtained objective function value via the proposed method was (−14.9571735) and the optimum solution is $\left[\, 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 2.999967,\ 2.978604,\ 2.978604,\ 1 \,\right]^T$.

Table 8 Number of runs = 50, results for test problem 6

Mutation	Niching	Best	Median	Worst
Deb [13] (**Maximum generation = 500**) **& population size = 130**				
No	No	−15	−15	−9.603
No	Yes	−15	−15	−10.959
Yes	Yes	−15	−15	−13
QBGA (Maximum generation = 500) & population size = 80				
		−14.95717359	−14.8235	−11.8823

6 Graphics Processing Units (GPU)

GPU originally evolved to accelerate 3D graphics on personal computers. The hardware structure of GPU supports high performance and computing power exceeds traditional CPUs. Modern GPUs contain thousands of arithmetic units, hundreds of processors, tens of thousands of concurrent threads, big memory bandwidth. All this features combined with their programmability and low cost platform attracted researchers to use GPU for general purpose computing (GPGPU) [25] [26].

GPU is optimized for single program multiple data (SPMD), meaning that single program can be executed on independent multiple data simultaneously, this kernel program called by CPU host to perform parallel processing on the GPU device. GPU is high-performance and valuable only for parallelized algorithms, so implementing parallel EAs over GPU proved high-quality in processing speed [26–29]. Parallelized individuals (chromosomes) for GPU can be organized via texture mapping [27, 28] and island models [26, 29].

6.1 Texture Mapping

The population is mapped into textures in organized manner, then transferred from CPU system memory to GPU global memory, every pixel of texture capsulate 4 genes of single chromosome similar to (R, G, B, A) color channel. This makes wide use of parallel processing by dealing with 4 genes from each individual simultaneously. The limitation of textures can be stated as follows:

- Only 16 textures can be processed simultaneously on nVidia GeForce FX 6800 [28].
- Capacity size not exceeding 4096 * 4096 for each texture on nVidia GeForce FX 6800 [28].

Parallel blend crossover and parallel uniform mutation applied in [27] to achieve diversity, while [28] assumed that applying parallel crossover process will be more time consuming, thus the author implemented evolutionary programming algorithm (EP) dependable on parallel Cauchy mutation operator only. Both crossover and mutation operators uses random numbers in their process. From our study

to [27] the author, presented the Random Numbers Generator section, with intelligent methodology, instead of transferring random texture from CPU to GPU with every generation, this random texture transferred once then parallel Linear Congruential Generator (LCG) used to update it on GPU. To minimize data transfer of individuals between CPU and GPU, indexed array used in [28] where selection and replacement strategies connected to this array and population textures kept on GPU. Experimental results performed in [27, 28] related to small size optimization problems,where adaptability of these algorithms will be lost for high dimensional ones, also both of those algorithms missing the computing power offered by fast shared memory.

6.2 Island Models (IMs)

IMs are the most widely used schemes to implement parallel EAs. When dealing with these models over GPU we study some issues as global memory, local shared memory, threaded blocks, synchronous and a synchronous version, migration topologies, selection and replacement operators. Three different parallel schemes of IM discussed in [29], (1) Parallel evaluation of the population on GPU, (2) Full distributed IM on GPU and (3) Full distributed IM on GPU using fast shared memory. The advantages, limitations of these schemes are presented in the same work. Migration mechanism performed in a synchronous manner since threads are not guaranteed to finish processing at the same time. Forcing synchronous IM on GPU can be achieved implicitly via CPU, but with decreased performance. Using local shared memory with IM proved good acceleration factors up to *1757 processing time compared to traditional CPU when increasing problem dimension and up to *2074 when increasing number of islands (population size) [29]. For high technical paper discusses IM from implementation side [26], where selection and parallel arithmetic crossover combined for each island to optimize processing time and memory usage. The same migration ring topology applied in [26, 29] to achieve diversity.

6.3 Quantum Inspired Evolutionary Algorithm Over GPU

Original QIEA uses intelligent quantum-gate operator to guide all individuals to the position of the best one. Figure 3 clarifies this process, where the large green circle represents the global best solution. QIEA did not search for optimal solution in neighborhood and this being a weak point. Our aim is to change the search directions inside the population by segmenting into subpopulations and selecting best individual (leader) for each group of individuals. Figure 4 shows this concept, where q-gate directs individuals to the leader positions instead of the global best solution, red circles are the leaders for groups.

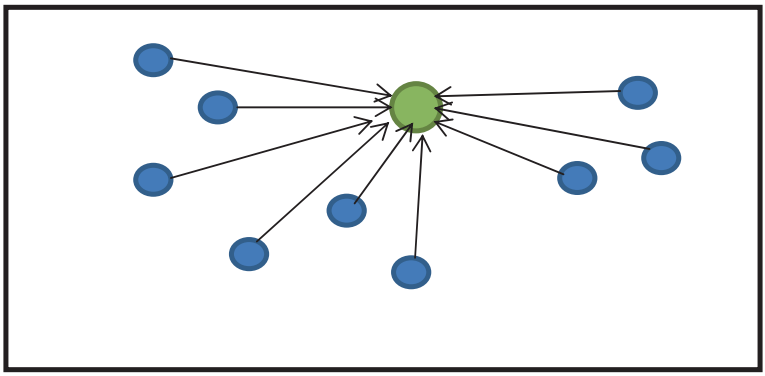


Fig. 3 Q-gate guides all individuals to the best position

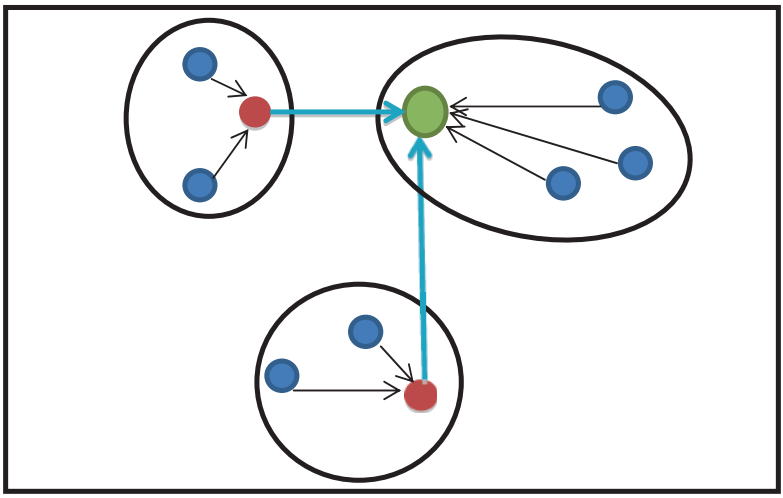


Fig. 4 Segmenting to subpopulations and searching neighborhood spaces

After periodic time (10 generations) each sub-group copy its leader to the global memory, selecting the global best individual, then all the leaders directed to this selected global. Leaders of groups are replaced if better solution obtained. Figure 5 represents simple flowchart of single island model over GPU using fast shared memory.

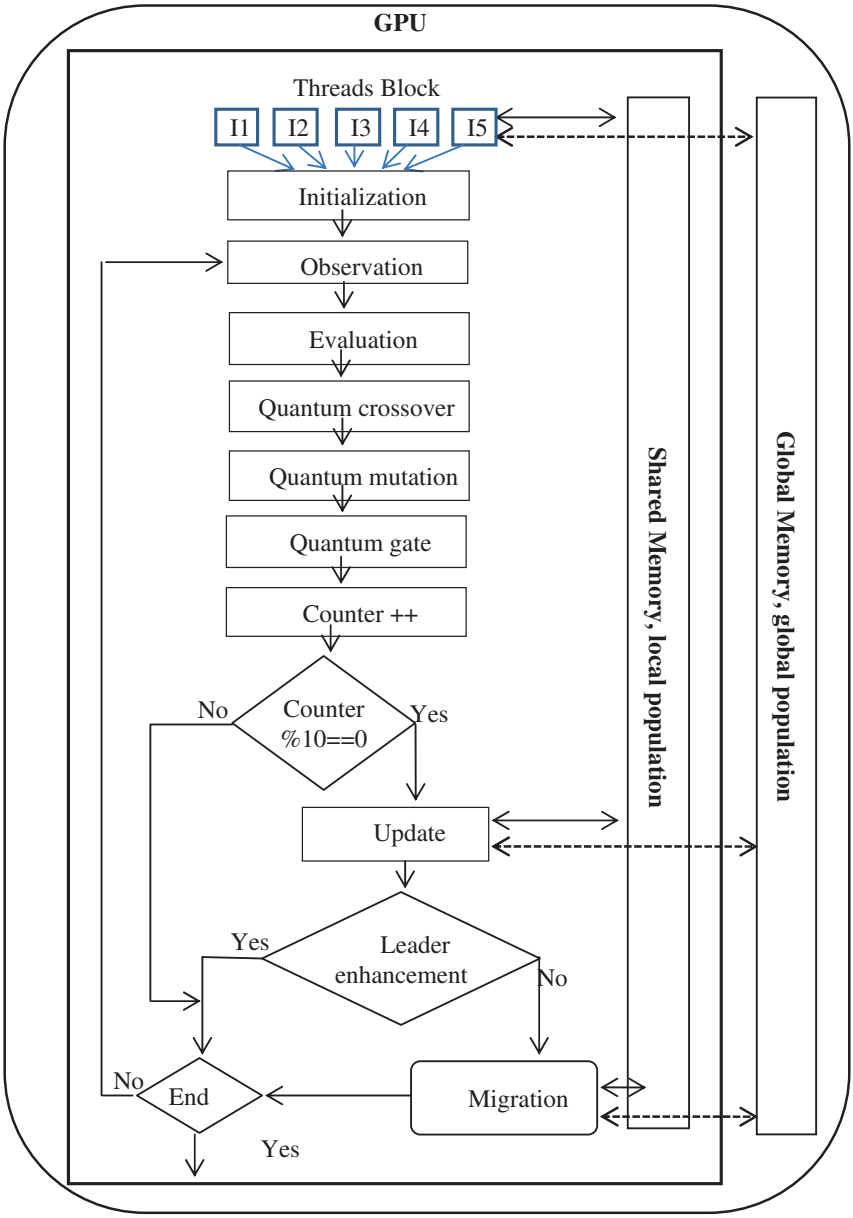


Fig. 5 Each subpopulation represented with single island model

7 Conclusions

The quantum behaved genetic algorithm proved good results when solving constraints optimization problems. The superposition of quantum computing gives big reasoning for not neglecting infeasible solutions. From experimental results, the proposed method can be considered as alternative intelligent evolutionary algorithm to find near optimal solution. We expect that quantum inspired computing over GPU will have excellent impact on the research domain of optimization, since World Wide Web missing material related to this important topic. Accelerating execution speed will be achieved by GPU while obtaining global solutions will be obtained with quantum inspired algorithms.

References

1. Nielsen, A.M., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
2. Bartlett, S.D.: Quantum computing: powered by magic. Nature **510**, 345 (2014)
3. Giraldi, G.A., Portugal, R., Thess, R.N.: Genetic Algorithms and Quantum Computation. CoRR (cs.NE/0403003) (2004)
4. Malossini, A., Blanzieri, E., Calarco, T.: Quantum genetic optimization. IEEE Trans. Evol. Comput. **12**, 231–241 (2008)
5. Sofge, D.A.: Toward a framework for quantum evolutionary computation. In: Proceedings of the CIS, pp. 789–794 (2006)
6. Han, K., Kim, J.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Trans. Evol. Comput. **6**(6), 580–593 (2002)
7. Han, K., Kim, J.: Quantum-inspired evolutionary algorithms with a new termination criterion, h-epsilon gate, and two-phase scheme. IEEE Trans. Evol. Comput. **8**(2), 156–169 (2004)
8. Draa, A., Meshoul, S., Talbi, H., Batouche, A.: Quantum-inspired differential evolution algorithm for solving the N-queens problem. Int. Arab J. Inf. Technol. **7**(1), 21–27 (2010)
9. Mohammed, A., Elhefnawy, N., El-Sherbiny, M., Hadhoud, M.: Quantum crossover based quantum genetic algorithm for solving non-linear programming. In: Proceedings of the 8th International Conference on Informatics and Systems (INFOS) (2012)
10. Zhou, S., Sun, Z.: A new approach belonging to EDAs: quantum-inspired genetic algorithm with only one chromosome. In: Proceedings of the Advances in Natural Computation, pp. 141–150. Springer, Heidelberg (2005)
11. Williams, C.P.: Explorations in Quantum Computing. Springer, Heidelberg (2011)
12. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. IEEE Trans. Evol. Comput. **4**, 284–294 (2000)
13. Deb, K.: An efficient constraint handling method for genetic algorithms. Comput. Methods Appl. Mech. Eng. **186**(2), 311–338 (2000)
14. Reklaitis, G.V., Ravindran, A., Ragsdell, K.M.: Engineering Optimization Methods and Applications. Wiley, New York (1983)
15. Deb, K., Goldberg, D.E.: An investigation of niche and species formation in genetic function optimization. In: Proceedings of the 3rd International Conference on Genetic Algorithms', pp. 42–50. Morgan Kaufmann Publishers, San Francisco (1989)
16. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing, Boston (1989)

17. Mohammed, A., Elhefnawy, N., El-Sherbiny, M., Hadhoud, M.: Quantum inspired evolutionary algorithms with parametric analysis. In: Paper Presented at the Conference on Science and Information (SAI) (2014)
18. Sadollah, A., Bahreininejad, A., Eskandar, H., Hamdi, M.: Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl. Soft Comput. **13**(5), 2592–2612 (2013)
19. Homaifar, A., Qi, C.X., Lai, S.H.: Constrained optimization via genetic algorithms. Simulation **62**, 242–253 (1994)
20. Fogel, D.B.: A comparison of evolutionary programming and genetic algorithms on selected constrained optimization problems. Simulation **64**, 397–404 (1995)
21. Lee, K.S., Geem, Z.W.: A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput. Methods Appl. Mech. Eng. **194**(36), 3902–3933 (2005)
22. Becerra, R.L., Coello, C.A.C.: Cultured differential evolution for constrained optimization. Comput. Methods Appl. Mech. Eng. **195**, 4303–4322 (2006)
23. Chootinan, P., Chen, A.: Constraint handling in genetic algorithms using a gradient-based repair method. Comput. Oper. Res. **33**(8), 2263–2281 (2006)
24. Michalewicz, Z.: Genetic algorithms, numerical optimization, and constraints, pp. 151–158 (1995)
25. Krüger, J., Westermann, R.: Linear algebra operators for GPU implementation of numerical algorithms. ACM Trans. Graph. **22**(3), 908–916 (2003)
26. Pospichal, P., Jaros, J.: Gpu-based acceleration of the genetic algorithm. GECCO competition (2009)
27. Yu, Q., Chen, C., Pan, Z.: Parallel genetic algorithms on programmable graphics hardware. In: Proceedings of the 1st International Conference on Advances in Natural Computation—Volume Part III, pp. 1051–1059. Springer, Heidelberg (2005)
28. Wong, M.-L., Wong, T.-T. Fok, K.-L.: Parallel evolutionary algorithms on graphics processing unit. In: IEEE Congress on Evolutionary Computation, pp. 2286–2293 (2005)
29. Luong, T.V., Melab, N., Talbi, E.-G.: GPU-based island model for evolutionary algorithms. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, pp. 1089–1096. ACM, New York (2010)

A Genetic Algorithm Approach for Optimizing a Single-Finger Arabic Keyboard Layout

Nourah Alswaidan, Manar I. Hosny and Abir Benabid Najjar

Abstract The use of cellphones and handheld devices in our daily activities is not limited to making calls or writing short text messages. The added features of wireless technology and related applications made it possible to write emails, notes and long text. Nevertheless, the currently used keyboards in portable devices are not optimized for such use, in terms of rapid and ergonomic typing. In this research, we aim to optimize the design of the Arabic keyboard layout for applications that predominantly use a single pointer, such as those used in portable devices. The main objective is to find the best single-finger Arabic keyboard layout that allows users of portable devices to write text and carry out written conversations for a long time with comfort, ease, and speed. Since the single-finger keyboard layout problem can be modeled in terms of the famous Quadratic Assignment Problem (QAP), which is known to be NP-hard, heuristics and meta-heuristics are recommended for solving such problem. To adapt the problem to the requirements of optimizing the single-finger Arabic keyboard, we added two measures to the classical—distance based—objective function of the QAP, which are: the keyboard row weight and the hit direction of the finger. A Genetic Algorithm (GA) approach with two different crossover types (*two-point* and *modified uniform crossovers*), and three different mutation operators (*swap*, *insertion*, and *Simulated Annealing* (*SA*)) was developed and thoroughly tested. The experimental results demonstrated that the simple *swap* mutation produced better results than the other mutations, with both crossover types. Moreover, experimental testing has shown that the added measures in the objective function had a positive effect, in terms of improving the typing speed, when compared to the original QAP objective function.

N. Alswaidan (✉) · M.I. Hosny · A.B. Najjar
College of Computer and Information Sciences, King Saud University,
Riyadh, Saudi Arabia
e-mail: nourah_swaidan@yahoo.com

M.I. Hosny
e-mail: mifawzi@ksu.edu.sa

A.B. Najjar
e-mail: abbenabid@ksu.edu.sa

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_16

Finally, comparing the resulting optimized keyboard layout with other existing keyboards showed that our keyboard layout is favorable, in terms of the optimization criteria considered in this research, than the other layouts tested.

1 Introduction

Nowadays, people extensively use handheld and portable devices, such as: smart phones, tablets, phablets, etc. These devices enable users to send emails, browse the Internet and indulge in different social media. However, the keyboard design of these devices is not currently optimized for such extensive use, in terms of enhancing the writing speed and increasing the accuracy and comfort of typing using only one finger or one pointer. This is particularly true for people with special needs, who cannot type using their hands and may use alternative devices, such as a mouth stick, for typing. This paper focuses on designing an optimized single-finger (s-finger) keyboard layout for the Arabic language, in order to enhance the typing speed as well as the ease and accuracy of typing.

Designing an optimized keyboard is, in fact, a challenging problem that has not been extensively researched. To the best of our knowledge, the first attempts to optimize an s-finger Arabic keyboard were in the two versions of this research [8, 14]. Although there is some research on the many-finger (m-finger) Arabic keyboard layout, its results cannot be adopted for single finger use, since the optimization criteria is different. For example, in order to increase the typing speed in the m-finger case, the most frequent pairs of letters should be placed far from each other, so that when the first letter is being typed another finger is ready to type the next letter [13]. Contrary to this, the frequent pairs of letters in the s-finger keyboard must be placed adjacent to each other, so that the finger can move faster from one letter to another.

The s-finger keyboard layout problem can be modeled in terms of the famous Quadratic Assignment Problem (QAP). The QAP is a well-known combinatorial optimization problem, which was proposed in 1957 by Koopmans and Beckmann [10]. It was used to model the facility layout problem, where n facilities must be located in n locations. Two matrices were used in this context, the first matrix holds the flow, $C = [c_{ij}]$, where c_{ij} is the flow between facilities i and j. The second matrix holds the distance, $D = [d_{kl}]$, where d_{kl} is the distance between locations k and l. When a facility i is assigned to location $p(i)$, and facility j is assigned to location $p(j)$, the cost of this assignment is calculated as $c_{ij}d_{p(i)p(j)}$. Then the total cost of all facilities assignment will be the sum of each assignment over all i and j. The objective function is defined as follows:

$$\text{cost} = \min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} d_{p(i)p(j)} \tag{1}$$

What makes the QAP important is its ability to model many real-life problems, such as: the backboard wiring problem [16], the campus layout problem [4], the

hospital layout problem [6], and the keyboard layout problem [12]. In general, the possible number of solutions that can be obtained when solving the QAP is huge (O(n!)), which makes the problem NP-hard [18]. Meta-heuristic methods are usually preferred for solving hard optimization problems like the QAP.

To model the keyboard layout problem in terms of the QAP, letters are mapped to facilities, and locations are mapped to keyboard keys. In addition, the flow between a pair of facilities corresponds to the frequency of occurrence of a letter pair, and the distance between the two locations is simply the distance separating the keys to which the pair is assigned on the keyboard. Thus, the optimization of the layout is based on the frequency of occurrence of letters. In our proposed approach, we calculated the frequency of Arabic Articles using articles from Arabic Wikipedia. Then a Genetic Algorithm (GA) was implemented for the optimization phase. Our GA adopts two crossover methods (two-point and modified uniform crossovers). In addition, three mutation operators were tried: swap, insertion and Simulated Annealing (SA). A problem specific objective function was adopted to guide our GA towards attaining the optimized layout design.

The rest of this chapter is organized as follows: first, a literature review of keyboard layout optimization problems and their solution methods for different languages is summarized in Sect. 2. Then our proposed algorithm for solving this problem is presented in Sect. 3. This is then followed by the experimental results in Sect. 4, and finally conclusions with a brief summary and intended future work are in Sect. 5.

2 Related Work

The most well-known keyboard layout for the English language is the QWERTY keyboard. It was presented in 1872 and was named after the first six characters in the top row of the keyboard. Originally, this format was used in typewriters, where the characters are arranged in a way that slows down typists so that the machine parts would not be jammed. Another keyboard was proposed in 1932 named Dvorak; it claims that placing the vowels in the middle row will overcome the limitations of the QWERTY keyboard. However, the long-term use of the QWERTY keyboard made it difficult to be replaced by another at that time, due to the cost of retraining the users [5, 11]. However, the current keyboard layouts do not support the typing speed and other ergonomic criteria [11]. Thus, a number of studies that tackle the keyboard layout optimization problem have emerged. These studies vary in their solution approaches as well as the languages used in the keyboard.

2.1 Latin Keyboards

Light and Anderson [12], designed an optimized English keyboard layout using simulated annealing. They modeled the problem as the QAP, and used the

frequency of pairs of letters and the time needed to travel to each letter. The results showed that the layouts produced by this algorithm are better and can type faster in comparison to the QWERTY and Dvorak's typewriters. Also, they noticed that the algorithm always manages to:

- Place the most common letters where they are typed using the index or the middle fingers, which are considered to be the strongest of all fingers.
- Place the least common used letters at the bottom row.

Eggers et al. [5], applied an Ant Colony Algorithm (ACO) to optimize French, German and English keyboard layouts. In order to calculate the frequency of the characters in these languages, they used texts that appeared in the newspapers of Le Monde, Der Spiegel and USA Today for the French, German and English languages respectively. Their aim was to achieve a higher typing speed without causing fatigue or typing errors, and to allow the use of 'touch-typing'. To achieve such goals, the objective function tests for six criteria, which are:

1. *Accessibility and load*: Distribution of the typing load with regard to the strength of the fingers.
2. *Key number*: The number of keys hit to type a text must be minimized. If the shift key is not used, the number of keys used to type a text is fixed.
3. *Hand alteration*: Having to change hands for each character will minimize the time and speed up typing. This is because while having one hand reaching to type a character, the other hand is moving towards the next character.
4. *Consecutive usage of the same finger*: Typing two characters with the same finger will slow down the typing speed.
5. *Avoid big steps*: When using the same hand for typing, the distance between two consecutive keys must be minimum.
6. *Hit direction*: When using one hand for typing two consecutive characters, the movement direction of the fingers must be from the little finger to the thump.

2.2 Other Languages Keyboards

Samimi [15] designed an optimized Persian keyboard layout using a hybrid SA and GA approach. He attempted to optimize 33 letters that will be placed on three rows. Each row has a different number of keys (12, 11, and 10 keys respectively). He calculated the frequency for each letter and pairs of letters using large data sets of various Persian texts. As for the objective function, it measures the comfort of the user while typing. This is accomplished by: dividing the load equally between the two hands, maximizing alternating between hands, minimizing the number of times the same finger is used to type, and allocating the most frequent letters on the middle row. As for the algorithm, the regular GA operators (crossover, mutation and replacement) were implemented. However, the replacement operator was done using SA. If the new population is better than the old one, it will replace it; otherwise, it will be accepted depending on a certain probability.

Deshwal and Dep [3], designed an optimal Hindi keyboard layout using a GA. The aim was to reach a keyboard layout that is better than the currently used one. The criteria used are the same as the ones mentioned in Eggers et al. [5]. The selection methods they used were roulette wheel and tournament selections. As for mutation, they used swapping. Crossover is performed as follows: first, it copies parts of the keyboard from one parent to the offspring, then the rest of the keys are assigned their characters from the other parent. After reaching the best solution, a Local Search (LS) is applied. They implemented two kinds of LS:

1. *Repeated mutation*: Which is an LS with swap as a neighborhood move.
2. *Swapping shift-modified and actual characters*: An LS that starts from the first key of the keyboard and swaps the direct access character with the one that is accessed through a shift key, then evaluates the new keyboard. If the keyboard did not improve, the algorithm moves to the next key, and so on. When an improved keyboard is reached, the LS restarts the process again from the first key using the improved keyboard.

2.3 Arabic Keyboards

Malas et al. [13] applied a GA to design an m-finger Arabic keyboard layout that optimizes typing speed. In order to accomplish this, the characters' frequencies were needed. They collected articles from Arabic Wikipedia to make sure that the data is not biased towards a particular field. Then, they calculated the frequency of each character and also of pairs of characters. The characters were divided into two groups based on the frequency. The characters with the highest frequencies will be placed on the main grid, while characters with the lowest frequencies will be accessed through a shift key. The arrangement of the main grid characters will be determined based on the frequency of the pair of characters and their typing time. The attributes that have been considered are: hand transition, row transition, finger transition, row weight and finger weight.

As for the GA in [13], the initial population is generated randomly. Furthermore, to support diversity half of the selected individuals for reproduction are selected randomly. Only mutation was used to reproduce, because, according to the authors, they did not have a clear vision on how to apply the crossover. To mutate an individual, a number of the parent's keys are selected and swapped. Then, the mutated individual returns to the population, so that if it is fit enough, it will be selected in the next generation for reproduction. The algorithm is terminated manually when it reaches good results. The "shift" set was not optimized, because it contains the lowest frequency characters of the Arabic language. They assigned them manually to make it familiar to the user by placing them in similar locations to the common layout. Their layout design solves some of the common Arabic keyboard problems such as: placing the letter "ئ" in an awkward position (next to number 1), and it eliminated the combination of the not very frequent character pair "لا". Furthermore, this design increased the typing speed by 35 %.

Khorshid et al. [9], also applied a GA approach to optimize an m-finger Arabic keyboard layout. The Qur'an holy text was used to calculate the frequency of each character and pairs of characters. The objective function was calculated the same way as in Egger et al. [5]. The keyboard keys are represented by four indices (x_1, x_2, x_3, x_4), where:

$$x_1 = \left\{ \begin{array}{l} 0 \ \textit{no shift} \\ 1 \ \textit{use shift} \end{array} \right\}, \quad x_2 = \left\{ \begin{array}{l} 0 \ \textit{right hand} \\ 1 \ \textit{left hand} \end{array} \right\},$$
$$x_3 = \left\{ \begin{array}{l} 0 \rightarrow 7 \ \textit{left hand column} \\ 0 \rightarrow 8 \ \textit{right hand column} \end{array} \right\},$$

and x_4 represents the rows 0–5, where 0 is the top row and 5 is the bottom row. Also, this representation is mapped to the position of the keys. The keys are numbered from 1 to 34. At the beginning, each character is given an index, which is equal to the key location. When the optimization process starts, the characters will be mapped using the 4D array. As for the GA, the initial population was generated randomly. They used crossover and mutation to reproduce, with crossover and mutation rates of 0.7, and 0.2 respectively. The population size was 500, and the number of generations was 200. In comparison to the currently used keyboard, their design improved typing speed by 36.3 %.

2.4 Single-Finger Keyboards

Li et al. [11] attempted to optimize movement time of key transition. Fitt's law [7], which is a model used to predict the time required for a human to move to a target area, was used to calculate the movement time. The next step was to calculate the frequency of occurrence of pairs of characters. For that, they used 15,000 of the most common words from the British National Corpus. They tackled this problem using two models:

1. *IP model*: In this model, the transition is fixed between any two keys, so the distance can be calculated using Fitt's law [7], or by just calculating the distance between the centers of the two keys. This algorithm took a lot of time to run, though, and did not reach a good solution.
2. *A two stage heuristic*: First, an exact search with a swap neighborhood move is performed. This process continues until there is no improved solution after 10 consecutive attempts. Then an SA algorithm is applied, which was able to reach a solution in less than a minute. They noticed that the vowels were placed close to each other, although their frequencies when they are paired are not high. Each vowel on its own has a high frequency, though.

Dell'Amico et al. [2] designed an optimized s-finger keyboard layout for the languages: English, Spanish, French, and Italian. The goal was to find the best allocation of the characters, where the average time it takes to write a text is the

minimum. Again, Fitt's law [7] was used to calculate the average time. Several meta-heuristics were developed to solve this problem, as follows:

1. *Local Search* (*LS*): this algorithm uses two neighborhood moves: one for finding a local optimum, which is based on reallocating characters to empty locations. Then a swap neighborhood move is used to improve the local optimum.
2. *Simulated Annealing* (*SA*): two SA algorithms were implemented: one uses a swap neighborhood move, while the other combines swap with moving a character to another location. Furthermore, before any change to the temperature, the LS algorithm is applied to the current solution.
3. *Tabu Search* (*TS*): two TS algorithms were implemented: the first one uses a neighborhood move that moves a character to another empty location, while the second combines the swap move with moving a character to an empty location. Furthermore, their LS algorithm is used to optimize the resulting solution.
4. *Variable Neighborhood search* (*VNS*): this VNS algorithm uses the *k-exchange* neighborhood. Experimentally they found that k must be between 3 and n, where n is the number of characters. Also, they applied their LS algorithm to optimize each neighborhood solution.
5. *Fast Ant System* (*FANT*): they implemented the FANT algorithm that Taillard [17] developed and used their LS algorithm to optimize the retrieved solution by the ants.

3 Proposed Approach

To the best of our knowledge, there is no previous published research on optimizing the s-finger Arabic keyboard. The first attempts seem to be two early versions of our research in [8, 14], where an SA and a GA algorithms were respectively developed to optimize the Arabic keyboard design for single-pointer applications. In this chapter, we extend the research of [8], and provide more details on the proposed approach and the results obtained. Our solution approach is divided into two stages: (1) Calculating the frequency of each Arabic letter and for pairs of letters, and (2) Optimization using a GA, which is intended to find the best allocation of letters in a 2D grid for the s-finger Arabic keyboard layout. In what follows, we explain both stages in detail.

3.1 Calculating the Frequency of Letters

In this stage, there are three steps that have been followed: collecting Arabic text, cleaning the text and calculating the letters frequencies.

1. *Collecting Arabic text*: To be sure that the calculated frequencies are as accurate as possible, the data must be chosen carefully. The data should be diverse

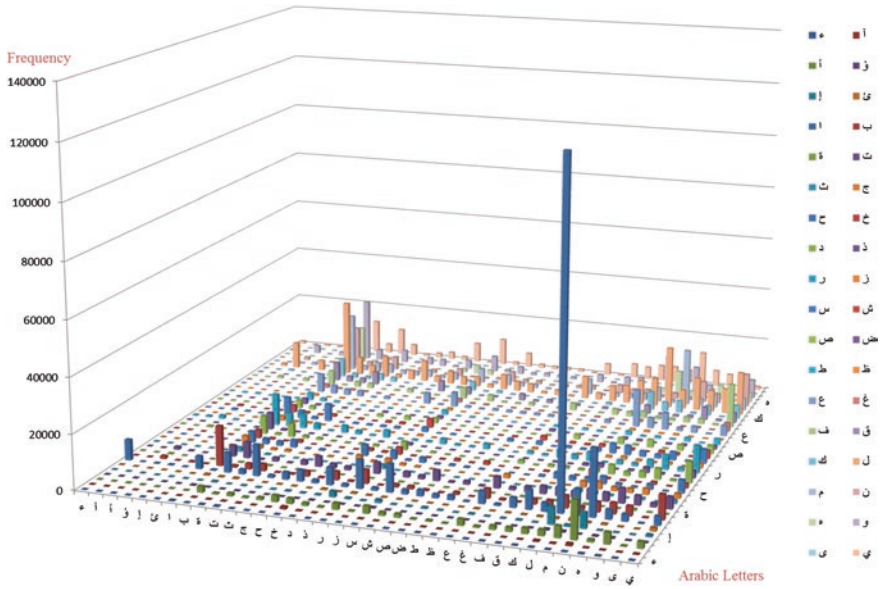


Fig. 1 Pairs of letters frequencies

and not inclined towards any particular filed or subject. So, Arabic Wikipedia seems a reasonable source of text to calculate the letters frequencies. The articles were downloaded from Wikipedia dump service [1] on September 9, 2013 with a file size of 1.6 GB.

2. *Cleaning the Arabic Wikipedia file*: The articles are downloaded in a single XML file that contains other meta information. Therefore, it was necessary to extract the Arabic text and prepare it before calculating the letters frequencies. NetBeans IDE 7.3 and the Regular Expression library in Java were used for this purpose. A total of 11,000 articles have been harvested.
3. *Calculating the frequency*: After collecting the articles, the frequency of each Arabic letter and the frequency of each pair of letters were calculated. Figure 1 shows the frequencies of pairs of Arabic letters obtained in this phase. As can be seen from the figure, the pair (ال) (meaning "the" in English) has the highest frequency in the Arabic language.

3.2 The GA Approach

As previously mentioned, the s-finger Arabic keyboard layout problem can be modeled in terms of the QAP. GAs are known to be successful in solving difficult optimization problems, like the QAP. In additions, for the keyboard layout

problem, a GA approach was used to optimize the m-finger Arabic keyboard in [9, 13], the m-finger Persian keyboard in [15], and the m-finger Hindi keyboard in [3]. Nevertheless, to the best of our knowledge, it has not been previously attempted for s-finger keyboards. In our proposed method, we try to utilize the success of GAs in solving similar optimization problems to solve the s-finger keyboard layout optimization. The steps of the classical GA framework are followed in this approach. Two crossover operators along with three types of mutation have been tried in our GA. In what follows, we explain the details of our proposed GA.

1. **Solution Representation**: To encode a solution for the s-finger keyboard problem, the following conditions should be satisfied: the keys are placed on a rectangular grid, all keys are of the same size, each letter must be assigned to only one key, each key must hold only one letter, and all the letters on the grid are different. It follows from these conditions that a 2D permutation encoding is a suitable representation for a problem solution. In our representation, each letter has a distinct integer value that represents it. As for the grid size, it is limited to cellphone sizes. The recently developed cellphones place the letters on a 3×10 or 3×11 grid; we use a 3×11 grid for our solution. Notice that the number of letters the grid can hold is 33, while the total number of the Arabic letters is 36. Hence, three letters were not included in the grid and will be accessed through a shift key.
2. **Objective Function**: To evaluate a solution (keyboard layout), three measures are taken into consideration in our objective function: (1) the frequencies of pairs of letters and their separating distances on the keyboard; (2) the frequency of a single letter and the row to which it is assigned; and (3) the "hit direction", which is typing from right to left in Arabic.

The first component corresponds to the QAP regular objective function, where the frequency of a letter pair (flow between facilities) is multiplied by the distance separating their assigned locations (the keyboard keys). The Euclidean distance is used to measure the distance between two keys i, j. The goal here is to place the most frequent letters close to each other. The first component of the objective function is shown in Eq. (2), where f_{ij} is the frequency of the letter pair, and d_{ij} is the distance between their assigned keys:

$$D_{ij} = f_{ij} \times d_{ij} \tag{2}$$

The second component considers to which row each letter has been assigned. Since the middle row is more convenient for typing, the most frequent letters should be placed on that row. As recommended in [13], each row is assigned a weight, and the frequency of a letter is multiplied by the weight of the row to which it is assigned. The selected weights are 1, 2 and 3 for the middle, upper, and bottom rows respectively, since the upper row is slightly more convenient for typing than the bottom row. Thus, if the frequency of occurrence of a letter i is f_i, and w_i is the row weight of the row assigned to letter i, this part of the objective function is represented as shown in Eq. (3):

$$R_i = f_i w_i \quad \text{where} \quad w_i = \begin{cases} y_i & \text{if } y_i = 3 \\ (3 - y_i) & \text{otherwise} \end{cases} \tag{3}$$

Finally, the Arabic language is written from right to left. To increase the typing speed this direction should be preserved as much as possible. Thus, when the most frequent pairs of letters are placed close to each other, the order of placement of the letters should support the typing direction from right to left. For example, the most frequent pair in the Arabic language is 'ال', so the letter 'ا' should be placed to the right of the letter 'ل'; this is called the "hit direction". To preserve the hit direction, a penalty is added when the pointer moves from left to right regardless of the row or column. This penalty is proportional to the frequency of the pair of letters. Equation (4) shows how this part of the objective function is formulated:

$$S_{ij} = f_{ij} v_{ij}, \quad \text{where } v_{ij} = \begin{cases} 0 \text{ if } x_i - x_j > 0 \\ 1 \text{ otherwise} \end{cases} \tag{4}$$

The overall score (objective) of the solution is shown in Eq. (5), where a, b and c are weights that determine the importance of each term.

$$F = \min a \sum_{i=1}^{n} \sum_{j=1}^{n} D_{ij} + \mathrm{b} \sum_{i=1}^{n} R_i + c \sum_{i=1}^{n} \sum_{j=1}^{n} S_{ij} \tag{5}$$

3. **Crossover Operators**: Two crossover methods were implemented, which are:

Two-Point Crossover (TPX): Two crossover points are randomly selected. Then the offspring will inherit the outer parts of the crossover points from the first parent, while the alleles in-between the crossover points will be inherited from the second parent. To inherit the letters from the second parent, the crossover starts from the bottom row. If the allele has not been inherited yet, it will be copied to the offspring; otherwise it will move to the next one, and then moving up row by row, as shown in the example in Fig. 2. The roles of parents are reversed to create a second child in the same way.

Fig. 2 Two-point crossover

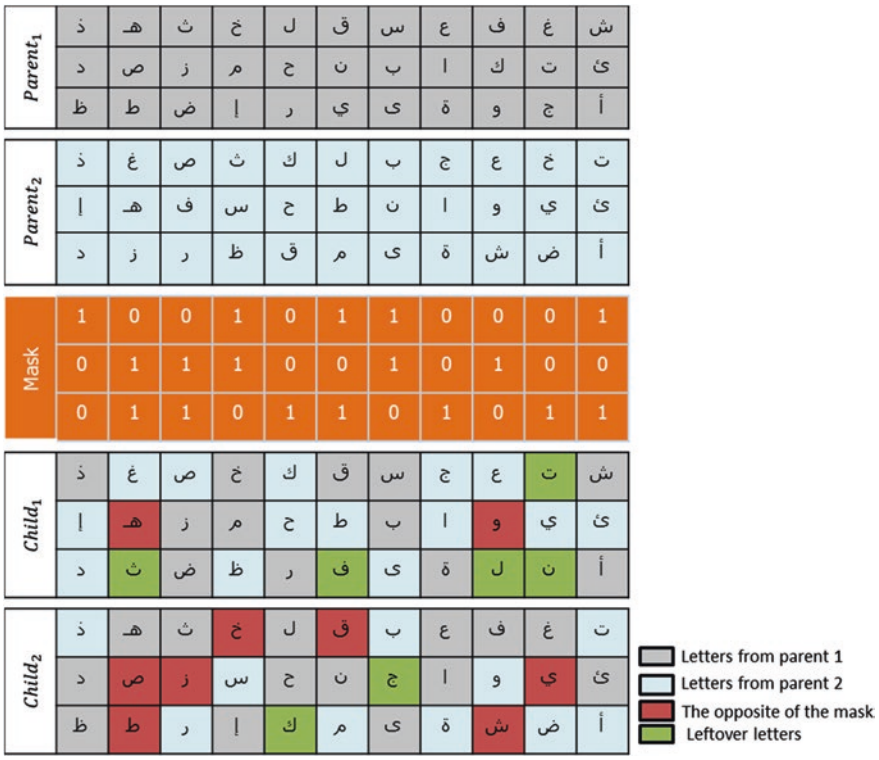


Fig. 3 Modified uniform crossover

Modified Uniform Crossover (MUX): This crossover is implemented based on the uniform order-based crossover mentioned in [18]. However, it is adapted to suit a 2D grid instead of a one dimensional permutation. In this crossover, two parents produce two children. A randomly generated mask is used to determine which alleles will be inherited from which parent. Starting from the first gene, if the mask value is 1, then child one will inherit the allele from parent one, while child two will inherit the allele from parent two. However, if the selected allele has previously been inherited, then the child will take the allele from the other parent, but if the allele of the other parent is already inherited, then the operator will leave the location empty and move on to the next location. After the end of this process, the letters that have not been assigned yet will be randomly distributed to the empty locations. See Fig. 3 for illustration.

4. **Mutation Operators**: Three mutation operators were implemented, which are:

Swap: two locations will be selected at random and their letters will be exchanged.

Insertion: a location will be selected at random and its letter will be inserted into another location in the same row, such that the rest of the letters will be shifted to accommodate the inserted letter.

Simulated Annealing (SA): the mutation operator is enhanced using an SA local search meta-heuristic. The neighborhood moves used in the *SA* were *swap* and *insertion* as explained above.

4 Testing and Computational Results

The Algorithm was implemented using Matlab r2012a. The initial population was created randomly, and the selection method used was roulette wheel selection. As for the replacement method, elitist replacement was used, such that the best individuals form the old generation replace the worst individuals from the new generation according to a certain replacement rate. The algorithm stops when no improvement is perceived after a certain number of iterations.

4.1 Parameters' Tuning

Before testing and evaluating the performance of the algorithm, a number of parameters should be tuned. After an extensive parameter tuning process, the final GA parameters were set as: 50 for the population size, 0.7 for the crossover rate, 0.2 for the mutation rate, 0.2 for the replacement rate, and the algorithm would stop after 20 iterations without improvement. As for the SA parameters, they were set to: 1,000 for the initial temperature, 10 iterations per temperature, and the cooling rate is 0.95. Moreover, there are three factors that affect the objective function of a solution, which are: the distance separating pairs of letters, the row weight, and the hit direction (see Eq. 5). A weight value ∈ [0–1] was used to determine the effect of each factor, such that the sum of the three weight values in the objective function is equal to 1.

In order to select the best weights, the most frequent pair of letters ("ال") was used as a guide in the tuning process. First, it is crucial to have the pair placed next to each other in the final keyboard layout, so the distance weight must be set to a high value to avoid separating the pair. Second, the pair should be placed in the middle row, so the row weight was gradually increased until the pair satisfied this condition. Third, the letter 'ا' should be placed to the right of the letter 'ل', so the weight of the hit direction was increased gradually until these letters were placed in the correct order ("ال") in the final solution obtained.

To achieve the above conditions, the tuning process started by setting the weights to the values: 1, 0, and 0 for the distance, row, and hit direction weights respectively. Then, the row weight was increased by 0.1, and the distance weight was decreased by the same value. If needed, this process will be repeated until the pair "ال" is placed in the middle row. After determining the row weight, the same process was done to determine the weight for the hit direction. So, the final objective function weights obtained were: 0.7, 0.1, and 0.2, for the distance, row, and the hit direction weights respectively.

4.2 Experiments

Each crossover method (two-point crossover, and modified uniform crossover) was tested with the three mutations, which are: swap, insertion, SA with a swap neighborhood move, and SA with an insertion neighborhood move. Each experiment was repeated 10 times. In order to see how much the initial solution improved, the averages of the objective value for the best solution in the initial population and best solution in the final population of each experiment (10 runs) were calculated. Table 1 shows the experimental results in terms of the average improvement in the objective function between the first and last generations in the 10 runs, the average processing time and the average number of generations.

It can be seen from Table 1 that the best improvement percentage was obtained from experiment 1 (TPX with swap mutation), followed by experiment 5 (MUX with swap mutation). Furthermore, the experiments showed that the TPX combined with swap mutation gave the best solutions in terms of the objective function value, followed by the MUX with swap mutation. It can also be noticed from the table that experiments 1 and 5 were the fastest among all other experiments (with the exception of experiment 2, which has apparently converged to a sub-optimal solution). Thus, it appears from the results that the simple swap mutation gave better results than both the insertion and the SA mutations. On the other hand, the worst results were produced when using the insertion mutation, either as a mutation or as a neighborhood move within the SA. In addition, it can be noticed that the SA mutation version was relatively slow in producing its best result.

Figure 4 also bears out these results by showing the best run from each experiment in terms of the value of the objective function in each generation. It can be seen from this figure that, in contrast to experiments 1 and 5 (which both rely on the swap mutation), all other experiments have experienced a rapid conversion towards a sub-optimal solution. This can be interpreted by the ability of the swap mutation to provide the best variation in the solution, which leads to avoiding premature conversion.

We also noticed that the keyboards with the best objectives all placed the letters of higher frequency in the middle row, and the letters (‘ا’, ‘ل’) were always placed

Table 1 Experimental results

Experiment	Crossover method	Mutation method	Improvement percentage (%)	Average time in sec	Average generations
1	*Two-point crossover (TPX)*	*Swap*	**26**	13.69	228
2		*Insertion*	23	7.38	126
3		*SA + swap*	18	98.65	63
4		*SA + insertion*	15	107.33	63
5	*Modified uniform crossover (MUX)*	*Swap*	**24**	23.621	181
6		*Insertion*	22	36.82	103
7		*SA + swap*	19	112.79	52
8		*SA + insertion*	20	95.17	57

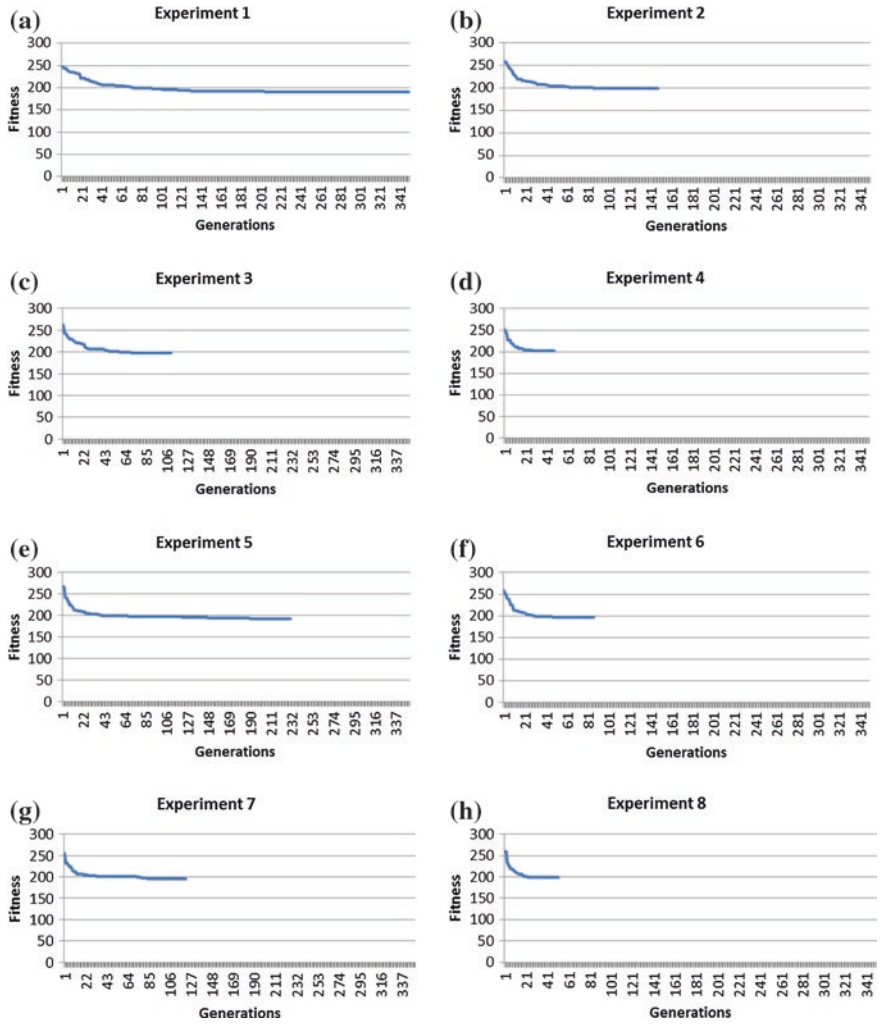


Fig. 4 Best run of each experiment. **a** TPX with swap, **b** TPX with insertion, **c** TPX with SA + swap, **d** TPX with SA + insertion, **e** MUX with swap, **f** MUX with insertion, **g** MUX with SA + swap, **h** MUX with SA + insertion

at the center. On the other hand, the letters with the lowest frequencies were always placed on the edges. Figure 5 shows the best keyboard layout, produced with experiment 5, in terms of the objective function value. To test the effectiveness of this layout, the keyboard shown in Fig. 5 was compared with other known keyboards, with respect to the criteria considered in this research. To do this, we assessed the Arabic keyboard layouts developed for m-fingers in Khorshid et al. [9], Malas et al. [13] and the iOS Arabic keyboard, using our objective function.

ذ	إ	أ	هـ	ن	م	ع	د	ق	ث	ض
ى	ص	ح	س	ل	ا	ي	ب	ة	ط	غ
ظ	خ	ش	ك	ت	و	ر	ف	ج	ز	ئ

Fig. 5 The optimized arabic keyboard

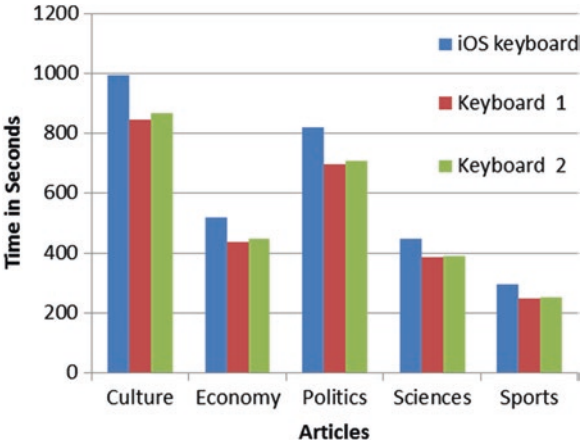


Fig. 6 Estimated average typing time

The results show that our keyboard layout is 58 % better than the keyboard in [9], 31 % better than the keyboard in [13], and 29 % better than the iOS Keyboard, in terms of improving in the objective function value.

The final experiment conducted was to evaluate the effect of the additional components of the objective function, which are the row weight and the hit direction. In order to do this, we repeated the best experiment (TPX with the swap mutation) but this time we considered only component 1 (i.e., the distance) in the objective function. We then evaluated the best keyboard obtained from the previous experiment shown in Fig. 5, having the complete objective function (we call this keyboard 1), with the best keyboard obtained from the second experiment (we call this keyboard 2). For this purpose, we implemented an algorithm that virtually estimates the speed of typing, using the distances between letters in a text. We considered 50 articles covering different disciplines (Culture, Economy, Politics, Science and Sports) with 10 articles from each. The articles were randomly chosen from the Arabic newspapers "Al Jazeera" and "Asharq Al-Awsat". We computed the total distance between the letters, and then the movement time needed to type a given text, using the keyboards mentioned above. The movement time is computed using Fitts' law [7]. Figure 6 shows the results of this experiment in terms of the

average typing speed of each keyboard. The figure also shows the estimated typing speed of the iOS keyboard. As can be seen from the figure, the results indicate that the average typing speed of keyboard 1 (obtained with the objective function that includes the distance, the row weight and the hit direction) is better than the average speed of keyboard 2 (obtained when only the distance is used in the objective function). Both were even better than the typing speed of the iOS keyboard. The overall averages obtained for all articles were (614.65, 522.85, 534.17) for iOS, keyboard 1, and keyboard 2 respectively. This seems to indicate that the measures added in the objective function actually improve the estimated typing speed, probably due to having a better arrangement of the most frequent letters on the keyboard as a result of the additional measures used.

5 Conclusion and Future Work

In this chapter we aimed to develop an optimized s-finger Arabic keyboard layout. The QAP was used as a model to solve this problem. An overview of different approaches for solving the multiple-finger and the single-finger keyboard layout problems for different languages has been presented. The contribution of this research is three fold. First, calculating the frequencies of Arabic letters using diverse articles from Wikipedia. Nonetheless, the frequency algorithm can be used on text from different sources, such as: twitter, newspapers, etc. Second, new measures in the objective function for optimizing the Arabic single-finger keyboard were introduced, which are: the row weight and the hit direction. These measures can be easily customized to take into account the user's convenience and preferences. Third, a new GA approach with different genetic operators was implemented and thoroughly tested.

The experimental results showed that a simple swap mutation works well with both crossover types, and was also faster than the other mutation operators tested in this research. No benefit was realized from augmenting the GA with SA as a mutation operator. Furthermore, the objective function factors were tested to see the impact of considering the row weight and the hit direction on a solution. The results indicate that the added measures have a positive effect on improving the virtual typing speed of the keyboard, as compared to considering the distance only.

Since the results of this research have been only evaluated from a theoretical point of view, for future work, we aim to test the resulting keyboard on actual users in terms of both convenience and the typing speed. We hope that these results can be adopted for applications that rely on typing using one pointer, and used by people with special needs.

References

1. Anonymous: Arwiki dump progress. http://dumps.wikimedia.org/arwiki/. (2013)
2. Dell'amico, M., Díaz, J., Iori, M., et al.: The single-finger keyboard layout problem. Comput. Oper. Res. **36**, 3002–3012 (2009)
3. Deshwal, P., Deb, K.: Design of an Optimal Hindi Keyboard for Convenient and Efficient Use. Indian Institute of Technology, Kanpur (2003)
4. Dickey, J., Hopkins, J.: Campus building arrangement using topaz. Transp. Res. **6**, 59–68 (1972)
5. Eggers, J., Feillet, D., Kehl, S., et al.: Optimization of the keyboard arrangement problem using an Ant Colony algorithm. Eur. J. Oper. Res. **148**, 672–989 (2003)
6. Elshafei, A.: Hospital layout as a quadratic assignment problem. J. Oper. Res. Soc. **28**, 167–179 (1977)
7. Fitts, P.M.: The information capacity of the human motor system in controlling the amplitude of movement. J. Exp. Psychol. **47**, 381–391 (1954)
8. Hosny, M.I., Alswaidan, N., Najjar, A.: An optimized single-finger arabic keyboard layout. In: Proceedings of the Science and Information Conference 2014 (SAI '14), London, U.K (2014)
9. Khorshid, E., Alfadli, A., Majeed, M.: A new optimal Arabic keyboard layout using genetic algorithm. Int. J. Design Eng. **3**, 25–40 (2010)
10. Koopmans, T., Beckmann, M.: Assignment problems and the location of economic activities. Econometrica **25**, 53–76 (1957)
11. Li, Y., Chen, L., Goonetilleke, R.: A heuristic-based approach to optimize keyboard design for single-finger keying applications. Int. J. Ind. Ergon. **36**, 695–704 (2006)
12. Light, L., Anderson, P.: Typewriter keyboards via simulated annealing. AI Expert **8**, 20–27 (1993)
13. Malas, T., Taifour, S., Abandah, G. Toward optimal arabic keyboard layout using genetic algorithm. In: Proceedings of 9th International Middle Eastern Multiconference on Simulation and Modeling, Amman, Jordan (2008)
14. Najjar, A.B.: Toward an optimized arabic keyboard design for single-pointer applications. In: Proceeding of the Fifteenth Annual Conference Companion on Genetic and Evolutionary Computation Conference Companion, pp. 1717–1718. ACM, Amsterdam (2013)
15. Samimi, N.: Optimization of farsi letter arrangement on keyboard by simulated annealing and genetic algorithms. Majlesi J. Multimedia Process. 1(3), (2012)
16. Steinberg, L.: The backboard wiring problem: a placement algorithm. SIAM Rev. **3**, 37–50 (1961)
17. Taillard E (1998) FANT: fast ant system. In: Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale
18. Talbi, E.: Metaheuristics: from Design to Implementation. Wiley, Hoboken (2009)

Dynamic Well Bottom-Hole Flowing Pressure Prediction Based on Radial Basis Neural Network

Paras Q. Memon, Suet-Peng Yong, William Pao and Jion Sean Pau

Abstract Reservoir simulation provides information about the behaviour of a reservoir in various production and injection conditions. Reservoir simulator is used to predict the future behaviour and performance of a reservoir field. However, the heterogeneity of reservoir and uncertainty in the reservoir field cause some obstacles in selecting the best calculation of oil, water and gas components that lead to the production system in oil and gas. This paper presents a dynamic well Surrogate Reservoir Model (SRM) to predict reservoir bottom-hole flowing pressure by varying the production rate constraint of a well. The proposed SRM adopted Radial Basis Neural Network to predict the bottom-hole flowing pressure of well based on the output data extracted from a numerical simulation model in a considerable amount of time with production constraint values. It is found that the dynamic SRM is capable to generate the promising results in a shorter time as compared to the conventional reservoir model.

1 Introduction

Multiphase flow in oil and gas field generally refers to simultaneous flow of more than one fluid in a reservoir [1] and it is commonly encountered as the flow of oil-gas-water is the reservoir. The physics involved in multiphase flow is very intricate due to the interaction between different fluids [2] and it is generally experienced during the production of oil and gas field. This indicates that oil and gas field is

P.Q. Memon · S.-P. Yong (✉)
Computer Information and Sciences, Universiti Teknologi Petronas, 31750 Tronoh, Malaysia
e-mail: yongsuetpeng@petronas.com.my

W. Pao · J.S. Pau
Department of Mechanical Engineering, Universiti Teknologi Petronas,
31750 Tronoh, Malaysia
e-mail: william.paokings@petronas.com.my

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_17

getting more complicated and challenging, having one or more reservoirs and many numbers of injection and production wells in its geological structure. The complexity of a reservoir leads the system to a dynamic nature and the recovery process changes from natural depletion to water-flooding as well as switches to an enhanced oil recovery process [3]. In order to deal with this complexity, large amount of researches has been done and performed with the introduction of mechanistic models [4] to predict the fluids and gas flow by changing their pressure and temperature value etc. This phenomena also leads to many numerical simulation models that are used as potential candidates to estimate the accurate flow characteristics of oil, gas and water in the production system [2]. Numerical simulation models that are used to simulate the behavior of production system in multiphase flow reservoir requires considerable amount of time on some parallel computer processing units. On the other hand, surrogate reservoir model (SRM) can also be considered as a potential solution to address this necessity. SRM can be used to predict the results of a reservoir, such as pressure, production rate and gas-oil ratio in less amount of time as compared to the other numerical simulation models. The objective of this paper is to develop a dynamic well SRM to predict the bottom-hole flowing pressure (BHFP) of a well based on production rate constraints that mines the output data from reservoir model. In real scenario the BHFP of a well always changes based on time period. This is because sometimes the reservoir and petroleum engineers prescribe (fix) the value of a well BHFP for some specific years and they do not want to put the well BHFP for those years into consideration. In order to cater for this kind of common scenarios, dynamic well SRM is proposed in this paper. Dynamic well SRM has the capability of producing the results of BHFP for all or specific years.

The structure of this paper is as follows: Sect. 2 explains the related work of SRM in oil and gas fields, Sect. 3 details the development of the proposed dynamic well SRM, while Sect. 4 shows a case study which is under consideration of this research and Sect. 5 spells out the results and discussion, followed by a conclusion in Sect. 6 of this paper.

2 Related Work

In 1999, the first surrogate reservoir model (SRM) was developed for hydraulic fracturing simulator in oil and gas. It was able to reproduce the results of hydraulic fracturing simulator called FracPro in less amount of time using artificial intelligence technique [5]. In later years, other attempts of SRM can be found in literature articles. For instance, SRM was able to calculate the porosity and permeability distribution in a heterogeneous and multiphase reservoir by matching the static and dynamic data that are available [6]. The surrogate model was also developed for Steam Assisted Gravity Drainage process in heterogeneous and multiphase petroleum reservoir [7]. Based on the previous work on surrogate model, another SRM was built for a giant oil in the Middle East, in that development

full field simulation model was taken, which includes millions of grid blocks and more than 165 wells in its geological structure. The SRM was able to replicate the results of full field simulation model based on time complexity [5, 8, 9]. Besides that, SRM was also developed for uncertainty analysis of coalbed methane (CBM) production to optimize the performance of reservoir [10]. In the same year, well SRM was developed to examine against the two-and-a-half year production of a reservoir. The SRM was used to accurately predict the simultaneous cumulative oil production and water cut for every well at each given time [11]. Pertaining to that work, SRM was also built for CBM reservoir to predict the cumulative production, that includes thirteen well to produce fifteen years production [12]. In subsequent years, well based SRM was developed for a reservoir that includes both natural and hydraulic fractures. SRM has been used to optimize the recovery process and predict the cumulative oil production [13, 14]. On the other hand, SRM was also used to predict the pressure and CO_2 distribution throughout the reservoir with good accuracy [15]. Recently, well based SRM was developed to generate a production rate as a function of time for all wells over the next 25 years with promising accuracy [16].

The success of SRM development is due to the state of the art technology in Artificial Intelligence, such as Artificial Neural Network (ANN). And the use of the ANN has been increasing in oil and gas industry over the past few years to solve many complex and highly non-linear problems [17]. ANN is considered as a non-linear tool and are good at predicting the complex and nonlinear system behavior. ANN is also used to solve many different kinds of problems related to reservoir engineering, such as, reservoir characterization [18], permeability prediction [19, 20], prediction of bottom-hole flowing pressure in vertical multiphase flow [21, 22], predicting the water inflow performance in solution gas drive [23]. In the past few years, some of the ANN study has been done on the history matching process [24] and the application of surrogate reservoir modeling [9, 25]. The benefit of ANN over other conventional techniques such as, response surface and reduced models in reservoir engineering, is its ability to perform complex and highly non-linear task accurately and rapidly. In most of the previous work related SRM, researchers have adopted backpropagation neural network (BPNN) in constructing the reservoir model. However, in the study of BPNN, there is a problem of trapping in local minima during training time because one needs to specify the number of hidden neurons in the network. In most of the time, the network will not reach at the global minima to find the minimum error value. However, another type of ANN which is known as radial basis neural network (RBNN), in which the non-existence of local minima problem will not occur because the number of hidden neurons increases automatically until the error value reaches its minimum value, which is considered as more objective based.

In this paper, we propose to adopt RBNN after comparisons were done on the performance of BPNN and RBNN, to construct the dynamic well SRM based on production rate constraint which defer from the previous developed SRM. The developed dynamic well SRM may cater the changes with respect to each time step on the specific task. Dynamic well SRM is considered as a complex task to be

implemented because it generates the reservoir response based on the time complexity which may vary in an inconsistent pattern. Dynamic well SRM is used to calculate the results of BHFP by prescribing the production rate value at all or some specific time steps.

3 Dynamic Well Surrogate Reservoir Model

Dynamic well surrogate reservoir model (SRM) is the collection of reservoir well constraints such as, well bottom-hole flowing pressure (BHFP, p_{wf}) and production rate, which changes with respect to the time span. In most of the real scenarios, the BHFP of a well always changes based on time period and sometimes reservoir and petroleum engineers have already known the value of a well BHFP for some specific years and hence they do not need to calculate the well BHFP for those specific years but only consider those unknown period of time. In order to predict BHFP for such a scenario, dynamic well SRM can be used to predict BHFP for all or specific years. In this paper, a dynamic well bottom-hole flowing pressure with prescribed production rate is developed. Equation 1 represents the BHFP value of a well using the BOAST simulator and well SRM with production rate is used to replicate the result of BHFP using Eq. 1. In the equation, p_{wf} represents the bottom-hole flowing pressure of well, Q_o represents the production rate, PI represents a productivity index of the reservoir's ability to transfer fluid to the well, λ_o shows mobility of the oil phase, B_o explains the volume factor of oil phase and p represents reservoir pressure.

$$p_{wf} = p + \frac{B_o}{PI \cdot \lambda_o} Q_o \tag{1}$$

Figure 1 represents the steps involved to build the surrogate reservoir model (SRM) for this paper. The first step to build the SRM requires data collection. In this paper, we use Black Oil Simulation Tool (BOAST) to build a spatio-temporal database and generate the output responses based on the input values which are tuned to the BOAST simulator.

BOAST is developed and provided by the Department of Energy (DOE) United States in 1982 as an open source package. It is considered as an implicit pressure-explicit saturation (IMPES) simulator [26], which finds the pressure distribution for a given time step first then calculates the saturation distribution for same time step. It is a three dimensional (X, Y, Z) and three phase (oil-gas-water) simulator for modeling the multiphase flow in porous channel and used in oil and gas field to simulate different scenarios. For example, primary (natural) depletion, secondary depletion in which pressure is maintained by water injection and tertiary depletion is considered as enhanced oil recovery such as gas injection is used to maintain the pressure. The well model in BOAST has the flexibility to change the operational constraints such as production rate specifications or well flowing pressure value on the well behavior and performance, and the user is permitted to add or replicates the wells during the simulation time [27].

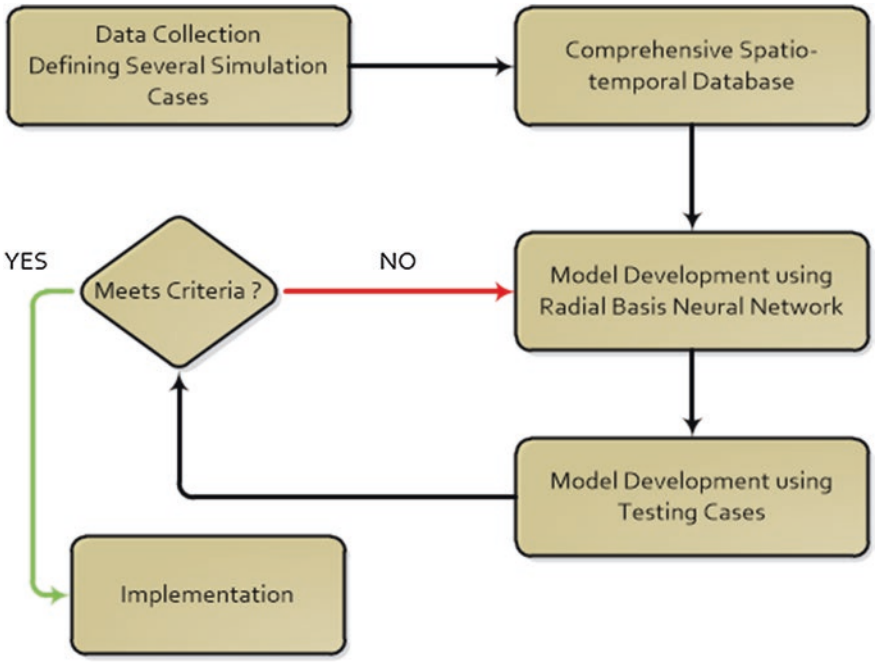


Fig. 1 Basic flow chart to build surrogate reservoir model

The spatio-temporal database represents the characteristics and behavior of the reservoir with its input-output parameters, which is considered as the training data sets for SRM. It is developed using the static and dynamic data, such as, porosity, permeability, pressure and production value at any time the reservoir.

Before the training starts in RBNN, all the training datasets can be divided into two matrices. One is assumed to be input data sets and another is considered as the output data sets. The input and output data sets are normalized in a specific range. In this research, a standard normalization function such as tangent sigmoid function is used to confine all the input and output data sets within the specific range of -1 and 1 before the training starts [28, 29]. The mathematical representation of this function is given in Eq. 2:

$$y = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{2}$$

During the training of RBNN, the training data set is always divided in three phases: training, validation and testing. The training data is used during the training of the neural network where as validation data is also used during the training process, but it is not used to train the neural network rather it is used to check the network learning during the training. Both training and validation data that used during the training time is considered as non-blind data [16].

3.1 Radial Basis Neural Network

Radial Basis Neural Network (RBNN) is used to develop a network with good generalization capabilities having a less number of hidden neurons in its structure [30]. A RBNN is considered as the special type of the ANN because it only requires one hidden layer in its architecture and it allows the input space to be represented in a new space with different hidden layer neurons. During the training process of RBNN, it behaves as a linear model because all the hidden neurons center and computations are fixed. The RBNN hidden layer neurons perform nonlinear transformations and maps all the inputs into new input space satisfactory. The output layer is considered as linear transformer, which is applied to new input space so that only weights of hidden neurons can be adjusted. The performance of the RBNN can be determined by adjusting the centers (widths) of the hidden neurons and there is no specific formula available to select the width of the radial basis function (RBF). But one should select the width of the RBF larger than the distance between two adjacent inputs and smaller than the distance all over the input space in order to get good generalization [31]. RBNN has been used in wide range of applications, such as, system prediction, pattern recognition, system approximation, signal processing and system equalization, system identification, speech recognition and adaptive control, etc. [32]. And it has been used to solve the problems of oil and gas field, i.e. gas-oil ratio (GOR) of reservoir, seismic, electromagnetic, resistivity, [33], well log data inversion [34], prediction of log properties from the seismic attributes of the reservoir [35] as well as the nonlinear relationship between the reservoir property and seismic attributes [36].

The growth and general architecture of the RBNN has been influenced by the RBF. Figure 2 represents the general architecture of the RBNN. In Fig. 2, $x = [x_1, x_2, \ldots, x_m]$ represents the input vectors of the network and $y = [y_1, y_2, \ldots, y_n]$ represents the final net output of the network and in hidden layer there are a number of neurons. Inside each hidden neurons there is a RBF [37], the RBNN inputs are directly connected to the each basis function that generates an output Φ_i as shown in the Eq. 3, which depends upon the input vectors.

$$\phi_i = \exp\left[-\frac{(\parallel x - u \parallel)^T(\parallel x - u \parallel)}{\sigma^2}\right] \tag{3}$$

where, x represents the input data points of network, u is the center of the radial basis function ($u = 0$), σ represents the radius of the RBF ($\sigma > 0$). Once the hidden neuron is calculated based on radius of RBF. Then it is passed to the output layer, where the sum of the product between the hidden layer neuron and weight vector is computed to produce final network output y_n.

$$y_n = \sum_{m=1}^{M} w_i \cdot \phi_i \tag{4}$$

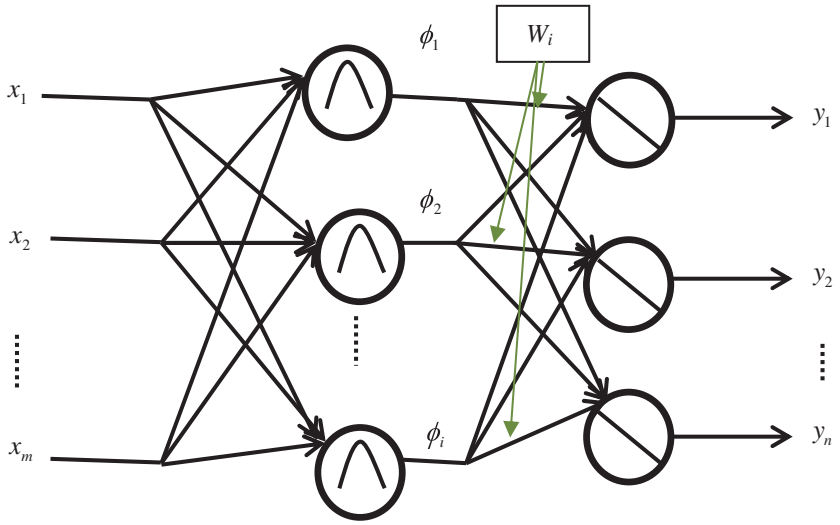


Fig. 2 Radial basis neural network with m-dimensional inputs and n-dimensional outputs

Once the model is developed using the training, it is ready to be tested with testing data, that the network never uses during the training process. This testing data is used to check the predictive capability of constructing a network [38]. If the network generalized this testing data with good accuracy, it means that the network has a capability to predict the output of new data with good approximation and this network is considered as a validated network model to serve the SRM.

4 Case Study

The base case study, which is considered in this paper has been taken from the Society of Petroleum Engineering (SPE) [39] to build the dynamic well SRM with production rate constraints. Figure 3 represents the grid view and configuration of the reservoir which is under consideration. In the figure, each grid block consists of 1,000 ft and the reservoir consists of 10×10 grid block in x and y directions. The total area occupied by the reservoir is considered as 100,000,000 ft^2. Figure 4 reveals the diagonal cross section model of the reservoir properties.

The reservoir model is based on the three layers labeled as LAYER 1, LAYER 2 and LAYER 3. The 8,325 ft is considered as the top value of the model, while H, FT represents the depth of the reservoir, which varies as 20, 30, 50 ft in LAYER 1, LAYER 2 and LAYER 3 respectively. Φ represents the porosity value of the reservoir, which is assumed to be homogeneous in whole reservoir. It represents the tiny spaces in the rock that hold oil or gas and is measured of total rock which is taken up by pore space [40]. Kx, Ky and Kz represents the permeability value of

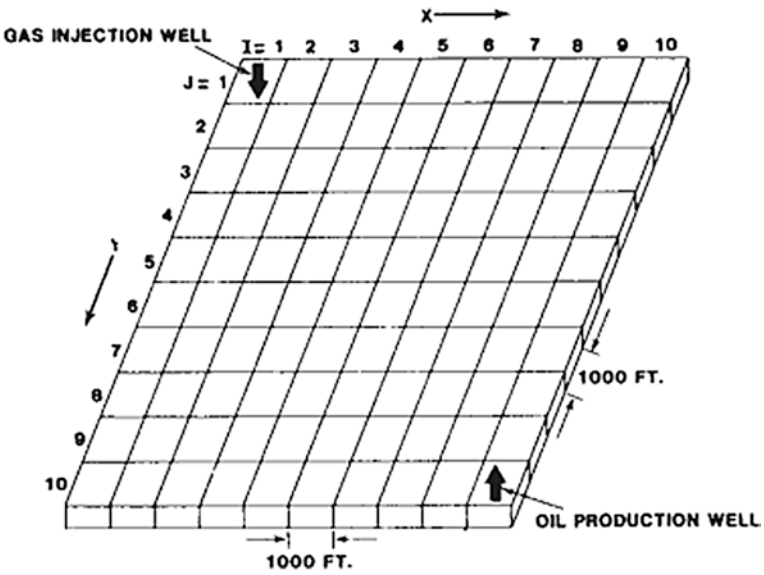


Fig. 3 Grid configuration and problem specification to build well SRM [39]

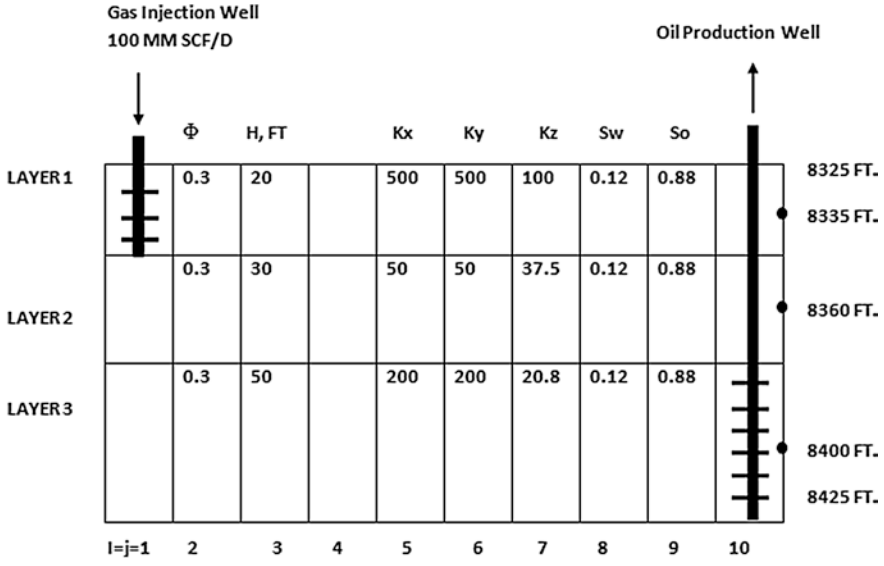


Fig. 4 Diagonal cross section of reservoir grid model [39]

reservoir in *x*, *y* and *z* direction respectively. The value of permeability is considered as homogeneous in *x* and *y* direction, but heterogeneous with respect to *z* direction. Permeability is considered as the ability of a reservoir to pass the fluid from the

Table 1 Reservoir characteristics used to build database for dynamic well SRM

Input parameters	Range
Porosity in Layer 1 (%)	2–4
Porosity in Layer 2 (%)	2–4
Porosity in Layer 3 (%)	2–4
Permeability in Layer 1 X direction (md)	470–530
Permeability in Layer 1 Y direction (md)	470–530
Permeability in Layer 1 Z direction (md)	70–130
Permeability in Layer 2 X direction (md)	40–60
Permeability in Layer 2 Y direction (md)	40–60
Permeability in Layer 2 Z direction (md)	30–45
Permeability in Layer 3 X direction (md)	170–230
Permeability in Layer 3 Y direction (md)	170–230
Permeability in Layer 3 Z direction (md)	15.83–25.83
Production rate (STB/D)	4,000–7,000

rocks' pores and permeability of each rock depends on the nature of the reservoir. *Sw* and *So* from Fig. 4 represents the initial water and oil saturation of the reservoir before the production come out from the reservoir. The value of initial water and oil saturation is considered as constant in the whole reservoir, that are 0.12 and 0.88 respectively. There is a one gas injection well in LAYER 1 at this first grid block of the reservoir with injection rate of 100,000 MMscf/D. While there is another well which is known as the production well of the reservoir and it is perforated in the LAYER 3 at the opposite corner of the gas injection well. The production well can produce a maximum production rate of 20,000 STB/D and minimum production rate of 1,000 STB/D. The minimum bottom-hole flowing pressure of production well is 1,000 psi. The middle layer is considered as empty because there is no oil, gas and water in this layer. The aquifer value of the reservoir is zero, therefore there is no flow at the boundaries of the reservoir grid system. Whereas the reservoir has an initial pressure of 4,800 psia and temperature is considered as 200 °K. This paper contains the porosity, permeability and production rate as the key input parameters to build the dynamic well SRM for multiphase flow simulation. Table 1 represents the base case study values and their range, which are considered to build the database for the proposed study. The mean value of input parameters such as porosity and permeability range is the same as the value of the base case study.

5 Results and Discussion

This section explains the results of the developed dynamic well SRM for bottom-hole flowing pressure (BHFP) with reservoir characteristics such as, porosity, permeability and production rate constraint values as input parameters. 100 training cases were generated from BOAST and 5-fold cross validated was conducted

Table 2 Quantitative measurements of 5-fold cross validation from 100 training cases using BPNN and RBNN

Model	RMSE	MAPE	σ	Accuracy (%)
BPNN	0.2283	3.5767	0.0135	87
RBNN	0.2059	3.2271	0.0136	**96**

Best result is indicated in bold

Table 3 Quantitative error measurement of 15 testing cases using RBNN

Algorithm	RMSE	MAPE	Accuracy (%)
BPNN	2786.4	695.5	63
RBNN	**7.9**	**14.66**	**64.5**

Best result is indicated in bold

towards the developed BPNN and RBNN. Table 2 shows the average results from the 5-fold cross validation training data. The statistical results such as root mean square error (RMSE), mean absolute percentage error (MAPE), standard deviation (σ) and accuracy were presented in the table. It is shown that RBNN performs better with 96 % accuracy as compared with BPNN only can achieve 87 % accuracy.

Once the training is conducted to build a static well SRM, 15 series of test cases were generated randomly to test the dynamic well SRM. Table 3 show the results of dynamic well SRM. The error values such as RMSE, MAPE and accuracy are calculated to measure the error between the target and predicted output. It again shows that RBNN outperforms BPNN with slight better accuracy of 65 %, however, there is quite a huge gap between on the error rates between the two techniques as shown in the table, albeit RBFF still performs better than the other. The time that the BOAST simulator takes to calculate one simulation run (test case) is about 1 min and for 15 simulation runs, it took a total of 15 min on the Intel (R) Core (TM) i5-3470 CPU @ 3.2 GHz CPU, whereas SRM using ANN only took a maximum of 1 s to compute the results for the all simulation runs (15 test cases) on the same CPU with a good approximation.

Once the dynamic well SRM is built according to the specified condition such as with production rate constraint. Then it is used to test developed SRM according to the end user requirement. To test the dynamic well SRM by changing the production rate constraint of a well at any time. Table 4 represents the arbitrary test cases values to test the dynamic well SRM by switching the constraint values.

Case 1 from Table 4 is used to predict the results of bottom-hole flowing pressure (BHFP) using BPNN and RBNN approaches. Figure 5 represents the results of dynamic well SRM by giving the production rate constraint value as input parameter to calculate the BHFP for year 1, 2, 4, 6, 8, 9 and 10. And for year 3, 5, 7 the BHFP value is already prescribed. However, RBNN and BOAST results matches with each other by accuracy of 63 %. Whereas, BPNN and BOAST results matches with each other by accuracy of 47 %.

Case 2 from Table 4 is used to produce the results of BHFP for year 1, 2, 5, 6, 9 and 10 by giving the value production rate as input parameter using dynamic well SRM, but for year 3, 4, 7 and 8 the BHFP is already prescribed as shown in the Fig. 6. However, RBNN and BOAST matches with each other by accuracy of 98.9 % and BPNN and BOAST results matches with each other by accuracy of 97.8 %.

Table 4 Reservoir characteristics used to build well SRM with production rate constraint

Input parameters	Case 1	Case 2
Porosity in Layer 1 (%)	4	4
Porosity in Layer 2 (%)	4	3
Porosity in Layer 3 (%)	2	3
Permeability in Layer 1 X direction (md)	511	524
Permeability in Layer 1 Y direction (md)	511	519
Permeability in Layer 1 Z direction (md)	114	72
Permeability in Layer 2 X direction (md)	42	43
Permeability in Layer 2 Y direction (md)	57	42
Permeability in Layer 2 Z direction (md)	37	31
Permeability in Layer 3 X direction (md)	185	206
Permeability in Layer 3 Y direction (md)	218	177
Permeability in Layer 3 Z direction (md)	20	23
Production rate (STB/D)	5,546	5,854

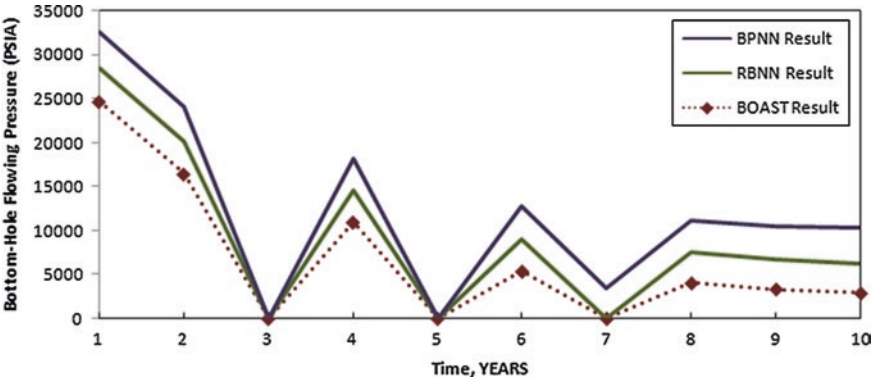


Fig. 5 Testing results of well SRM with production rate (Case 1)

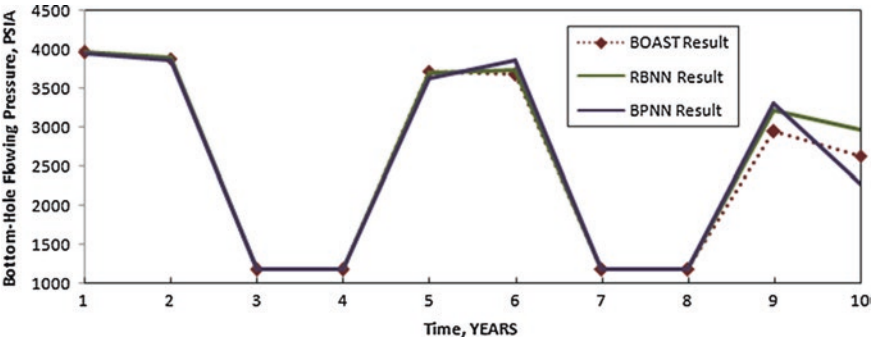


Fig. 6 Testing results of well SRM with production rate (Case 2)

From the stated cases, case 1 represents a case performed at the lower tile whereas case 2 shows a case with performance at the higher tile.

6 Conclusion

In this paper, results of dynamic well SRM with production rate constraint have been presented with two different types of ANN, i.e. RBNN and BPNN. From the case study conducted, RBNN outperforms BPNN in building the dynamic SRM. Also, two statistical error measurements have been conducted to see the absolute error of target and predicted output of the trained network. The study have carried with both BPNN and RBNN to build the SRM in order to predict the future results in less amount of time. It is also obvious dynamic well SRM has a capability of fast and accurate replication of numerical simulation models results at different time steps. The future work and challenge will involve complex reservoir with larger numbers of grid blocks in its geological structure with many number of injection and production wells to optimize the production of reservoir.

Acknowledgment The authors also like to thank Universiti Teknologi PETRONAS for sponsoring the project funding under YUTP-EOR MOR.

References

1. Pourafshary, P.: A Coupled Wellbore/Reservoir Simulator to Model Multiphase Flow and Temperature Distribution. ProQuest, USA (2007)
2. Ferro, S.P., Goldschmit, M.B.: A numerical model for multiphase flow on oil production wells. In: Latin American and Caribbean Petroleum Engineering Conference Society of Petroleum Engineers (2007)
3. Jiang, Y.: Techniques for modeling complex reservoirs and advanced wells. Ph.D. thesis, Stanford University (2007)
4. Roux, A., Corteville, J., Bernicot, M.: Wellsim and pepite: accurate models of multiphase flow in oil wells and risers (1988)
5. Mohaghegh, S., Popa, A., Ameri, S.: Intelligent systems can design optimum fracturing jobs (1999)
6. Queipo, N.V., Salvador, P., Rincon, N., Contreras, N., Colmenares, J.: Surrogate modeling-based optimization for the integration of static and dynamic data into a reservoir description. In: SPE Annual Technical Conference and Exhibition Society of Petroleum Engineers Inc, Dallas. Copyright 2000 (2000)
7. Queipo, N.V., Goicochea, J.V., Salvador, P.: Surrogate modeling-based optimization of sagd processes (2001)
8. Mohaghegh, S.D., Hafez, H., Gaskari, R., Haajizadeh, M., Kenawy, M.: Uncertainty analysis of a giant oil field in the middle east using surrogate reservoir model. In: Abu Dhabi International Petroleum Exhibition and Conference Society of Petroleum Engineers, Abu Dhabi (2006)
9. Mohaghegh, S.D., Modavi, A., Hafez, H., Haajizadeh, M., Kenawy, M., Guruswamy, S.: Development of surrogate reservoir models (SRM) for fast-track analysis of complex

reservoirs. In: Intelligent Energy Conference and Exhibition Society of Petroleum Engineers, Amsterdam (2006)
10. Jalali, J., Mohaghegh, S.D.: Reservoir simulation and uncertainty analysis of enhanced CBM production using artificial neural networks. In: SPE Eastern Regional Meeting. Society of Petroleum Engineers, Charleston (2009)
11. Hafez, N.A., Haajizadeh, M., Guruswamy, S., Mohaghegh, S.D., Modavi, A.: Development of surrogate reservoir model (SRM) for fast track analysis of a complex reservoir. Int. J. Oil Gas Coal Technol. **2**(1), 2–23 (2009)
12. Mohaghegh, S.D., Jalali, J., Gaskari, R.: Coalbed methane reservoir simulation and uncertainty analysis with artificial neural. Chem. Chem. Eng. **17**, 65–76 (2010)
13. Kalantari Dahaghi, A., Esmaili, S., Mohaghegh, S.D.: Fast track analysis of shale numerical models. In: SPE Canadian Unconventional Resources Conference Society of Petroleum Engineers, Calgary (2012)
14. Kalantari Dahaghi, A., Mohaghegh, S.D.: Numerical simulation and multiple realizations for sensitivity study of shale gas reservoir. In: SPE Production and Operations Symposium Society of Petroleum Engineers, Oklahoma (2011)
15. Amini, S., Mohaghegh, S.D., Gaskari, R., Bromhal, G.: Uncertainty analysis of a CO_2 sequestration project using surrogate reservoir modeling technique. In: SPE Western Regional Meeting, Society of Petroleum Engineers Bakersfield (2012)
16. Mohaghegh, S.D., Liu, J.S., Gaskari, R., Maysami, M., Olukoko, O.A.: Application of surrogate reservoir models (SRM) to am onshore green field in saudi arabia; case study. In: North Africa Technical Conference and Exhibition. Society of Petroleum Engineers, Cairo (2012)
17. Mohaghegh, S.: Virtual-intelligence applications in petroleum engineering: Part 1 artificial neural networks. J. Petrol. Technol. **52**(9), 64–73 (2000)
18. Mohaghegh, S., Arefi, R., Ameri, S., Aminiand, K., Nutter, R.: Petroleum reservoir characterization with the aid of artificial neural networks. J. Petrol. Sci. Eng. **16**(4), 263–274 (1996)
19. El-Sebakhy, E.A., Asparouhov, O., Abdulraheem, A.A., Al-Majed, A.A., Wu, D., Latinski, K., Raharja, I.: Functional networks as a new data mining predictive paradigm to predict permeability in a carbonate reservoir. Expert Syst. Appl. **39**(12), 10359–10375 (2012)
20. Tahmasebi, P., Hezarkhani, A.: A fast and independent architecture of artificial neural network for permeability prediction. J. Petrol. Sci. Eng. **86**, 118–126 (2012)
21. Mohammadpoor, M., Shahbazi, K., Torabi, F., Reza, A., Firouz, Q.: A new methodology for prediction of bottomhole flowing pressure in vertical multiphase flow in iranian oil fields using artificial neural networks (anns) (2010)
22. Osman, E.S.A., Ayoub, M.A., Aggour, M.A.: Artificial neural network model for predicting bottomhole flowing pressure in vertical multiphase flow. In: SPE Middle East Oil and Gas Show and Conference Society of Petroleum Engineers, Kingdom of Bahrain (2005)
23. PAlrumah, M., Startzman, R., Schechter, D., Ibrahim, M.: Predicting well inflow performance in solution gas drive reservoir by neural network (2005)
24. Ramgulam, A., Ertekin, T., Flemings, P.B.: An artificial neural network utility for the optimization of history matching process. In: Latin American and Caribbean Petroleum Engineering Conference. Society of Petroleum Engineers (2007)
25. Carnevale, C., Finzi, G., Guariso, G., Pisoni, E., Volta, M.: Surrogate models to compute optimal air quality planning policies at a regional scale. Environ. Model Softw. **34**, 44–50 (2012)
26. Arastoopour, H., Hariri, H.: Analysis of two-phase flow in a tight sand gas reservoir, using boast (1986)
27. Bujnowski, S.W., Fanchi, J.R., Harpole, K.J.: Boast: a three-dimensional, three-phase black oil applied simulation tool (version 1.1). Bartlesville Project Office U.S. department of energy Bartlesville, Oklahoma (1982)
28. Sampaio, T.P., Ferreira Filho, V.J.M., Neto, A.D.S.: An application of feed forward neural network as nonlinear proxies for use during the history matching phase. In: Latin American and Caribbean Petroleum Engineering Conference Society of Petroleum Engineers, Cartagena de Indias (2009)

29. Vogl, T.P., Mangis, J.K., Rigler, A.K., Zink, W.T., Alkon, D.L.: Accelerating the convergence of the back-propagation method. Biol. Cybern. **59**(4–5), 257–263 (1988)
30. Musavi, M.T., Ahmed, W., Chan, K.H., Faris, K.B., Hummels, D.M.: On the training of radial basis function classifiers. Neural Networks **5**(4), 595–603 (1992)
31. Williams, M.: Application of artificial neural networks in the quantitative analysis of gas chromatograms. Master's thesis (1996)
32. Yingwei, L., Sundararajan, N., Saratchandran, P.: Radial basis function neural networks with sequential learning. World Scientific Publishing Co.. Inc., Singapore (1999)
33. Kaftan, I., Salk, M.: Determination of structure parameters on gravity method by using radial basis functions networks case study: Seferihisar geothermal area (western turkey). In 2009 SEG Annual Meeting. Society of Exploration Geophysicists (2009)
34. Huang, K.Y., Shen, L.C., Weng L.S.: Well log data inversion using radial basis function network. In Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International, pp 4439–4442. IEEE (2011)
35. Russell, B.H., Hampson, D.P., Lines, L.R.: Application of the radial basis function neural network to the prediction of log properties from seismic attributes—a channel sand case study (2003)
36. Li, L., Wei, X., Shifan, Z., Wan, Z.: Reservoir property prediction using the dynamic radial basis function network (2011)
37. Gengaje, S.R., Alandkar, L.S.: Prediction of survival of burn patient using radial basis function network (1988)
38. Edara, P.K.: Mode choice modeling using artificial neural network. Master's thesis, Virginia Polytechnic Institute and State University (2003)
39. Odeh, A.S.: Comparison of solutions to a three-dimensional black-oil reservoir simulation problem. J. Petrol. Technology **33**, 13–25 (1981)
40. Are, S., Dostert, P., Texas, A.M., Ettinger, B., Liu, J., Sokolov, V., Wei, A.: Reservoir model optimization under uncertainty (2006)

Modeling Energy Consumption in a Educational Building: Comparative Study Between Linear Regression, Fuzzy Set Theory and Neural Networks

Henrique Pombeiro and Carlos Silva

Abstract Quantifying the impact of energy saving measures on a given space requires representative models that can describe how energy is consumed in that space with dependence on known input variables. For this purpose, it is commonly accepted that linear regressions can be used to define those models, named energy consumption baselines. In this paper, we want to assess the performance of linear regressions to model electricity consumption compared to other modeling techniques that can capture nonlinear dynamics like fuzzy and neural networks models in three experimental places in a Portuguese University campus: a set of offices in a department, a classroom amphitheater and the library. Five input variables were defined for the study: day type, occupation, day length, solar radiation and heating and cooling degree days. The novelty of this paper is the comparative assessment between these different modeling techniques, which are usually addressed individually in the literature. From the results obtained in this research, we can outline the importance of selecting representative input variables, study their inter relation, fine tuning the models, and analyze the different models when being trained and tested. We generally conclude that neural networks have the best performance values, fuzzy models increase their performances when trained with varying epochs (with the exception of the amphitheater, where the model over fits and so as the testing performance) and linear regressions present the lowest performance. Hereupon, we discuss the encouragement of applying non-linear models such as the presented ones rather than traditionally used linear regression models, when evaluating consumption baseline to determine energy savings.

Keywords Baseline model · Energy savings quantification · Linear regression · Fuzzy · Neural networks · Service buildings

H. Pombeiro (✉) · C. Silva
IN+ Centre for Innovation, Policy and Research, Instituto Superior Tcnico,
University of Lisbon, Tagus Park - Av. Prof. Dr. Cavaco Silva, Porto Salvo, Lisbon, Portugal
e-mail: henrique.pombeiro@tecnico.ulisboa.pt

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_18

1 Introduction

Energy modeling is a research field that has increasing for the last years, standing as an essential step towards the increase of efficiency in the energy systems value chain [1].

The building sector is responsible ca. 40 % of the final energy worldwide, which means that this is the largest energy consumer worldwide [2]. With the increase in urbanization mainly in the poorest countries, it is expected that buildings continue to have a high impact in the worldwide consumed energy. For the EU, the increase of energy efficiency in 20 % is one of the three targets for 2020 [3], especially in the building sector, which has the highest energy saving and energy efficiency potential [4].

Increasing energy efficiency in buildings requires planning and implementing energy efficiency measures. Quantifying the impact of a given measure—the energy savings—requires modeling how energy evolves with the surrounding variables. A complete methodology for quantifying energy efficiency measures can be found in the International Performance and Verification Protocol (IPMVP), further adapted to the eeMeasure for the application in European Commission funded projects [5]. In these standards, the application of linear regression is suggested, as it can be easily implemented using spread sheets. However, these models do not capture more complex nonlinear dynamics that can be found in office buildings like the university campus. Thus, in the literature, modeling tools that can capture nonlinear dynamics, as fuzzy set models or neural networks are suggested as alternative methods.

We could not find a quantitative comparison between the most well accepted model by the IPMVP (linear regression) with other nonlinear and under development models, which have been increasingly implemented in energy modeling: fuzzy set theory and neural networks. This is one of the main purposes of this paper: to understand how the accuracy is increased at the expenses of using much more complex models that are difficult to develop and implement without specific software tools.

This paper addresses the implementation of three models (linear regression, fuzzy set theory and neural networks) in three experimental spaces each (a class amphitheater, a library and an set of offices), assessing the development of the models and their performance.

Section 2 explains the theoretical fundamentals of the three models, enlightening their applications in the energy field. Section 3 describes the methodological approach on the data treatment and the development and performance evaluation of the models. Section 4 presents the results from the models and their discussion. Finally, this paper finishes with general remarks and future work to be developed.

2 Energy Consumption Models

Several references can be found in the bibliography tackling energy modeling in buildings. This section describes the methods used in this paper.

2.1 Linear Models

Linear regressions can determine the degree of similarity between two datasets by measuring the quadratic error between each point. It gives a linear model with coefficients that try to explain those relations. The usual performance index that is used is the squared error: R^2 [10].

The linear regression models are quite well accepted in the definition of the baseline, namely according to the IPMVP standards. However, a linear model has a limited capacity as there are variables can only be explained with non-linear correlations [11].

A system can be considered *linear* if the relation between inputs and outputs can be described by linear equations, i.e. if the outputs can be explained by aggregating the inputs, each being multiplied by a corresponding coefficient. The *principle of superposition* in an important theorem that can explain the properties of a linear system, which states that the influence of all the inputs acting simultaneously in the system output is the same as the sum of the influence of the sum of each input acting alone [6].

$$c_1 x_1 + c_2 x_2 = c_1 y_1 + c_2 y_2 \tag{1}$$

A linear system can be computed through linear regressions. Conceptually, a linear regression wishes to minimize the total sum of squares (SST), which is equal to the addition of the error of sum of squares (SSE) with the regression sum of squares (SSR).

$$\begin{gathered} \sum_{i=1}^{n} (y_i - \overline{y})^2 = \sum_{i=1}^{n} (y_i - \widehat{y_i})^2 + \sum_{i=1}^{n} \\ SST = SSE + SSR \end{gathered} \tag{2}$$

A linear model has the advantage that both the performance and the statistical significance can be easily studied with the coefficient of determination R^2 and the *p-value,* respectively. R^2 is the result of:

$$R^2 = \frac{\text{Explained variation of y}}{\text{Total variation of y}} = \frac{SSR}{SST} \tag{3}$$

R^2 varies from [0; 1], corresponding 1 to the higher correlation. The *p-value* is a factor that has to be equal or lower than α, which we have defined as 0.05. If p is higher than 0.05, then we cannot reject the null hypothesis, which means that we cannot guarantee that the results were not generated by chance. This, however, is a reflection of statistical confidence and, even choosing a model with parameters higher than α, other studies can be undertaken to see if the model is actually adequate or not. As an example, we can check if the model is bias, i.e. if the model tends to underestimate or overestimate the output, by calculating the median of the generated outputs.

2.2 Fuzzy Set Theory

We find in the literature the application of Fuzzy models in the Energy field for the development of optimization problems and also for the development of energy models. Fuzzy models are adequate when trying to model systems where conventional models are not precise enough, when there is a high degree of uncertainty, or there is a strong non-linear behavior or even when there is time varying characteristics [11]. Fuzzy logic can be described as an approximation of human classification and reasoning, which gives a high interpretability of this type of models [12].

Dounis and Caraiscos [13] undertook a literature review in the energy system analyzes and energy demand modeling. Namely, they have identified the application of hybrid uncertainty models that use hybrid fuzzy stochastic models for regional energy systems planning and management.

Modeling regional energy systems is an important issue especially for designing regional policy. Beyond this, we can find the application of fuzzy models in energy modeling at a lower scale. Zhibin and Jiuping [14] applied fuzzy set theory to deal with uncertainty for the cost optimization in the application of a combined heat and power model. Babuska [12] also found very important applications of fuzzy set theory in modeling energy and comfort in a building environment. Moradi et al. [15] developed interesting applications of building energy modeling through fuzzy set theory. Finally, [12, 15] found that applying fuzzy modeling together with neural networks results in neuro-fuzzy modeling, robust self-learning models could be developed. Overall, fuzzy set theory can be considered important and feasible for the implementation of a more efficient and adequate building energy management system [16, 17].

Fuzzy set theory has been developed in the past years, belonging to a computational philosophy named soft computing, aiming at dealing with complex intelligent systems. On opposition to a crisp data set, a fuzzy set adds a membership degree of a given input to a given set. Taking a simple example of having two glasses of a transparent liquid: on one glass we have a label depicting that the probability of that liquid is deadly poison is 0.1, and on the other glass the label depicts that the liquid has a deadly poison membership of 0.9. While we have 90 % of chances to survive if drinking from the first glass, we know that we will not die if we drink from the second one, since the liquid does not belong 100 % to the deadly poison membership; so we would have a stomach pain but not a deadly one.

From the previous example, we understand that fuzzy modeling applies a degree of gradual transitions between sets, which can be very helpful when we want to design an intelligent decision above predetermined antecedent parameters.

A fuzzy system is processed from the inputs (antecedents) to the outputs (consequents). In order to develop such a system, there is a very important step which is also in common for the development of any model (e.g., linear or neural network): the ***inputs generation***. In fact, the quality of the inputs that we feed the model are crucial for the model to be accurate. From the inner quality of the measured data (e.g. accuracy of the device, data gaps, outliers, etc.) to the clustering

and relevance of the inputs; this step involves the comprehension of what we are modeling and if we find the input data relevant for the exercise, and also if the variables have any relation between themselves.

Having a treated input data, we have to determine ***which set of variables*** and the ***universe of discourse*** that will be used to model the problem. On this issue, we have to define which data will be used for training and which for testing the model. Usually, a 60/40 % or a 50/50 % ratios are used. In this paper, the second one has been chosen.

A further step in building a fuzzy system is the ***fuzzification***, which includes the definition of the membership functions and the fuzzy rules for the rule base. Membership functions are usually built with clustering algorithms. In this paper, we have used fuzzy c-means (FCM) to generate the membership functions used in the inference engine of the fuzzy models. FCM are partition data algorithms forming overlapping sets based on pattern similarities. A generalization of hard c-means is given by the following equations [7, 8].

Given the following data set:

$$x_k = [X_{1k}, X_{2k}, \ldots, X_{nk}]^T \in \mathbb{R}^n, \quad k = 1, \ldots, N \tag{4}$$

The fuzzy partition matrix (having the membership functions for the objects x) and the cluster centers are found.

$$U = \begin{bmatrix} \mu_{11} & \cdots & \mu_{1N} \\ \cdots & \cdots & \cdots \\ \mu_{c1} & \cdots & \mu_{cN} \end{bmatrix}, \mu_{ij} \in [0, 1] \tag{5}$$

$$V = V_1, \ldots, V_c, V_i \in \mathbb{R}^n \tag{6}$$

The process undergoes by repeating the following processes, either by initializing U or V, assuming the partition matrix is fixed:

$$V_i = \frac{\sum_N^{k=1} \mu_{ik}^m X_k}{\sum_{k=1}^N \mu_{ik}^m} \tag{7}$$

Then the distances from the cluster centers are calculated and the partition matrix is updated, assuming that the cluster centers are fixed.

$$d_{ik}^2 = (X_k - V_i)^T (X_k - V_i) \tag{8}$$

$$\mu_{ik} = \frac{1}{\sum_{j=1}^C (\frac{d_{ik}^2}{d_{jk}^2})^{1/(m-1)}} \tag{9}$$

The process finalizes when the stopping criteria is satisfied, which can be:

$$\|\delta U\| < \epsilon \tag{10}$$

The ***fuzzy rules*** for the rule base are fired at this point. By combining the different inputs throughout their membership degree in each adjudicated membership function, the model then applies the inference operators to choose the decisions that compose the outputs. Common inference operators are Kleene-Dienes, Lukasiewicz, Mamdani or Sugeno (or Tagaki-Sugeno). For this paper, Sugeno type inference system was chosen.

While a Mamdani-type fuzzy inference systems (FIS) computes the output consequence with a membership function as the rule strength, followed by a *defuzzification* process to reach for a membership degree, a Sugeno FIS gives a crisp or a linear equation as an output [7, 8]. The overall output is a weighted average of the individual rule outputs and given by:

$$\hat{y} = \frac{\sum_{k=1}^{n} w_k y_k}{\sum_{k=1}^{n} w_k} \tag{11}$$

The development of a fuzzy model finishes with its ***fine tuning***. If the general output does not have a satisfactory performance, the parameters can be adjusted, such as the number of clusters, a new selection of variables (leaving some aside or including other that were not previously chosen) or training the model varying the number of epochs (iterations).

2.3 Neural Networks

Neural networks (NN) try to apply the human physiological brain reasoning in the development of models. They model the human brain as a continuous-time nonlinear dynamic system. With different weights that can be applied to the artificial neurons, adaptive models can be developed [7].

Babuska [12], Moradi et al. [15] have also performed a detailed review on the application of several energy models with artificial networks, with particular relevance for the ability of self-learning that NN can provide. Further, Kalogirou [18] has developed artificial NN to predict energy consumption in a building and they proved faster convergences than simulated dynamic programs.

NN to mimic how the human brain reasons, simulating neurons connected between themselves (through "synapses"), iteratively learning the best combination of weights to be given at each input in order to outcome the most fitted output. NNs are a powerful instrument for dealing with complex systems such as perception, pattern recognition, ability to learn from examples and adaptability and fault tolerance [8].

Given the above features, NN are generally used when input and/or output are multidimensional, when the mathematical structure of the system is unknown and, at the same time, when the interpretability of the model is not required. In fact, NN act as a "black box", reasoning through several iterations across the nodes (functionally representing neurons in the human brain), giving weights to each variable under a structure that is not understood by the developer [7].

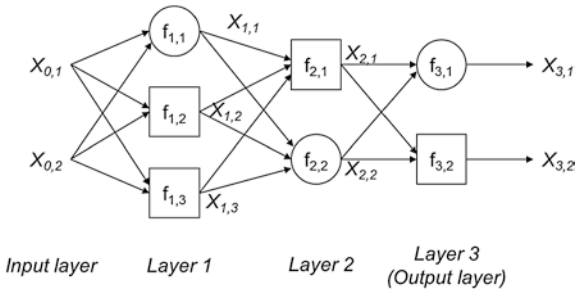


Fig. 1 Representation of a feedforward adaptive network with two hidden layers (adapted from [7])

In this paper, we present the application of an adaptive NN with a *feedforward* architecture, as represented in Fig. 1.

Jang [7] described in a very complete way how a NN is designed. As depicted in Fig. 1, an adaptive network is a structure composed by nodes which are connected by directional vectors, being each node a processing point and the connector the causal relationship between them. The output from each node depends on the input parameters, conferring them, in this way, adaptiveness skills. The way the model output is compared with the real output is called learning rule, which is represented by a mathematical expression. We can define the error measure as the sum of squared errors between the training and the desired for the pth output as:

$$E_p = \sum_{N(L)}^{k=1} (d_k - x_{L,K})^2 \tag{12}$$

where d_K is the kth component of the the pth real output and $x_{L,K}$ is the modeled output.

The basic learning rule of adaptive NN is the steepest descent method, which was used by Rumelhart et al. in [9], naming backpropagation learning rule to this procedure.

Figure 1 gives us the understanding that a feedforward backpropagation network has a unidirectional relationship between inputs and outputs, in contrast to the other possible architecture: recurrent NN. Having the configuration of parameters and the learning rule that was chosen, the model will try to minimize the distance between its outputs with the real ones. The modeler will try different configurations of the NN in order to have the most desirable performance, changing the number of hidden layers, the number of nodes (neurons), the learning rule and, as in any other model, changing the input variables.

3 Methodology

Underpinning the general goal of developing the most adequate model that can describe how electricity consumption varies with given inputs in three experimental places, this section describes the methodology undertaken in this work.

3.1 Data Treatment

Three experimental spaces are addressed in this paper: a class amphitheater, a library and a set of offices composed by 11 offices, In a University building in Lisbon, Portugal. The electricity consumption of both has been monitored and so it was possible to gather consumption data for the following periods:

- Class amphitheater: 25-02-2013 to 20-06-2013
- Library: 18-03-2013 to 05-09-2013
- Offices: 26-03-2013 to 04-09-2013

The amphitheater has a fixed class schedule from Mondays to Fridays. The exams period begins in May 25th, which is a period with no classes and therefore with no consumption. With no occupation and null consumption values, a high correlation between occupation and consumption would be achieved with such a data set. In order to eliminate the weekends and holidays effect in the model, in which there is no consumption and so this would bias the model, these days were also eliminated. This space has no direct access to the exterior. It has an exterior wall, two interior walls and an interior wall that points to a common lobby of the building.

No data was eliminated in holidays, weekends and exams periods for the library data set as this space is operating with occupation of students and so there is still variable consumption in those days, although two rooms of the library are closed in those days and also from 18h00 to 09h00 from Mondays to Fridays. However, after the post-exam period, August 2th, the library presented very low consumption values because the whole building was closed for two weeks.

The offices are occupied by research staff, gathering PhD students, administrative staff, teachers, researches and management. With the exception of the administrative staff, the remainders benefit from a certain schedule freedom, which gives a non-routine occupation pattern.

Seven input variables were considered for the development of the models: *day type, occupation, day length, average temperature, solar radiation, and HDD/CDD* with a fixed temperature at 15 °C.

Day type is a variable that was defined by the authors. This reflects the expected usage intensity of the spaces. Regarding the class amphitheater, this variable is the reflection of the class schedule that is predetermined in before the semesters begin. The days are normalized from [0; 1], respectively from the day with the lowest to the highest number of classes:

$$\hat{x} = \frac{x - min}{max - min} \tag{13}$$

where $\hat{x}$ is the normalized output, x the real output, min the minimum value for the outputs and max the maximum one. All variables were normalized in this way.

The measured consumption data regards the power plugs, illumination and ventilation. Heating and air conditioning for those rooms is provided by a chiller and Air Treatment Units, which were not considered for this study.

3.2 Models

In this paper, we have applied a modeling methodology that comprises the data treatment, parameters definition and fine tuning of the model (corresponding to iteratively change the input parameters of the models).

We have applied multivariate linear regressions, trying all possible combinations of inputs and choosing the model with the highest correlation factors and statistical significance. The fuzzy models are the Sugeno-type, with variation of inputs and fine tuning it by choosing the number of clusters and training the models with different epochs. The NN models have the feedforward backpropagation architecture, varying the number of hidden layers (0–2) and number of neurons in each (1–5). Each model was trained with 50 % of the data set, in an alternated order, and tested with the remaining 50 %. The overall performance of the models was quantified with:

Mean absolute error (MAE) is used to quantify the mean error between the modeled (f_i) and real (y_i) outputs across all entries of the model (n) given by:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i| \tag{14}$$

Mean squared error (MSE) is the quadratic loss between the modeled and the real outputs, accounting for the estimator variance and, thus, its bias.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2 \tag{15}$$

Median gives an extra perception on bias. If different from zero (positive or negative), it depicts that the model is bias (overestimation or underestimation, respectively). Median is any real number that satisfies the following:

$$P(\leq m) \geq \frac{1}{2} \text{ and } P(X \geq m) \geq \frac{1}{2} \tag{16}$$

Absolute error (AE) gives the absolute information that MAE gives, providing the total error across the modeled period.

$$AE = n * MSE = n\sum_{i=1}^{n}|f_i - y_i| \tag{17}$$

Variance accounted for (VAF) describes the similarity between two data sets (in this case, the output from the model and the real one).

$$VAF_i = (1 - \frac{var(y_i - \hat{y}_i)}{var(y_i)}) * 100\ \% \tag{18}$$

4 Results and Discussion

This section addresses the main results from the application of the different models to the experimental places. Table 1 depicts the input parameters which were used to fine tune the models, thus with highest performances.

Consumption in the three spaces is better explained without all inputs, having a performance decrease when adding the remaining inputs. Day type and occupancy are the common inputs, with the exception of linear regression in the classroom since the *p-value* is above 0.05, thus we cannot reject the null hypothesis. The class amphitheater is an underground space, thus it does not have access to natural light and is well insulated from the external temperature. The library has little use of natural light as well.

Analyzing the linear regression models (Eqs. 18–20), we can assess that day type has the highest weight, therefore the most important one, followed by occupation and, for the offices, average temperature and day length.

$$kWh_{amphitheater} = 0.21 + 0.66Day\_t \tag{19}$$

$$kWh_{library} = 0.12 + 0.71Day\_t + 0.12Occp \tag{20}$$

$$kWh_{offices} = 0.33 + 0.37Day\_t + 0.23Occp - 0.12Day\_l - 0.18\hat{T} \tag{21}$$

Training the Sugeno-type fuzzy models in the library and offices improved its performance, as depicted in Table 2, with the highest levels for 5 and 15 epochs, respectively. Finally, the feedforward backpropagation NN models had the highest performances when using two hidden layers, each having 5, 10 and again 5 neurons respectively for each space.

Table 1 Models parameters: input variables (Day_t—day type; Occp—Occupation/h; Day_l—Day length [h]; $\hat{T}$—Average T [°C]), Eps—epochs in the fuzzy models training, Hddn_lyr—hidden layers in the NN models, and Nrn—nr of neurons in each Hddn_lyr

		Day_t	Occp	Day_l	$\hat{T}$	Eps	Hddn_lyr	Nrn
Amphitheater	Linear regression	x						
	Fuzzy	x	x					
	Trained fuzzy	x	x			–		
	Neural network	x	x				2	5 + 5
Library	Linear regression	x	x					
	Fuzzy	x	x					
	Trained fuzzy	x	x			5		
	Neural network	x	x				2	10 + 10
Offices	Linear regression	x	x	x	x			
	Fuzzy	x	x	x	x			
	Trained fuzzy	x	x	x	x	15		
	Neural network	x	x	x	x		2	5 + 5

Table 2 Performance indicators for the developed models (MAE, MSE, AE—absolute error, RE—relative error, median, VAF and R^2) with respect to the experimental places (amphitheater, library, and offices)

			MAE	MSE	AE	RE (%)	Med	VAF (%)	R^2 (%)
Amphitheater	Lin. regression	Train	1.73	4.14	53.7	15.9	−0.1	83.5	40.2
		Test	**1.90**	**6.37**	**58.8**	**16.1**	**−0.8**	**37.6**	**5.2**
	Fuzzy	Train	1.75	4.14	54.1	16.0	−0.2	84.3	40.3
		Test	**1.91**	**6.41**	**59.1**	**16.2**	**−0.6**	**37.5**	**5.4**
	Trained Fuzzy	Train	–	–	–	–	–	–	–
		Test	**–**	**–**	**–**	**–**	**–**	**–**	**–**
	Neural network	Train	1.41	3.44	43.7	13.0	0.9	48.1	29.8
		Test	**1.59**	**3.91**	**49.3**	**13.5**	**0.3**	**69.1**	**20.7**
Library	Lin. regression	Train	13.36	350.0	1,135.2	8.9	−1.8	95.2	90.2
		Test	**14.11**	**417.7**	**1,199.4**	**9.4**	**−1.3**	**93.6**	**87.9**
	Fuzzy	Train	13.36	348.3	1,135.9	9.0	−2.1	95.3	90.3
		Test	**14.36**	**428.6**	**1,220.3**	**9.5**	**−1.2**	**93.5**	**87.6**
	Trained fuzzy	Train	9.23	205.4	784.2	6.2	−0.1	96.6	94.3
		Test	**13.71**	**377.5**	**1,165.1**	**9.1**	**−1.3**	**94.5**	**89.2**
	Neural network	Train	10.17	258.6	864.1	6.8	−1.1	95.7	92.9
		Test	**12.71**	**322.1**	**1,080.2**	**8.4**	**−2.8**	**95.3**	**91.0**
Offices	Lin. regression	Train	2.96	14.04	240.0	15.1	−0.4	88.5	69.3
		Test	**3.06**	**14.54**	**247.7**	**15.8**	**0.4**	**87.4**	**66.0**
	Fuzzy	Train	2.96	14.04	240.0	15.1	−0.4	88.5	69.3
		Test	**3.05**	**14.48**	**246.8**	**15.7**	**0.4**	**87.4**	**66.0**
	Trained fuzzy	Train	1.73	5.16	140.4	8.8	0.1	95.3	88.7
		Test	**2.76**	**11.77**	**223.9**	**14.3**	**0.2**	**90.0**	**73.8**
	Neural network	Train	2.11	8.05	170.9	10.8	−0.1	92.1	82.4
		Test	**2.41**	**9.30**	**195.0**	**12.4**	**0.0**	**91.5**	**78.5**

Table 2 depicts the performance results of the best models for each space, outlining the values for models train and test. The results analysis can be undertaken together with Fig. 2, where we can see the consumption profiles generated by the highest performance models and the real consumption profiles for all spaces. Regarding the amphitheater, further training the fuzzy model varying the number of epochs provides a significant over fitting and, therefore, considerable lower performances, even resulting in negative VAFs, which means that no similarity exists between the modeled and the real profiles.

The lowest performances are achieved in the amphitheater consumption models, with higher values in the training of the model for the linear regression and fuzzy models, but with higher performances for the test of the NN model. This may happen due to low relation between inputs and the output (electricity consumption). In fact, the highest consumption types are concerned to illumination, a projector and the lecturer's laptop, which may vary with the type of class (with different occupations) but occupation by itself has been seen to decrease the performance of the model, which can also be explained by the normal usage of the

Fig. 2 Graphical representation of the modeled and the real electricity consumption for the three experimental spaces, concerning all models

same illumination intensity regardless the class has 50 or 10 students. Hereupon, a randomness factor plays an important role in consumption behavior.

The highest performance levels are achieved in the library models, overpassing VAFs of 93 % and an R^2 of 91.0 % with the feedforward backpropagation NN.

Nevertheless, since this is a high intensive consumption space, the minimum kWh/day of error that we could achieve was 12.71 kWh/day.

Regarding the offices, quite acceptable performance values have been reached, overcoming VAFs of 90 % for the training fuzzy model and the NN model, with corresponding R^2 of 73.8 and 78.5 %.

All models seem to be just slightly bias, being those corresponding to the offices the lowest bias. Generally, we can argue that linear regression models are the ones which depict the lower performances for each parameter, being the feedforward back-propagation NN models the ones with the highest performance, thus being considered the most adequate to tackle electricity consumption in this experimental space.

The relative errors decrease the most in the NN networks, which can also be related to the total kWh of error that decrease and also to monetary expenditure. Drawing a simple exercise, by considering a 0.10 €/kWh rate, NN models would confer a decrease of 0.03 €/day, 1.40 €/day and 0.65 €/day, respectively for the class amphitheater, library and offices. Arguing that the development of NN models require a higher level of expertise and access to higher level software such as Matlab® (which has been the main software used by the authors), maybe the investment in these modeling capabilities should have a return of investment for an intensive service building. At the end of the year, we can estimate that the total decreased error from linear regressions to NN models would be around 650 kWh only for these three experimental spaces.

This work is a development that has been undertaken from the one presented in [19], where there was not applied the training of the fuzzy models, data was not normalized in [0; 1], the offices had not been considered and the input variable Day type had not been developed, which is in fact the most relevant for this experimental setup. We can see that results highly improved with these experimental steps, leading to models with considerable better performances, although the class amphitheater model is still far from what we desire but, as explained before, this may be related to consumption behavior randomness. With a higher relation between consumption and variables, e.g. illumination and occupancy, this means that the usage of the space is more efficient as the equipment is being used not at maximum intensity but according to the needs.

5 Conclusions and Future Work

This paper presented the research developments on modeling electricity consumption in three experimental spaces in a Portuguese university building. A previous work presented in [19] serves as a preliminary study on how to model electricity consumption in this building. Results considerable improvements from that preliminary approach, mainly due to the consideration of a variable Day type, which is a representation a priori of the expected occupancy of the experimental spaces, mainly taking into consideration the operating schedule of the room and the season (classes, exams or weekends and holidays).

Energy modeling is a bursting research field and several references can be found that outline the performance of different models. However, we could not find a work that compares specifically linear regression, fuzzy and NN models under the same experimental setup. This paper undertook this challenge and we have identified that NN are the models to which better performance values are regarded, reaching VAFs of 69.1, 95.3 and 91.5 %, and R^2 of 20.7, 91.0 and 78.5 %, respectively for the amphitheater, library and the offices.

The models that were developed for the amphitheater still lack accuracy and this is explained by consumption behavior randomness. A deeper understanding has to be undertaken and, eventually, implement efficiency measures that encourage users to change their consumption patterns according to the studied variables.

We argue that the investment in modeling capabilities to decrease the modeling error may give a feasible return since the presented results can roughly correspond to an error decrease in kWh between linear regressions and NN models corresponding to 650 €/year solely for the three presented spaces. Having developed more accurate models, we can now study the impact of the implementation of energy efficiency actions that have already been undertaken in these spaces after this experimental procedure.

Acknowledgments This paper was written under the scope of the Project Smart Campus Building-User Interaction for Energy Efficiency, CI: CIP-ICT-PSP-2011-5, GA: 297251.

References

1. Kangji, L., Hongye, S., Jian, C.: Forecasting building energy consumption using neural networks and hybrid neuro-fuzzy system: a comparative study. Energy Build. **43**(10), 2893–2899 (2011). doi:10.1016/j.enbuild.2011.07.01
2. Buildings Performance Institute Europe: Europes buildings under the microscope. Available at: http://www.institutebe.com/InstituteBE/media/Library/Resources/ (2011). ISBN: 9789491143014
3. Klein, L., Kwak, J., Kavulya, G., Jazizadeh, F., Becerik-gerber, B., Varakantham, P., Tambe, M.: Coordinating occupant behavior for building energy and comfort management using multi-agent systems. Autom. Constr. **22**, 525–536 (2012). doi:10.1016/j.autcon.2011.11.012
4. European Commission: Europe 2020, the targets, Europe 2020 targets. Available at: http://ec.europa.eu/europe2020/targets/eu-targets/ (2014)
5. Woodal, G.: SMART 2011 / 0072: Methodology for Energy-Efficiency Measurements Applicable to ICT in Buildings (eeMeasure) D1.2 Non-Residential Methodology. Empirica Gesellschaft fr Kommunikations, Germany (2011)
6. Reddi, T.: Applied Data Analysis and Modeling for Engineers and Scientists. Springer. ISBN 978-1-4419-9612-1 (2011)
7. Jang, J-S.: Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence. Prentice Hall, Japan (1997). ISBN 0-13-261066-3
8. Sousa, J.M.C., Kaymak, U.: Fuzzy Decision Making in Modeling and Control. World Scientific, Singapore (1993). ISBN 981-02-4877-6
9. Rumelhart, D.E., Hinton, G. E., Williams, R. J.: Learning internal representations by error propagation. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, pp 318–362. MIT Press Cambridge, MA (1986)
10. Ardente, F., Beccali, M., Cellura, M., Mistretta, M.: Energy and environmental benefits in public buildings as a result of retrofit actions. Renew. Sustain. Energy Rev. **15**(1), 460–470 (2011). doi:10.1016/j.rser.2010.09.022

11. Hooi, D., Kubota, T.: Development of an adaptive thermal comfort equation for naturally ventilated buildings in hothumid climates using ASHRAE RP-884 database. Frontiers Archit. Res. **2**(3), 278–291 (2013). doi:10.1016/j.foar.2013.06.003
12. Babuska, R.: Fuzzy Modeling for Control. Library of Congress Cataloging-in Publication Data, Kluwer Academic Publishers, Delft, The Netherlands (1998)
13. Dounis, A., Caraiscos, C.: Advanced control systems engineering for energy and comfort management in a building environment—a review. Renew. Sustain. Energy Rev. **13**(67), 1246–1261 (2009). doi:10.1016/j.rser.2008.09.015
14. Zhibin, W., Jiuping, X.: Predicting and optimization of energy consumption using system dynamics-fuzzy multiple objective programming in world heritage areas. Energy **49**, 19–31 (2013). doi:10.1016/j.energy.2012.10.030
15. Moradi, M., Hajinazari, M., Jamasb, S., Paripour, M.: An energy management system (EMS) strategy for combined heat and power (CHP) systems based on a hybrid optimization method employing fuzzy programming. Energy **49**, 86–101 (2013). doi:10.1016/j.energy.2012.10.005
16. Zhao, H., Magoules, F.: A review on the prediction of building energy consumption. Renew. Sustain. Energy Rev. **16**, 3586–3592 (2012). doi:10.1016/j.rser.2012.02.049
17. Yang, R., Wang, L.: Multi-objective optimization for decision-making of energy and comfort management in building automation and control. Sustain. Cities Soc. **2**(1), 1–7 (2012). doi:10.1016/j.scs.2011.09.001
18. Kalogirou, S., Bojic, M.: Artificial neural networks for the prediction of the energy consumption of a passive solar building. Energy **25**(5), 479–491 (2000). doi:10.1016/S0360-5442(99)00086-9
19. Pombeiro, H., Silva, C.: Linear, Fuzzy and Neural Networks models for definition of baseline consumption: Early findings from two test beds in a University Campus in Portugal. Science and Information Conference, Thistle Hotel Conference Center, London, UK (2014). ISBN: 978-0-9893193-1-7

Delivering Faster Results Through Parallelisation and GPU Acceleration

Matthew Newall, Violeta Holmes, Colin Venters and Paul Lunn

Abstract The rate of scientific discovery depends on the speed at which accurate results and analysis can be obtained. The use of parallel co-processors such as Graphical Processing Units (GPUs) is becoming more and more important in meeting this demand as improvements in serial data processing speed become increasingly difficult to sustain. However, parallel data processing requires more complex programming compared to serial processing. Here we present our methods for parallelising two pieces of scientific software, leveraging multiple GPUs to achieve up to thirty times speed up.

Keywords GPU · CUDA · GPU cluster · Parallelisation

1 Introduction

Some of the strategic drivers for software development in computational science and engineering are outlined by EPSRC [1]. In particular, the focus "development of novel code, the development of new functionality for existing codes and the development and re-engineering of existing codes. Strategic drivers are: developing code for emerging hardware architectures; developing researchers with key software engineering skills and software sustainability" [2] is pertinent to code used in HPC. We consider this strategy one of the key drivers in the context of software sustainability [3], and an important challenge in the development of scientific and engineering software.

In our research we have focused on improving the efficiency and scalability of existing software. The examples here have been designed to address the challenges

M. Newall · V. Holmes (✉) · C. Venters
High Performance Computing Research Group, University of Huddersfield, Queensgate, Huddersfield HD1 3DH, UK
e-mail: v.holmes@hud.ac.uk

P. Lunn
Birmingham City University, Franchise Street, Birmingham B42 2SU, UK

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_19

in processing large radio telescope data (SETI), and optical inferometry data used in surface measurements. The existing codes were re-engineered to support different GPU architecture, and enable scaling to larger GPU systems. In doing this we are addressing some 'software for the future' issues, taking into account the new hardware trends in GPUs deployment for HPC software.

Using GPUs in addition to more traditional High Performance Computing Resources to perform complex tasks or process large volumes of data has become increasingly common in supercomputing centres over the recent years. This trend can be seen by looking at the Top500 (A ranking of the worlds top scoring supercomputing sites [4]) over the past few years.

3D graphics rendering typically executes a single instruction at a time for every pixel to be rendered, and calculations for a single pixel are independent from those for other pixels [5]. This has resulted in graphics processors becoming largely parallel devices with hundreds of stream cores on a single device, capable of performing an instruction on a constant stream of data at high speed. Driven by the lucrative video games industry, GPUs are not only outpacing CPUs in terms of the rate of technological improvement, but also have much lower cost and power demands per core [6]. Owing to their original intended use in graphics processing, a fundamentally data parallel problem, GPUs can provide a significant speed boost to tasks which exhibit high data parallelism. Many fields of scientific research use software that fits these criteria, and GPUs are seeing increased use in this area [7–9]. In response to this new GPU architectures have been designed specifically for general purpose processing, such as Nvidias TESLA series, shown in Fig. 1.

To explore the potential for speed up in scientific applications, two existing software cases have been examined for sections appropriate for parallelisation.

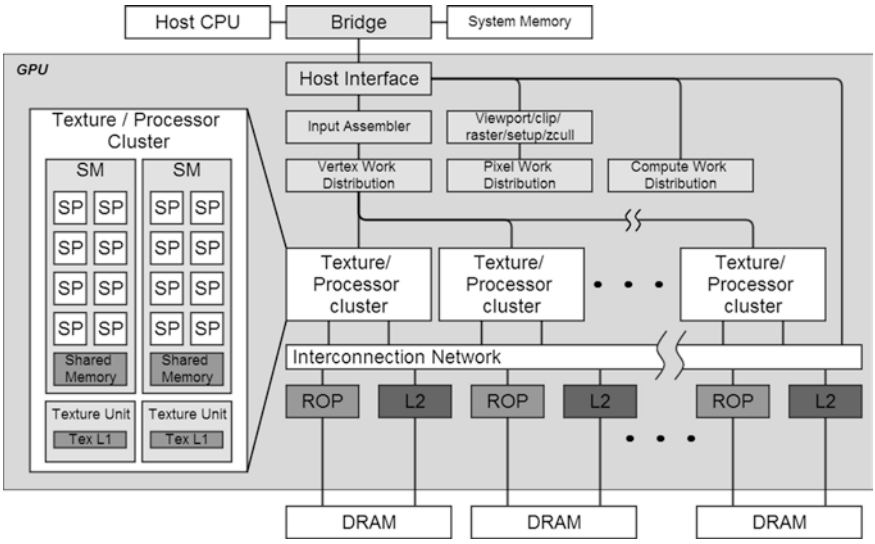


Fig. 1 Detail of the TESLA graphics and computing GPU architecture. Terminology: *SM* streaming multiprocessor; *SP* streaming processor; *Tex* texture, *ROP* raster operation processor [10]

These examples were rewritten to allow them to execute on a GPU cluster, the deployment of which is detailed in [11].

2 GPU Programming Models

In order to make general purpose processing on GPUs more accessible, there have been numerous models and libraries developed. Currently, the most mature of these are OpenCL and CUDA. Both models use the concept of kernels to contain parts of program structure which interact with compute devices, but differ in hardware support and scope.

OpenCL is an open source parallel programming standard, with notable contributors such as Apple, ARM, AMD, Samsung and Nvidia. It allows programs to take advantage of a very diverse array of processing devices such as GPUs, CPUs, DSPs, and FPGAs. The standard provides mechanisms for hardware vendors to add mechanisms for access to hardware specific features, which serves to increase its flexibility [12].

CUDA is developed by Nvidia for its own series of GeForce, Quadro and Tesla processors. It is flexible in its scalability and will run on an arbitrary number of processors without the need to recompile. This relieves the programmer of the burden of requiring specific knowledge of the hardware, which today can have vastly different clock speeds, RAM and numbers of cores depending on the model [13]. As CUDA functions are called from standard C or C++ it makes GPU programming much more accessible than has previously been possible. An example of the required effort to produce CUDA compatible code can be seen in listings 1 and 2. The CUDA programming model was used in our case study to accelerate processing of radio astronomy data produced by SETI, as well as increasing the throughput of wavelength scanning interferometry data analysis.

3 Accelerated Processing of Radio Telescope Data

The Search for Extra-terrestrial Intelligence (SETI) employs various methods in their attempt to discover evidence of technology based signals generated by civilizations outside of our own solar system. To this end vast amounts of radio telescope data must be analysed. The data is explored with signal processing techniques or image based techniques, such as SETILive, where images of this data are observed by the public who try to detect patterns in this data. Sonification is a process where data is transformed to sound [15]. SonicSETI is a project where radio astronomy data produced by SETI [16] is converted into sound (or sonified) so that the public can listen to this data to detect anomalous sounds.

However, processing this data is somewhat time consuming, taking almost 12 h to process an 8 GB set of data. The solution to this problem is to use GPU accelerated FFT libraries, such as the one provided by Nvidia [17].

The original software, written in JAVA, reads data from a file then determines how many FFTs to perform, before processing the data and saving to a new file. The time taken to process each data set was deemed unacceptable, at around 12 h per 8 GB dataset. The first effort towards acceleration was to replace the FFT function with calls to a CUDA accelerated FFT function, CUFFT. In the JAVA code this was done via JCUDA, a java wrapper for various cuda functions, demonstrating that GPU acceleration is accessible from a variety of languages.

Listing 1: A standard C function

```
void serial_function (int n, float a, float *x, float *y)
{
    for (int i = 0; i<n; i++)
        y[i] = a*x[i] + y[i];
}
//perform on 1M elements
serial_function(4096*256, 2.0, x, y);
```

Listing 2: The same function as might be written for execution on a CUDA supported GPU[14]

```
void gpu_function (int n, float a, float *x, float *y)
{
    int i = blockIdx.x*blockDim.x + threadIdx.x;
    if (i<n) y[i] = a*x[i] + y[i];
}
//perform on 1M elements
gpu_function<<4096, 256>>(n, 2.0, x, y);
```

To further increase acceleration, it was deemed necessary to rewrite the software in C++, in order to have more complete access to various CUDA functions. Shown in Listing 3 is a section the final C++ CUDA code which shows the host to device memory copy and using CUFFT to perform FFT on the device.

Listing 3: Using CUFFT to execute Forward FFT of Complex array 'fftData' on the GPU

```
//Allocate device memory
mem_size=sizeof(cuDoubleComplex) * N;
cuDoubleComplex *d_cufftData;
checkCudaErrors(cudaMalloc((void**)&d_cufftData, mem_size));

//Copy data to device
checkCudaErrors(cudaMemcpy(d_cufftData, cufftData, mem_size,
cudaMemcpyHostToDevice));

//Set FFT parameters and execute
cufftHandle plan;
checkCudaErrors(cufftPlan1d(&plan, N, CUFFT_Z2Z, 1));
printf("Starting FFT %d of %d \n", ffts, num_ffts);
checkCudaErrors(cufftExecZ2Z(plan, d_cufftData, d_cufftData,
CUFFT_FORWARD));

//Copy data back to host
checkCudaErrors(cudaMemcpy(cufftData, d_cufftData,
mem_size,cudaMemcpyDeviceToHost));

//Destroy CUFFT complex
checkCudaErrors(cufftDestroy(plan));
cudaDeviceReset();
```

3.1 Evaluation of Results

The graph in Fig. 2 compares the performance of the software, in Java, Java modified to use JCUDA, C++, and C++ with CUDA. Running regular FFT code compared to the GPU accelerated CUFFT library.

The program was rewritten using MPI, to allow it to take advantage of multiple GPUs. Figure 3 shows the run time of the FFT part of each C++ method; this is the part which has been implemented on the GPU so gives the best indication of acceleration. While restructuring the code to take advantage of both GPUs, the way in which data was copied to the GPU was changed to better utilise the memory on-board the device. Previously, enough data for a single FFT was copied to the device before being executed and copied back. In the MPI version, enough data is sent to fill the GPU memory before executing a batch of FFTs. This change reduced copy operations from 680 to 34.

An interesting finding was that Java performance was poorer than C even without GPU acceleration. It was determined that this was the result of slower disk access and the fact that JAVA uses big endian memory organization, so byte order has to be swapped before sending to GPU.

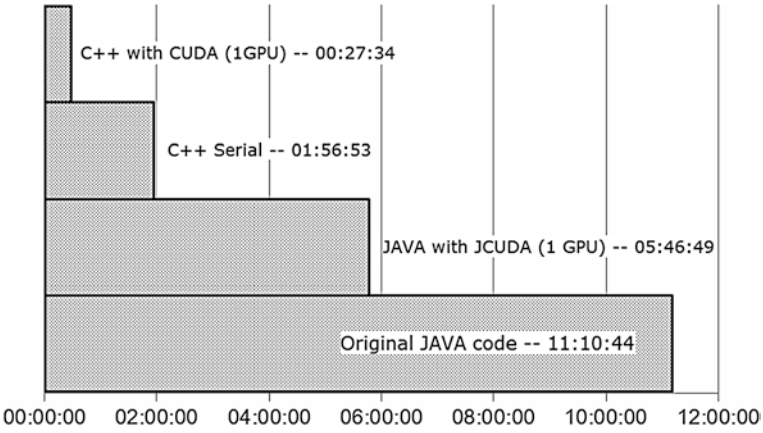


Fig. 2 Run time of each method

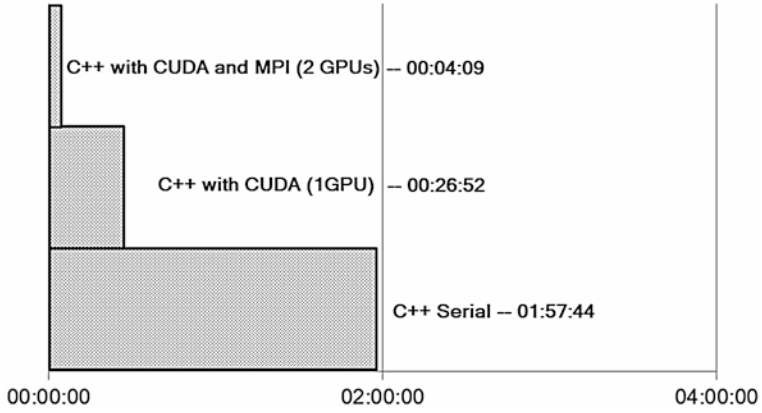


Fig. 3 Run time of the parallel/FFT part of each method

As this approach uses MPI, it would be relatively simple to scale this to any number of GPUs, the only mitigating factor being that network overhead would increase for every additional node, eventually making the addition of more nodes impractical.

4 Accelerated Surface Measurement with Environmental Noise Compensation

Optical interferometry is a widely used surface metrology technique. Wavelength scanning interferometry developments have been made that allow the process to be immune to environmental noise using phase compensation. However this compensation as well as data analysis processes limit performance, and hamper

efforts to inspect this data as the measurement takes place. The paper [18] details a method which uses CUDA to accelerate this process with a single GPU. Using a Multi-GPU system such as VEGA [11] this process can be accelerated further to allow a greater number of frames to be processed without a significant increase in process time.

The original CUDA program loads a set of bitmap frames, and the noise cancellation is calibrated by loading a matrix which has been processed by MATLAB. After calibration the data is processed using Nvidias CUFFT GPU accelerated parallel FFT algorithm, and all data is saved to disk. By using an MPI based method to submit to 2 GPUs, two sets of frames can be processed in parallel effectively doubling throughput, or alternatively one set can be divided in two to reduce processing time and increase the efficiency of in-process analysis. As with the sonification study, the program is split into a master process and a worker process—which must be able to run an arbitrary number of times, while the master co-ordinates. As there are 2 GPUs in our system we run 3 processes—one master and two workers. Figure 4 shows the main function of the program, Fig. 5 describes the MPI program which allows the CUDA program to executed on multiple GPUs.

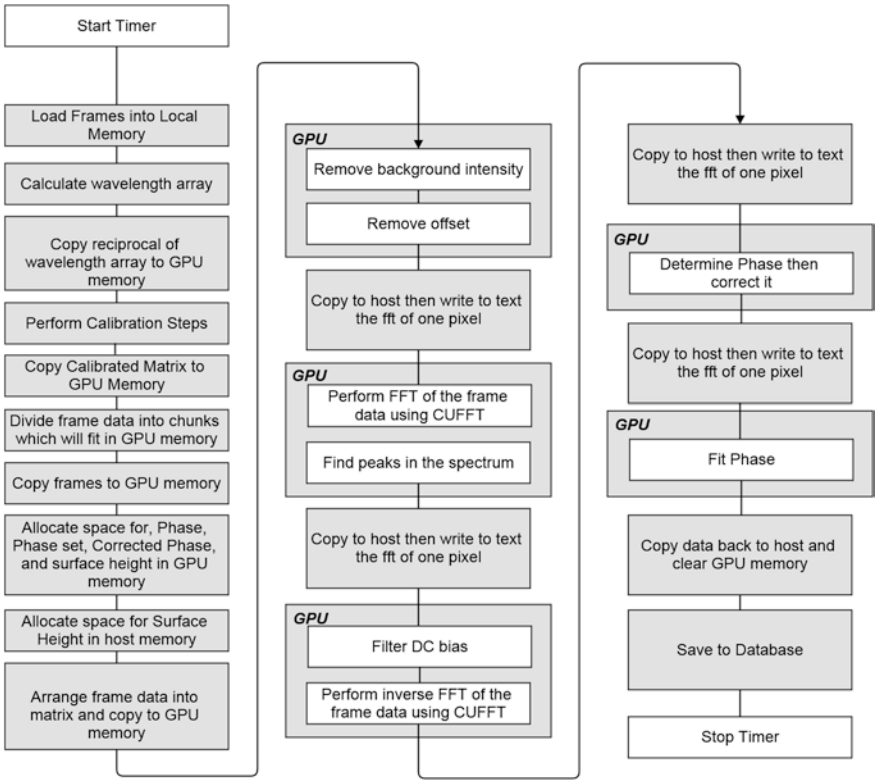


Fig. 4 Program flow for the original CUDA code

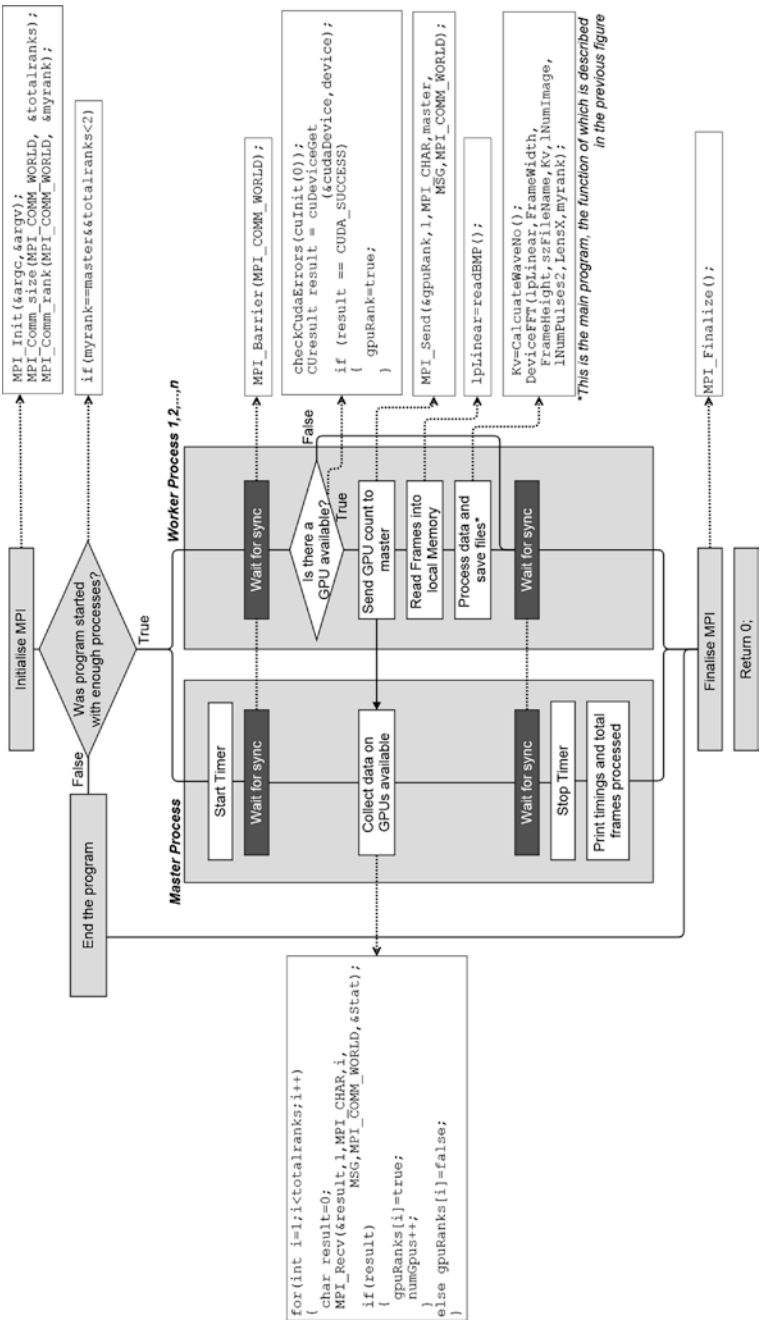


Fig. 5 Program flow for the MPI version using multiple GPUs

4.1 Evaluation of Results

The graph in Fig. 6 compares total runtime for a single GPU versus two. When running on one GPU 256 frames are processed, when running on 2 GPUs 512 frames are processed. It can be seen that running on 2 GPUs adds an overhead of approximately 400 ms, however Fig. 7 shows that running on 2 GPUs significantly reduces the per-frame processing time, being 1.9 times faster.

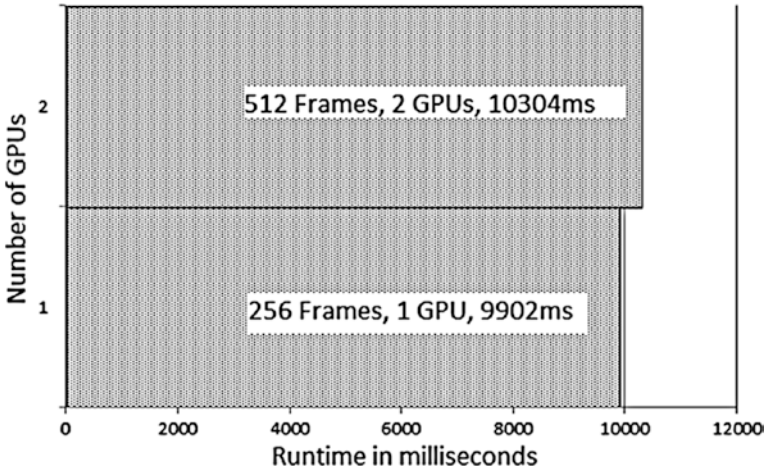


Fig. 6 Total run time

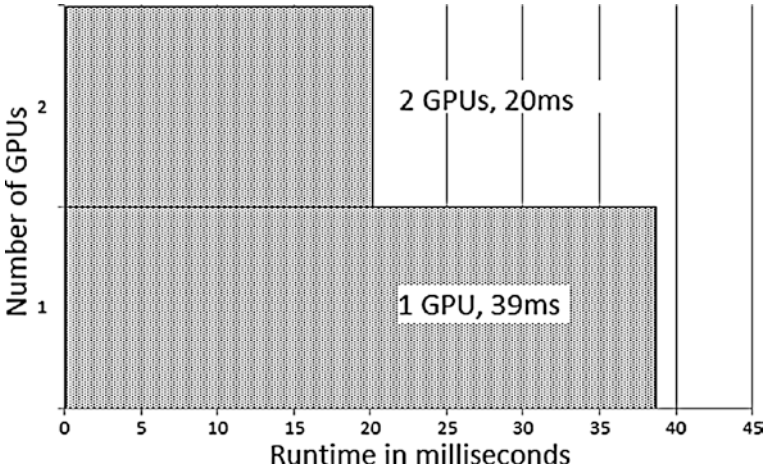


Fig. 7 Processing time per frame

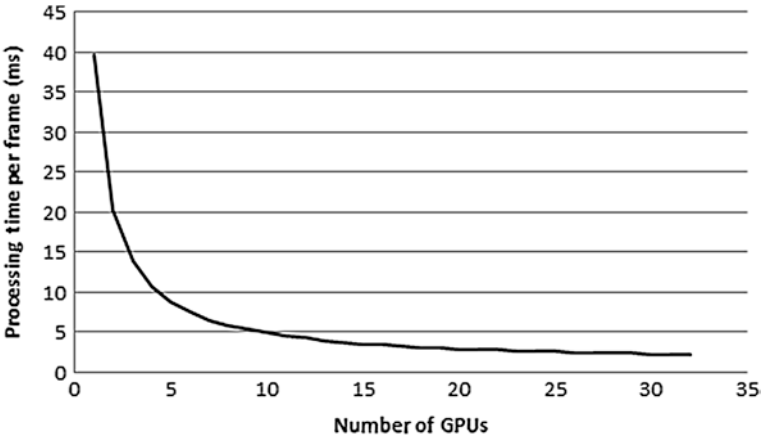


Fig. 8 Projected per-frame runtime on multiple GPUs

While only 2 GPUs were used in this case, our system has a capacity for 16. It can be speculated, given the results already gained, what the potential speed-up would be if 16 GPUs were used. Given that a single GPU processes 256 frames in 9,902 ms, and the addition of a second GPU adds a 400 ms overhead, it is not unreasonable to suggest that 16 GPUs may be able to process 4,096 frames in around 14 s (when including inevitable network overhead)—an 11 fold increase in throughput over processing on a single GPU, and a 5 fold increase over 2 GPUs. As the software already utilises MPI, were the hardware available the software could run at this scale without modification. The law of diminishing returns will apply here however, as network overhead increases with the number of processes it would be come less beneficial to keep adding more GPUs. Using these assumptions we can predict system performance, as shown in Fig. 8, which illustrates that as we add more GPUs the relative benefit is less every time. This is where it is important to consider speed versus efficiency. Using the methods outlined in [19] we can identify that the efficiency of the software, based on these projections, peaks at 5 GPUs, after which the improvements tend towards zero. Hence, while speed up does continue to increase after this point, the resources required to do this might be best used for other tasks.

5 Conclusion and Further Work

In this chapter we have presented our work in parallelising existing codes for processing radio telescope and surface metrology data. Writing sustainable code for modern, multi-core, multiprocessor systems still presents a challenge. Existing programming environments for parallel and distributed platforms do not provide

software developers with the tools necessary to test programs for the newest most powerful hardware.

Using the examples detailed here, and by utilising our own GPU cluster, we have shown that speed-up of up to 30 times is possible even on a modest GPU system. This will enable scientists and researchers to process complex problems and large volumes of data in near real-time.

To further explore the challenges of parallelisation we will investigate how these software examples scale onto much larger systems by running them on EMERALD, the UKs largest GPU cluster at Rutherford Appleton Laboratory [20].

In order to address the energy efficiency of our code, and software sustainability with respect to energy efficiency, we will build on our current research project funded by the innovate UK (technology strategy board) in Energy-Efficient computing [21]. Our focus will be on energy efficient data structures and algorithms for GPU technology. The resulting software will be evaluated and will be optimised under energy efficiency constraints creating more efficient software for affordable and sustainable high performance computing.

References

1. EPSRC (2014)
2. EPSRC: Software for the future ii (2014)
3. Lau, L., Griffiths, M., Holmes, V., Ward, R., Jay, C., Dibsdale, C., Venters, C., Xu, J.: The blind men and the elephant: towards an empirical evaluation framework for software sustainability journal of open research software. J. Open Res. Softw. **2**, e8 (2014)
4. Top500: Titan - cray xk7, opteron 6274 16c 2.200ghz, cray gemini interconnect, nvidia k20x (2013)
5. Nickolls, J., Dally, W.J.: The GPU computing era. Micro IEEE **30**(2), 56–69 (2010)
6. McKenney, P.E.: Is parallel programming hard, and, if so, what can you do about it? (2011)
7. Tarditi, D., Puri, S., Oglesby, J.: Accelerator: using data parallelism to program GPUs for general-purpose uses. In: Proceedings of the 12th International Conference on Architectural, pp 325–335 (2006)
8. Harris, M.: Mapping computational concepts to GPUs. In: ACM SIGGRAPH 2005 Courses, SIGGRAPH '05, ACM, NY, USA (2005)
9. Takizawa, H., Kobayashi, H.: Hierarchical parallel processing of large scale data clustering on a pc cluster with GPU co-processing. J. Supercomput. **36**(3), 219–234 (2006)
10. Lindholm, E., Nickolls, J., Oberman, S., Montrym, J.: Nvidia tesla: a unified graphics and computing architecture. Micro IEEE **28**(2), 39–55 (2008)
11. Newall, M., Holmes, V., Venters, C., Lunn, P.: GPU cluster for accelerated processing and visualisation of scientific data (2014)
12. Nvidia: Opencl (2013)
13. Nvidia: Introduction to cuda (2008)
14. Nvidia: The cuda parallel computing platform (2013)
15. Bonebright, T., Cook, P., Flowers, J., Miner, N., Neuhoff, J., Bargar, R., Barrass, S., Berger, J., Evreinov, G., Tecumseh Fitch, W., et al.: Sonification report: status of the field and research agenda (1997)
16. Nvidia: The allen telescope array (2013)
17. Nvidia: Nvidia cuda zone (2013)

18. Muhamedsalih, H., Jiang, X., Gao, F.: Accelerated surface measurement using wavelength scanning interferometer with compensation of environmental noise. In: Procedia Engineering: 12th CIRP Conference on Computer Aided Tolerancing, Apr 2012
19. Eager, D.L., Zahorjan, J., Lozowska, E.D.: Speedup versus efficiency in parallel systems. IEEE Trans. Comput. **38**(3), 408–423 (1989)
20. Oxford University: Emerald: e-infrastructure south GPU supercomputer (2013)
21. Innovate UK (2014)

Probabilistic Roadmaps and Hierarchical Genetic Algorithms for Optimal Motion Planning

Abdelhalim Lakhdari and Nouara Achour

Abstract In this paper we present a motion planning algorithm that combines between Probabilistic Roadmaps (PRM) and Hierarchical Genetic Algorithms (HGA) in order to generate optimal motions for a non holonomic mobile robot. PRM are used to generate a set of paths that will be optimized by HGA, the obtained trajectory leads a non holonomic mobile robot from an initial to a final configuration while maintaining feasibility and no-collision with obstacles.

Keywords Optimal motion planning · Probabilistic roadmap · Hierarchical genetic algorithms · Non holonomic mobile robot · Potential field

1 Introduction

The association, motion planning and genetic algorithms, has for a number of years, aroused the curiosity of researchers. Genetic Algorithms proposed for the first time by [1] have been used for planning since several years, and in [2] GA's were used to calculate the motion of a planar arm and solving the inverse kinematics problem. There are also other contributions by several researchers [3, 4].

The common problem to all methods is how to choose the initial population. Most of these methods use a set of paths encoded in the chromosomes, the optimal path is calculated after several iterations. The necessary step in these algorithms is the determination of the fitness function (optimization criterion), Garai and Chaudhuri [5] proposed to use a function of performance determined by the linear combination of distance, the smoothing angle and the robot position from

A. Lakhdari (✉) · N. Achour
LRPE Laboratory, USTHB University, Algiers, Algeria
e-mail: alakhdari@usthb.dz

N. Achour
e-mail: nachour@usthb.dz

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_20

the obstacles. Some papers have focused on dynamic environments [6] and others have tried to investigate in the synergism of respectively fuzzy logic-GA's [7] and neural networks-GA's [8].

For a car like robot (CLR) moving in a given environment, the general problem of motion planning is to determine a motion allowing it to move between two given configurations while respecting a number of constraints that may be related either to the environment (obstacles) or to the robot itself [9]. One of the constraints of the CLR is non holonomy which forces the robot to follow a trajectory imposed by the angle of its steering wheels. Because of its simplicity of implementation, PRM planners succeeded in solving a wide class of problems of motion planning in complex environments, where exact methods have failed. In this work we use PRM to find a trajectory leading the robot between two given configurations. The obtained trajectory can be optimized by HGA which are effective tools for solving complex optimization problems.

In this paper, we propose a novel approach for calculating an optimal collision free trajectory for a car like robot while maintaining a good clearance. Populations of paths are generated using the PRM, and HGA are used to get optimized paths, while path clearance is achieved through a fitness artificial potential function type.

Section 2 describes briefly the PRM-based path planning Sect. 3 lays the mathematical of the non holonomic robot, Sect. 4, details our approach in using Hierarchical genetic algorithms to plan optimized paths. In the last section, we do an analysis of our approach and report a series of actual runs.

2 Path Planning Using PRM

Probabilistic Roadmaps shown in [10] is a path planning method that allow to find collision-free paths for robots moving among stationary obstacles (static workspaces).

Because of its simplicity of implementation, the method PRM succeeded in solving a wide class of problems of motion planning in complex environments where exact methods are useless.

In this work we use PRM to build a graph $G = (N, V)$ capturing the connectivity of the free configuration space CS_{free} [10], where: N is a set of free nodes randomly generated using a uniform sampling strategy. The arcs of G, gathered in the set V, are collision-free paths connecting pairs of nodes.

PRM proceeds usually in two phases: a construction phase and a research phase: construction phase and research phase.

2.1 Construction Phase

In this phase, a uniform sampling strategy is used to generate random nodes in the free space which are added to the graph G. Then a selecting strategy is used to select a set of neighboring nodes for each new generated node.

The first problem that we met in the construction of PRM is the choice of discretization's parameters, in particular the number of nodes N_{max} and sampling distance d_{min} which is the minimum distance between two nodes of the network. The selection of the network size (number of nodes N_{max}) is very important to cover the maximum of the free space.

To calculate the number N_{max}, we used a method based on calculating the ratio between the free surface and the total surface of the environment for a given d_{min}. The idea is to generate a number of random nodes; the ratio between the free surface and the total one tend to the ratio between the number of free nodes and the total number of generated nodes.

$$\frac{free\ surface}{total\ surface} \approx \frac{number\ of\ free\ nodes}{total\ number\ of\ nodes} \tag{1}$$

To demonstrate this relation, let's take an example of a workspace that contains two obstacles in a known surface (Fig. 1):

Table 1 shows the ratio between the free surface and the total surface calculated with different number of generated nodes, and also the error between this ratio and the real one which is equal to 0.764.

The number N_{max} will be the integer part of the ratio between the free surface and the surface of the disk that includes a square of side d_{min}.

$$N_{max} = \text{integer}\left(\frac{\text{free surface}}{d_{min}^2 \times \frac{\pi}{2}}\right) \tag{2}$$

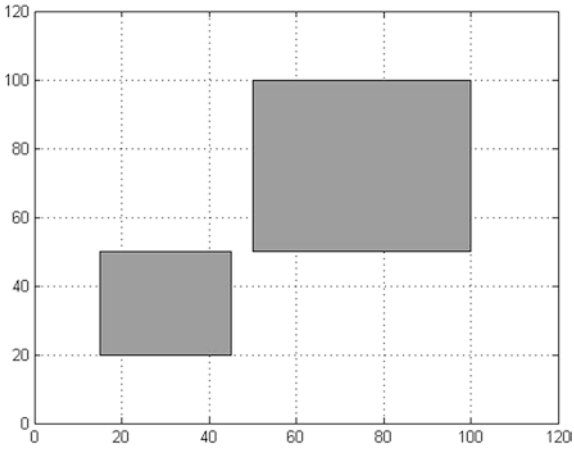


Fig. 1 Example of a known workspace

Table 1 Evolution of the ratio versus number of generated nodes

Number of generated nodes	10	100	500	5,000	10,000	50,000
Free nodes/total nodes	1	0.85	0.806	0.772	0.768	0.765
Error	0.236	0.086	0.042	0.008	0.004	0.001

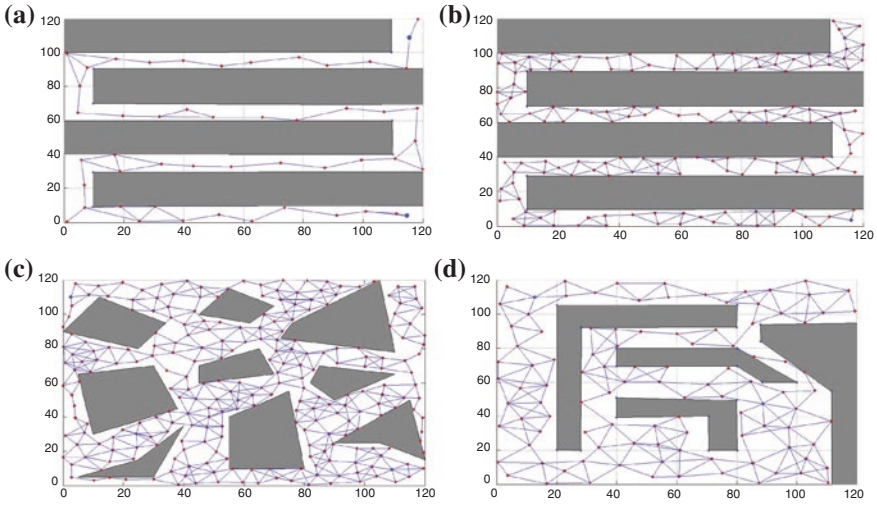


Fig. 2 PRM built in different environments with different d_{min}

To build the PRM, we used a uniform sampling method, where nodes are randomly generated, provided that the distance between each two nodes is greater than or equal to a minimum distance d_{min}. For a given node $q_{i,}$ it is considered neighbor of q_i any node situated at a distance lower or equal to d_{max} which is the maximum distance between two neighboring nodes distance.

The maximum distance d_{max} is chosen proportionally to d_{min} ($d_{max} = 2\ d_{min}$) in order to ensure a good connection to the network.

A collision detector allows to decide whether a node belongs or not to the free space. It is also used to test whether a given path is included or not in the free space.

The resulting network contains N_{max} nodes where each one is connected to its neighbors by segments (Fig. 2).

$$\boldsymbol{a} : d_{min} = 12 \,\&\, N_{max} = 50\,; \quad \boldsymbol{b} : d_{min} = 5 \,\&\, N_{max} = 145$$
$$\boldsymbol{c} : d_{min} = 5 \,\&\, N_{max} = 250; \quad \boldsymbol{d} : d_{min} = 8 \,\&\, N_{max} = 108.$$

2.2 Research Phase

In this phase, we will use the graph G built in the previous phase to find paths between a start node and a goal one. To find the shortest path connecting the start and the goal nodes, we used an algorithm composed of three steps:

The first step is to apply the A star algorithm [11] to find the shortest path in the network previously built. Because of the probabilistic nature of the PRM, the calculated path may contain irregular parts. Therefore a shortcutting of a process is carried out in the second step to remove the irregular parts of the calculated path.

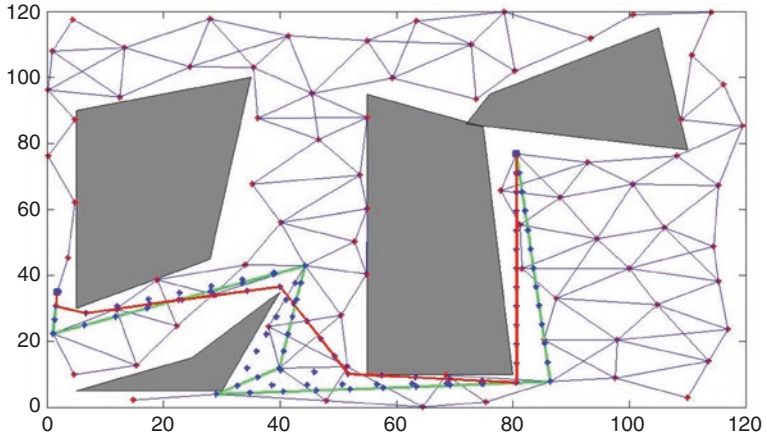


Fig. 3 Shortcutting and discretization applied on a found path using A star

The path obtained after the execution of the shortcutting process is not necessarily the shortest, to improve it we call in the third step a discretization process that will split the path into several parts, each one is of length greater than or equal to a predefined discretization step. The second and third steps are successively repeated enough times until the gain of shortcutting will be negligible (Fig. 3).

3 Nonholonomic Mobile Robot

The example most often of nonholonomic mobile robot is the car-like robot (CLR), this type of robots subject to a kinematic constraint called nonholonomic related to the rolling without slipping, this constraint forces the robot to follow a path imposed by the steering angle of its steered wheels.

In our case the robot is a car-like robot A modeled as a rectangle object moving in a Workspace $W = \Re 2$. The configuration of A is $q = (x, y, \theta)$ where x and y are the coordinates in the frame FW of the midpoint M of the rear wheels axle and $\theta \in [0, 2\pi]$ is the angle between the main axis of A and the abscissa axis of FW.

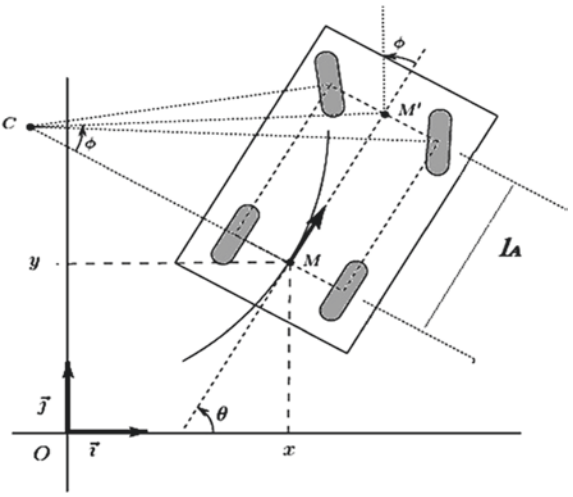
Assuming no slipping, the velocity of A is at each instant t pointed along the main axis. This is a nonholonomic constraint and it is modeled by the non-integrable following equation:

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0$$

The steering angle ϕ is constrained to take values in the interval $[-\phi_{max}, +\phi_{max}]$ with $\phi_{max} = \pi/4$ (Fig. 4).

The kinematic model of the CLR is usually expressed as follows:

Fig. 4 The car-like robot



$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \dfrac{v}{\mathrm{l}} \tan \phi \\ \dot{\phi} = \omega \end{cases} \tag{3}$$

where $\mathrm{v} = \mathrm{r} \cdot \dot{\theta}$ represents the linear velocity, r being the path P curvature, $\dot{\theta}$ the angular velocity of the reference point M of the robot and ω the steering velocity of the steering wheels.

3.1 Path Tracking

The previous path remains unfeasible because of nonholonomic constraints of the mobile robot; it must be adapted so that the robot can follow it. To make it feasible we use a path tracking function that allows extracting the different steering angles along the path previously found.

This function will calculate two points P'_1 and P'_2 of the steering (the beginning and end of the steering) for every three successive points of the path (P_1, P_2 and P_3) (Fig. 5).

Then the path that connects these three points will be split in two subpaths, the first is a straight line from the first point P_1 to the steering start point P'_1, and the second is the minimal radius arc ($\pm\phi_{\max}$) which connects the two steering points P'_1 and P'_2 as it is shown in Fig. 5.

P'_1 and P'_2 are calculated using the following relation:

$$d_1 = d_2 = \frac{r_{\min}}{\tan \theta}$$

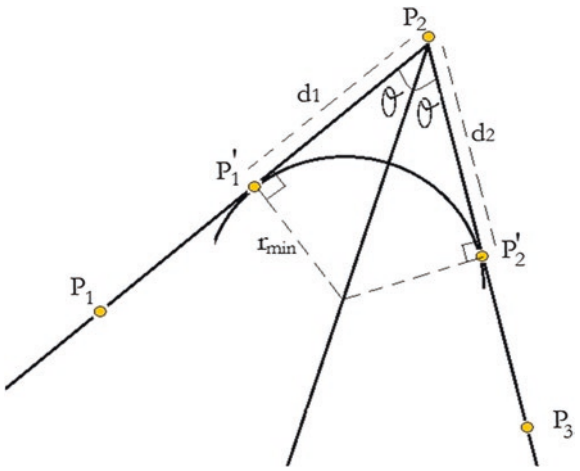


Fig. 5 The steering points P'_1 and P'_2

Repeating the process for every three successive points of the path allows finally to obtain a sequence of steering angles that allows the robot to follow correctly this path.

The robot configuration q_i is calculated using the discrete kinematic model of the CLR for a constant velocity v and a fixed steering angle ϕ.

$$\begin{aligned} x_i &= x_{i-1} + \Delta t \ \frac{v}{21} \cos\left(\theta_{i-1} + \tan\varphi \ \Delta t\right) \\ y_i &= y_{i-1} + \Delta t \ \frac{v}{21} \sin\left(\theta_{i-1} + \tan\varphi \ \Delta t\right) \\ \theta_i &= \theta_{i-1} + \Delta t \ \frac{v}{1} \tan\varphi \end{aligned} \tag{4}$$

4 Optimization Using HGA

Although the standard genetic algorithms (SGA) provide very good results in the solution of difficult optimization problems where other techniques have failed, they show some limitations when it comes to treat not only the problem of parameters optimization, but also the optimization of the structure of the chromosome.

During the execution of the trajectory, there may be a collision (Fig. 6), even if it is not the case, the calculated path is not necessarily the best, we must find the sequence of steering angles corresponding to the shortest path.

In the SGA, the structure of chromosomes must be predefined, but in some optimization problems (as in our case) it is difficult to do it, this is why we use HGA which will perform on two criteria; collision and path length to calculate the optimal feasible path.

HGA differ from SGA in the structure of chromosomes, they have a hierarchical structure that consists of two types of genes: parameter genes and control genes, parameter genes carry genetic information of the chromosome, while the control genes have a role in activation or deactivation of the parameter genes.

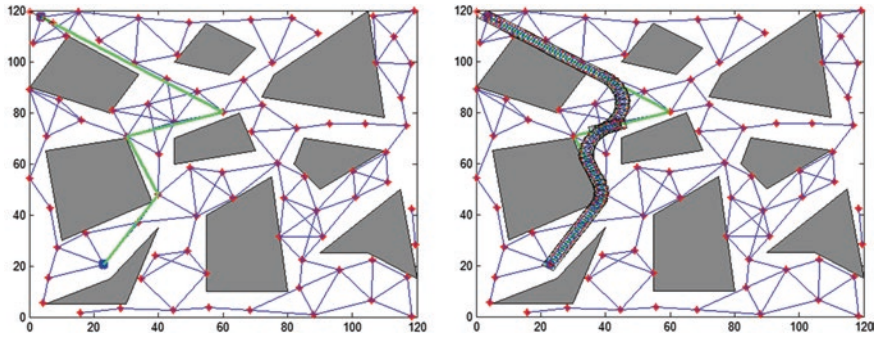


Fig. 6 Path tracking by the car-like robot

The activation of parameter genes is controlled by the value of the first level of the control gene, which in turn is controlled by the value of the second level of the control gene, and so on; the structure of a chromosome contains both active and inactive gene.

Because of inactive genes still exist in the chromosome and then can be transmitted to new generations, this hierarchical architecture implies that the chromosome contains more information than the conventional structure of SGA. Therefore, the artificial power is greatly improved [5].

4.1 Coding and Initialization

To generate the initial population, we will repeat ***N*** times the PRM execution to get ***N*** paths connecting start and goal positions; due to the probabilistic nature of PRM, every path will be different from the others.

To improve their quality, the process of discretization and shortening is applied repeatedly on each path, then we apply the path following function on each path for extracting the corresponding sequence of steering angles, all sequences will construct individuals of the initial population of genetic algorithms.

So each individual will have two chromosomes: the first contains genes carrying values of the steering angles φ_i extracted during each path following and the second contains the angle θ_{start} of the startup configuration.

The size of the first chromosome differs from one individual to another depending on the path in which it takes its genes. To overcome this problem, we have added to the first chromosome of each individual a number of genes (set to zero) until its size will be equal to the size of the largest chromosome, then we assign to each parametric gene a control gene g_i that will disable it if it is an added gene or enable it otherwise. The structure of an individual is shown in Fig. 7.

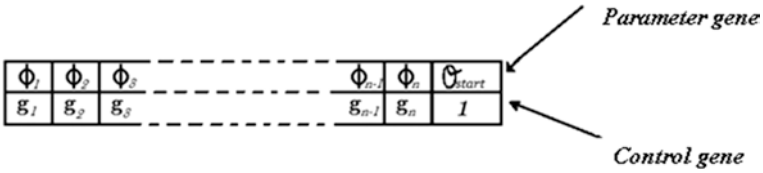


Fig. 7 An individual structure

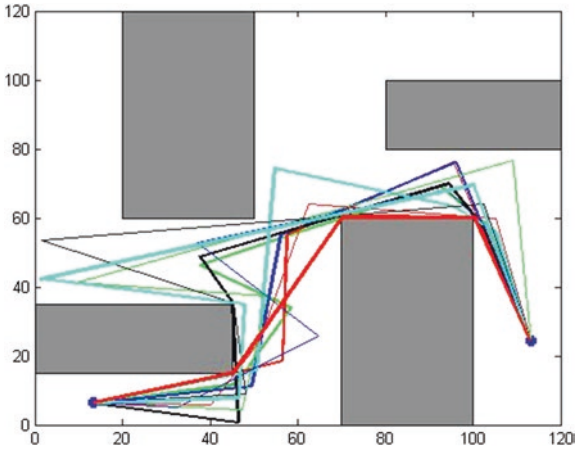


Fig. 8 Ten Paths obtained by PRM executed ten times and the reference path (in *bold red*)

4.2 Fitness Function

To evaluate individuals we use the visibility graph method [12] to calculate a reference path that connects start and goal configurations (Fig. 8), then we define a potential v_i for each configuration q_i that is a function of the distance between the reference point of the robot and the obstacles on one hand and the reference path on the other hand, where the reference path will exert an attractive field on the robot against the repulsive field exerted by obstacles [13], consequently the evaluation of each gene is proportional to the result of these two fields and the number of enabled parameter genes.

The potential v_i of a configuration q_i can be expressed by:

$$v_i = \begin{cases} \frac{1}{n}\left(\frac{K_a}{(d_a+1)^2} - \frac{K_r}{(dr_i)^2}\right) & if\ dr_i \leq d_0 \\ \frac{1}{n}\frac{K_a}{(d_a+1)^2} & if\ dr_i > d_0 \end{cases} \quad (5)$$

where K_a and K_r are positive constants, d_a the distance between the configuration q_i and the reference path, dr_i the distance between the configuration q_i and the nearest obstacle, d_0 is the influence distance of obstacles and n is the number of active genes parameters.

The evaluation of an individual is therefore the sum of the potentials v_i of its active parameters genes, and the fitness function can be expressed as follows:

$$f = \sum_{i=1}^{n} g_i \times v_i \tag{6}$$

4.3 Selection

The selection of individuals is performed using the elitist method where individuals are sorted in descending order according to their evaluation [1]. The best individuals are selected to participate in the next generation; they are combined using the operators of crossover and mutation to generate new individuals.

4.4 Crossover

We make a random draw with replacement of two selected individuals who will undergo to two types of crossover: the first is a barycentric crossover carried out on parameters genes with a weight taken randomly in the interval [0 1]. The second is a classical crossover (exchange of genes) performed on control genes (Fig. 9).

With:

$$\begin{cases} \phi'' = \alpha\,\phi' + (1-\alpha)\,\phi \\ \theta'' = \alpha\,\theta'_{start} + (1-\alpha)\,\theta_{start} \end{cases} \tag{7}$$

We repeat the crossover process until a sufficient number of individuals is obtained which will create the new population.

4.5 Mutation

In this work, we used a uniform mutation [14] which consists in adding a Gaussian noise with a very small variance ($\phi_{max}/25$) to the genes of the first chromosome

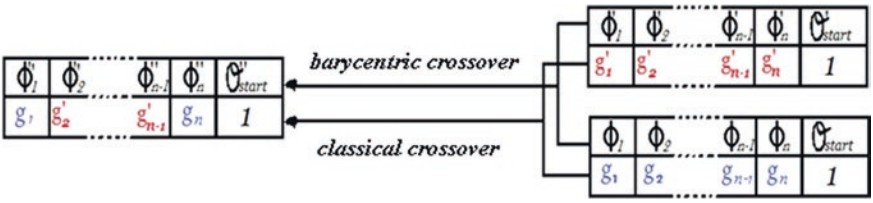


Fig. 9 Crossover

with a probability of about 20 %. For a steering angle ϕ, the mutation process can be expressed as follows:

$$\phi' = \phi + N(0, \frac{\phi \max}{25}) \tag{8}$$

The angle ϕ' obtained after mutation is accepted if it is included in $[-\phi max, +\phi max]$. Selection, crossover and mutation are repeated until we get a set of steering angles that allow the robot to follow the shortest feasible path.

5 General Algorithm

The algorithm presented in this paper can be resumed in Fig. 10.

6 Results and Discussion

Let us take the start and goal configurations in a given environment, the results are shown in Fig. 11.

Like all heuristic algorithms, our algorithm can take a large time to converge to the optimal solution, as it can quickly converge, it depends on several parameters, the first is the initial population (which depends of the built PRM) as well as the space discretization and the number M of times the shortcutting and discretization processes are applied (Fig. 2).

In cases where individuals of the initial population are similar, the search space of the solution is very small, the mutation operator plays a very important role to expand this area of research and give birth to individuals who may be more efficient.

The initial population may also contain a very close path to the optimum one, so the solution will converge quickly.

The choice of fitness parameters (d_0, K_a and K_r) plays a significant role in the convergence of the solution, in fact more they increase, more the solution approaches the reference path and moves away from the obstacles.

The evolution of the trajectories is given in Table 2.

In our case, we found that the algorithm has converged to the optimal solution after 300 generations, and the evolution of the solution is slightly sensitive from one generation to another, this is generally due to the initial population or obstacles distribution that may contain narrow passages. Thus we can conclude that it is difficult to define a stopping condition for the algorithm because the solution can stagnate until the mutation operator improves the solution and this can be after any number of generations, so we repeat the genetic operators enough time to have the best solution.

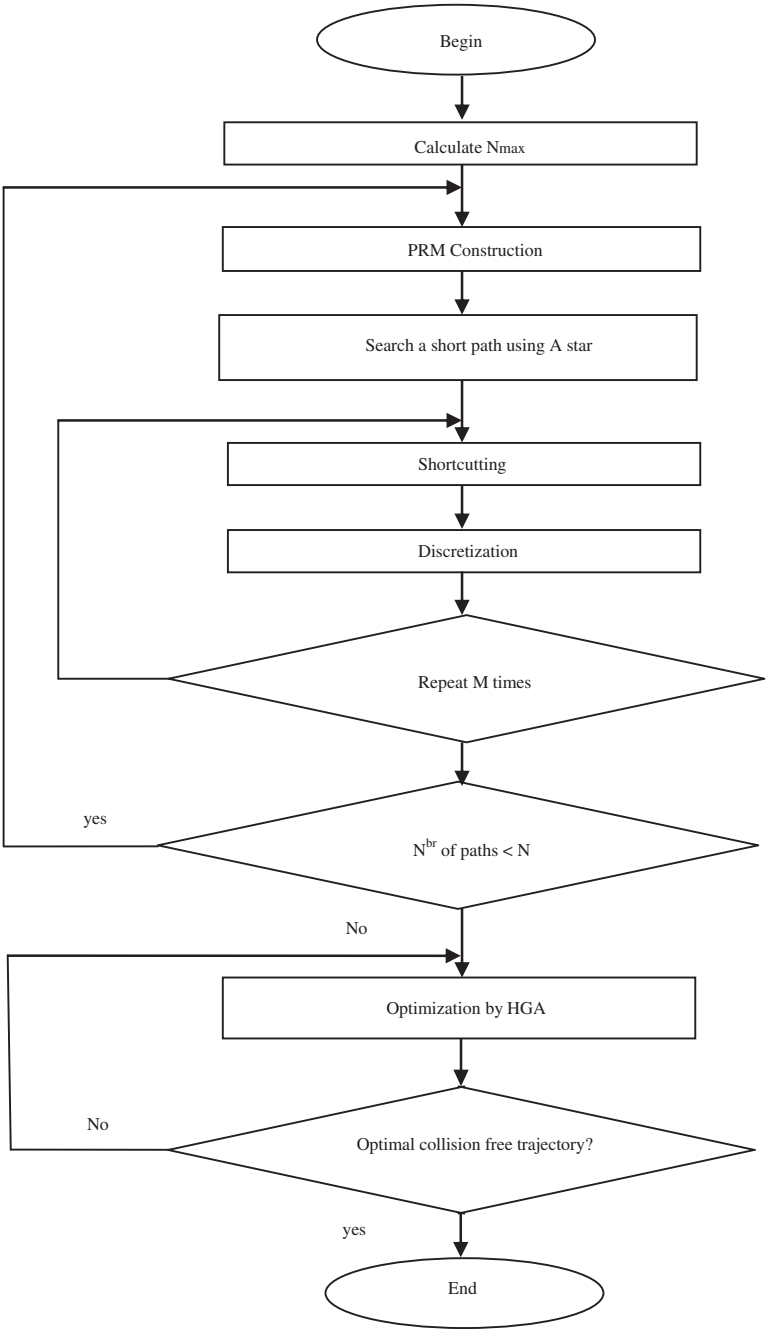


Fig. 10 The general algorithm of motion planning

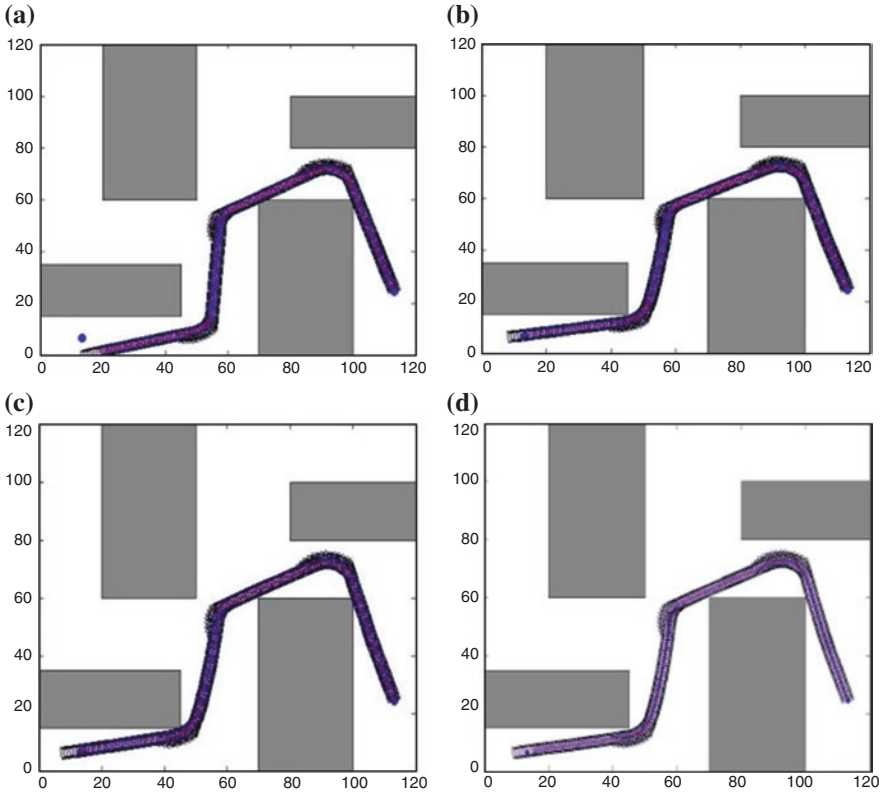


Fig. 11 Paths achieved (**a** before optimization, **b** after 100 generation, **c** after 200, **d** after 300 generation)

Table 2 Example of trajectory evolution

Generation	Initial	50	100	200	300
Fitness	−299.17	−533.19	−727.78	−886.01	−998.97
Trajectory length	185	178	176	175	176
Distance from goal	10.3	2.08	1.41	1.29	1.62
Collision	Yes	Yes	Yes	Yes	No

7 Conclusion

We have shown in this work that the problem of optimal motion scheduling for a car-like robot in a static environment can be solved in three steps: calculate free paths using the PRM method; In the second step we calculated the commands required to follow these paths, we finally used the HGA algorithms associated to the artificial potential field principle to generate the adequate controls for

optimized motion between a start and a goal configuration. The use of a fitness function based on the principle of potential field was a wise choice to ensure the balance between optimization, feasibility and safety of trajectories. The results showed the advantages of genetic algorithms in motion optimization.

References

1. Holland, J.: Outline for a logical theory of adaptive systems. J. Assoc. Comput. Mach. **3**, 297–314 (1962)
2. Ahuactzin, J.M., Talbi, E., Bessière, P., Mazer, E.: Using genetic algorithms for robot motion planning. In: Geometric Reasoning for Perception and Action. Lecture Notes in Computer Science, vol. 708, pp. 84–93 (1993)
3. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning, 1st edn. Addison-Wesley, Boston (1989)
4. Thomas, C.E., Pacheco, M.A.C., Vellasco, B.R.: Mobile robot path planning using genetic algorithms. In: Foundations and Tools for Neural Modeling, vol. 1606/1999, pp. 671–679. Springer, Berlin (1999)
5. Garai, G., Chaudhuri, B.B.: A distributed hierarchical genetic algorithm for efficient optimization and pattern matching. Pattern Recognit. **40**, 212–228 (2007)
6. Yang, S., Cheng, H., Wang, F.: Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing. IEEE Trans. Syst. Man Cybern. **40**(1), 52–63 (2010)
7. Fayad, C., Webb, P.: Development of a hybrid crisp-fuzzy logic algorithm optimised by genetic algorithms for path-planning of an autonomous mobile robot. J. Intell. Fuzzy Syst. 15–26 (2006)
8. Noguchi, N., Terao, H.: Path planning of an agricultural mobile robot by neural network and genetic algorithm. J. Comput. Electron. Agric. **18**(2–3), 187–204 (1997)
9. Latombe, J.C.: Robot Motion Planning. Kluwer, Norwell (1991)
10. Kavraki, L., Svestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic roadmaps for fast path planning in high dimensional configuration spaces. IEEE Trans. Robot. Autom. **12**(4), 566–580 (1996)
11. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. **4**(2), 100–107 (1968)
12. Laumond, J.P., Nissoux, C.: Visibility-based probabilistic roadmaps for motion planning. J. Adv. Robot. **14**(6), 477–494 (2000)
13. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. Int. J. Robot. Res. **5**(1), 90–98 (1986)
14. Boumaza, A.M., Louchet, J.: Dynamic files: using real-time Parisian evolution in robotics. In: Boers, E.J.W. et al. (ed.) Applications of Evolutionary Computing, vol. 2037 of LNCS, pp. 288–297. Springer, Berlin (2001)

Using Mouse and Keyboard Dynamics to Detect Cognitive Stress During Mental Arithmetic

Yee Mei Lim, Aladdin Ayesh and Martin Stacey

Abstract To build a personalized e-learning system that can deliver adaptive learning content based on student's cognitive effort and efficiency, it is important to develop a construct that can help measuring perceived mental state, such as stress and cognitive load. The construct must be able to be quantified, computerized and automated. Our research investigates how mouse and keyboard dynamics analyses could be used to detect cognitive stress, which is induced by high mental arithmetic demand with time pressure, without using intrusive and expensive equipment. The research findings suggest that when task demand increased, task error, task duration, passive attempt, stress perception and mouse idle duration may increase, while mouse speed, left mouse click and keystroke speed decreased. The significant effects of task demand and time pressure on mouse and keystroke behaviours suggest that stress evaluation from these input devices is potentially useful for designing an adaptive e-learning system.

Keywords Stress detection · Keyboard dynamics · Mouse dynamics · Mental arithmetic · Adaptive e-learning

Y.M. Lim (✉)
Faculty of Applied Sciences and Computing, Tunku Abdul Rahman University College, Kuala Lumpur, Malaysia
e-mail: ymlim@acd.tarc.edu.my

A. Ayesh · M. Stacey
Faculty of Technology, De Montfort University, Leicester, UK
e-mail: aayesh@dmu.ac.uk

M. Stacey
e-mail: mstacey@dmu.ac.uk

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_21

1 Introduction

The research is an investigation of the use of mouse and keyboard dynamics to measure cognitive stress without intrusive special equipment, when experimental subjects do mental arithmetic with different levels of difficulty and time pressure. It is motivated by the desire to develop e-learning systems that can adapt their behaviour and the content they deliver to the needs of their users. Learning is a complex process, which is not simply about acquiring knowledge, but involves mix of working memory organization, attention and cognitive control processes, which could be impacted by motivational and emotional factors. Negative emotion inducing environment and high-stakes situation that generates stress, fear of failure, anxiety or stereotype threat (such as female cannot do programming) could often cause students to perform at their worst. Beilock and Ramirez [1] study the relationship between emotion and cognitive control, and they found that high-pressure and negative emotion-inducing situations reduce student maths performance. On flip side, if they are placed under less emotion-inducing situations, the students are more readily available for executing a more challenging task. This is because negative emotion could inhibit appropriate cognitive resources that are necessary for optimal skill execution to be recruited by human mind. Other factors that affect cognitive load also include causal and assessment factors. Causal factors involve the characteristics of the subject such as skills or expertise possessed, task complexity, environment (such as noise) and their mutual relations. Assessment factors contain mental load, mental effort and performance [2]. To measure cognitive load, one or more assessment techniques can be utilized, which include subjective methods, physiological tests and task-performance based measurement [3]. Subjective methods, such as self-report survey, are done based on the assumption that humans are able to measure their thought (for instance, the amount of mental effort they expended or the level of stress they experienced). Although this method is simple, it is considered unreliable as human thinking is highly subjective and people can easily deny their thoughts. Therefore it is important to have an effective measure to quantify cognitive load. Physiological tests are able to detect changes in cognitive functioning that are reflected in measurable physiological measurements, such as heart rate or eye activity. However they cannot be easily implemented without special equipment (which is normally expensive), so not as part of normal system. Furthermore, physiological tests are invasive to the experimental subjects as the equipments are attached to their bodies, so they may not feel comfortable to carry out the task normally. Task-performance-based techniques measure actual performance of the given tasks. This technique is more reliable than the subjective method, as quantitative data such as success and failure rates of the task could be collected. However, solely relying on task-performance-based techniques may not be good enough as task performance could be affected by other factors such as attitude (e.g. lack of interest or seriousness in work), rather than weak cognitive function. It is better if some of these techniques can been combined to give a relative indication of the acceptable level of cognitive load.

To introduce a cost-effective, non-invasive and computational efficient method, automatic analysis of how users produce mouse and keyboard input during task execution is potentially useful. If mouse and keystroke behaviours are related to task performance and cognitive load, then they can be applied for designing adaptive instructional contents in an e-learning system. Furthermore, if the system can evaluate users' mental states or behaviours by measuring their emotions and stress levels, then the system is able to affect the attitude of the users towards learning and help them overcome learning obstacles [4]. This is because measures of cognitive overload, which leads to difficulty in coping with task demands (overstress in Selye's terminology [5], and underload, leading to boredom and lapses of attention (understress in Selye's terminology), are particularly important. Our research aims to analyse how keystroke and mouse behavioural patterns change according to the task demand, which is varied by mental arithmetic problem complexity and time pressure. We would like to observe how cognitive stress relates to task-performance (such as error rate, the duration spent on a task and the attempt of giving up a task), mouse and keystroke behaviours. If correlations between user's cognitive stress, task performance, mouse and keyboard dynamics can be found, then this information is potentially useful in designing an adaptive e-learning system.

2 Related Work

2.1 Mental Arithmetic and Cognitive Load

Mental arithmetic problems under time pressure are widely used to induce cognitive stress [6–8]. A study by Imbo and Vandierendonck [9] suggested that larger numbers and borrow operations in arithmetic problems, which involve longer sequences of steps and require maintenance of more intermediate products, will place greater demands on human working memory. Once the demand has exceeded the working memory capacity and temporal limitations, then the task is deemed too challenging to be continued [10]. Although much research has investigated how attention, memory and computational processes support arithmetic calculations, but less work has addressed how math performance can be influenced by emotional factors, such as stress. Beilock and Ramirez [1] suggested that stressful and emotion-inducing situations could lead to unwanted performance degradation even for relatively simple calculations in math performance, due to negative emotion could prevent or inhibit the recruitment of the appropriate cognitive resources necessary for optimal skill execution. However, Weinberg et al. [11] argued that human attention to emotion stimuli may not be automatic nor obligatory. When the context of the emotion stimuli is not relevant to the task (such as seeing a picture of a crying face), human may demonstrate little-to-no impact on the emotional modulated arithmetic task. In other words, the effects of the stimuli on cognitive process may depend on both of the attentional demands of the task

and the salience of the stimuli [12]. The impact of negative emotion on performance decrement may be caused by the task demands itself (such as high requirements), or other factors that are related to the task (such as time pressure).

2.2 Mouse and Keyboard Dynamics in Emotion Detection

Mouse and keyboard dynamics analyses shed light on non-invasive emotion detection research since they have generated promising results in biometrics or authentication work. These input devices are not only cheap in cost, but they also provide greater advantages for a solution that can be fully automated and computerized, as compared to physiological methods. Both mouse and keyboard dynamics have been shown to differ according to different emotion, but most previous work has considered them in isolation. Lim et al. [13] investigated the effects of Web menu design on users' emotion, search task performance and their mouse behaviours. Their results showed that the effects of menu design on users' search task performance and their mouse behaviours are statistically significant. Bad setting of menu design generally increases mouse idle duration and occurrences, and reduces mouse speed and mouse click. Tsoulouhas et al. [14] used mouse dynamics to test students' boredom. Their research demonstrates that mouse speeds, mouse inactivity occurrences, mouse inactivity durations and movement directions are significantly different between bored and non-bored users, which they recorded their best results with the intervals of 10 s (false acceptance rate at 2.7586 %). Pusara and Brodley [15] and Shen et al. [16, 17] analysed mouse dynamics by focusing on user behavioural modelling. Shen et al. [18] stated that the user's distinctive mouse operation patterns can be caused by changes to several factors, which include user's emotional states such as anger, despair, happiness, nervous, excitement, pressure and so on, and physical conditions such as tiredness and illness. Vizer [19] analysed keystroke dynamics and linguistic features by detecting changes in typing associated with physical stress and cognitive stress. Their research showed that keystroke features can be significantly changed by cognitive stress, which include keystroke pause length (key latency), time per keystroke (keystroke speed), deletion keys (backspace key and delete key), and use of navigation keys and other keys (such as letter and number keys). However, although using mouse and keyboard dynamics to detect emotion is proven effective, there is very little research done that unifies mouse and keyboard dynamics in emotion detection. The unification of both methods is important as there is a risk of collecting misleading information from only one channel. For instance, if we only analyse keystrokes, the results may be affected by long stops and irregular restarts [20], which could be due to the user's attention being diverted to another activity, or the user using a mouse rather than a keyboard to perform an action (such as drag-and drop or clicking a button to execute a command). Moreover, in a real application, users may use either mouse or keyboard, or a combination of both for specific tasks.

3 Research Questions and Design

We begin by hypothesizing that an automatic evaluation of cognitive stress can be obtained through acquisition and processing of three datasets, which are task performance ($B(T)$), mouse behaviour ($B(M)$) and keystroke behaviour ($B(K)$). Our research questions are as follows:

1. Do task demand and time pressure affect cognitive stress?
2. Do task demand and time pressure affect user's task performance, mouse behaviour and keystroke behaviour?
3. Are there correlations between task demand, cognitive stress, task performance, mouse and keystroke behaviours?

We would like to examine the potential significant effects of cognitive stress, which is induced by task demands with time pressure, on the changes of behavioural patterns in $B(T)$, $B(M)$ and $B(K)$. If the answers for the questions above are positive, then a rule-based adaptive e-learning system can be designed. Figure 1 shows our proposed system architecture using model-view-controller design. The models include modelling of keystroke behaviour, mouse behaviour and task performance (see Sect. 4.1). These behaviours are formed based on mouse and keystroke raw data such as mouse locations, time-stamps, keys pressed, etc., which are collected in every 10 ms. Then based on the needs of the system developer, the user behaviour can be analysed for an interval of designated time, t. Due to huge temporal variations of mouse and keyboard dynamics of a user, and also high behavioural differences between individuals, calibration of mouse and keyboard dynamics should be collected during login process, so that baseline condition (non-stressed) can be formed. Furthermore, these huge variations can be sensitive to generate significant difference even small departures from homogeneity and the assumption of normality, hence the collected data should be transformed using appropriate function (such as logarithm and square root).

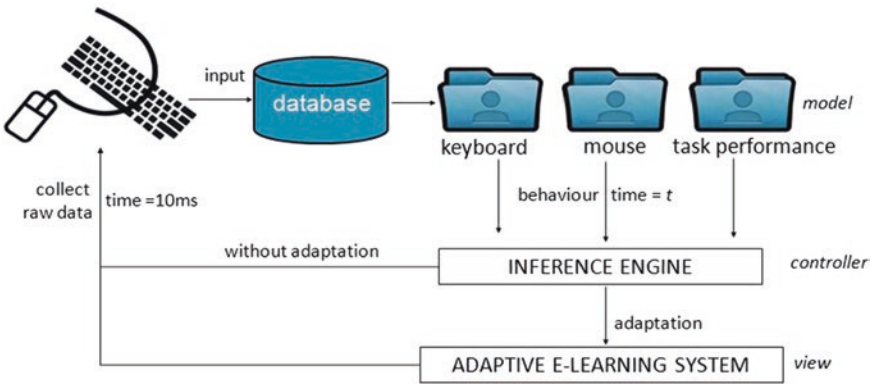


Fig. 1 Proposed system architecture

Privacy must be embedded into the design and architecture of the system, and we must be offering measures as strong privacy defaults, appropriate notice and empowering user-friendly option [21]. Therefore the users should be given an option for not to be observed by the adaptive system. We also need to ensure that at the end of the process, all data are securely destroyed, in a timely fashion, and no data that reveals individual identity would be kept. The actual data of the keys used, which reflect the original content of the text (such as username and password) must not be stored. These data must be encoded for the use of the analysis purpose only (for instance, all number keys or character keys are represented as 'k'). After the necessary transformation and formation of individual user behaviour, the system could then compare the subsequent behaviours with the baseline condition so that the behavioural patterns can be analysed. Once the rule that detects significant increment of stress level is fired, then the instructional content of the e-learning system can be adapted to motivate the learner to continue the task.

4 Methodology

To enable necessary data to be collected for the formation of Task Performance ($B(T)$), Mouse Behaviour ($B(M)$) and Keystroke Behaviour ($B(K)$), a program is written in Java to capture the features of $B(M)$, and another separate program is written in VB.NET to obtain the virtual-key codes by the Windows platform for $B(K)$. For every 10 ms, the mouse location is captured and its respective time in milliseconds is recorded. For every keystroke and mouse click, the key information and the time (in milliseconds) of the event is stored. To simulate an e-learning environment, an imitation of the online assessment system is built. Ten different mental arithmetic problems with diverse complexity are given to the students (see Table 1). Each question is displayed on different individual Web pages. The students must answer all questions by doing mental arithmetic, and must type the answer into a designated textbox on each page. No calculator or calculation on paper is allowed. To force the student to use the mouse, the "Enter" key is disabled, and he or she must click the "Submit" button in order to submit the answer. On each page of the question displayed, a "give up" button is given so that the students can choose to skip the question if they do not wish to continue. Before starting, an instruction page is displayed and their agreements to continue the experiments must be obtained. Once they click the start button, the start time (in milliseconds) will be recorded and the first arithmetic question is revealed. When the participant submitted the page, the end time (in milliseconds) is recorded and the data needed by $B(T)$, $B(M)$ and $B(K)$ will be computed automatically.

Each time after the students completed a question (or skipped the question), a self-report survey will be displayed as follows:

You felt stressed when answering the previous question

Table 1 Mental arithmetic questions. We assume that task demand increased from Question 1 to Question 10 according to the increment of number of digit per number and amount of numbers in the question

No.	Maximum digit per number	Amount of numbers	Arithmetic problem
1	1	2	6 + 2
2	1	2	9*4
3	1	3	6*5−1
4	1	3	(8 + 9)*2
5	2	3	7−8*10
6	2	4	58 + 20*(8−6)
7	2	4	67−2*(4 + 2)
8	3	5	(880 + 12 + 50−520)*2
9	3	5	105 + 83*5−3*60
10	3	5	561−81*5 + 3*610

This survey enables them to assess their stress perceptions when solving the arithmetic problem, following 7-point Likert scale (1 for strongly disagree, 7 for strongly agree). Therefore, this provides us the subjective measurement of the user's cognitive stress, *SP*.

All participants are required to run the experiments in a computer laboratory of a higher education institution in Malaysia. All the computers in the laboratory were equipped with Windows 7, 3.10 GHz CPU, 4 GB RAM, 17″ monitor with the resolution of 1,024 × 768 pixels, external standard QWERTY HID keyboard and external HID-compliant mouse. The website runs on Google Chrome by default. Before they started the assessment, instructions are displayed on the screen and they must provide their consensus in order to continue the experiments.

4.1 The Control Group and Experimental Group

Seventy-seven year-2 students from Bachelor Degree in Computer Science and Bachelor Degree in Information Technology from the Malaysian higher education institution participated the experiments. However, due to outliers and missing cases, only 60 of them provided valid samples. Among these 60 students, all of them were between 18 and 24 years old, and 90 % were male. These students were divided into 2 groups. For control group, the participants were required to answer the arithmetic questions without time constraint. For the experimental group, they were given 30 s time limit for each question. The page would be submitted automatically if they could not complete the task on time.

4.2 Formulation of Task Performance, Mouse Behaviour and Keystroke Behaviour

Task performance is a dataset that measures activities related to the tasks that a student has completed. Task performance, $B(T)$, is defined as follows:

$$B(T) = \langle TD, Err, PA \rangle \tag{1}$$

TD The duration to complete one task (milliseconds (ms))
Err Error of task ($Err = 0$ if no error; $Err = 1$ if the answer is wrong)
PA Passive attempt ($PA = 999$ if attempt to give up; $PA = 1$ if attempt to wait until the time is up

We define the mouse behaviour as a dataset that captures the mouse features for each task. The mouse behaviour, $B(M)$, is defined as follows:

$$B(M) = \langle MS, MID, MIO, MC \rangle \tag{2}$$

MS Average mouse speed (pixels per ms)
MID Total mouse inactivity duration (ms)
MIO[1] Total mouse inactivity occurrences
MC $\langle MCL, MCR$[2]$\rangle$, which is a dataset that consists of left click rate per ms (MCL) and right click rate per ms (MCR)

Lastly we define the keystroke behaviour, $B(K)$, as a dataset that captures the keystroke features for each task as below:

$$B(K) = \langle KL, KS, EK \rangle \tag{3}$$

KL Average key latency (ms)
KS Average typing speed per key (per second)
EK[3] $BSK + DK$, the total occurrences of error keys used (EK), which includes backspace (BSK) and delete (DK) keys.

5 Results

To observe the behaviourial patterns of $B(T)$, $B(M)$ and $B(K)$ according to the changes of cognitive stress, SP, we conduct some statistical tests to perform the analyses. First, we use Levene's test to ensure homogeneity between the 2 groups. However due to the fact that Levene's test can be sensitive to detect even

[1] MIO was removed later due to inhomogeneous data.

[2] MCR was removed later due to no data.

[3] EK is removed due to insufficient data for BSK and no data for DK.

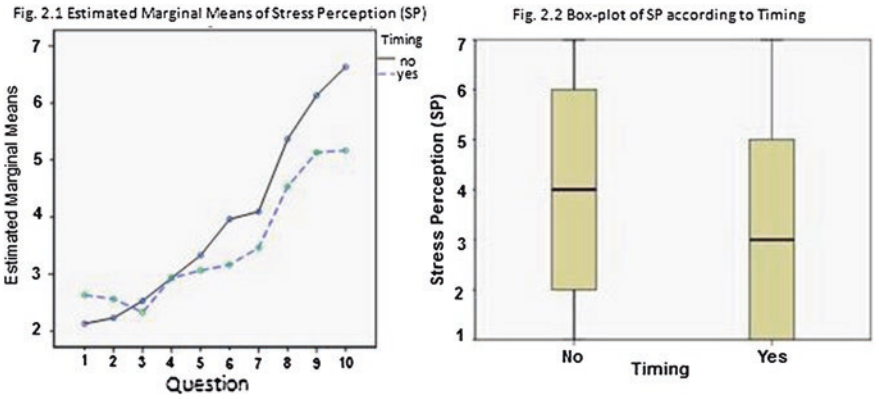


Fig. 2 *SP* increased according to Demand. The differences between questions are significant ($p < 0.5e^{-46}$). The Timing effect on SP is also significant ($p = 0.0025$)

small departures from homogeneity and the assumption of normality [22], *TD*, *MID* and *KS* are transformed using arctangent function and *MCL* is transformed using square root function. The following subsections discuss the results of the 3 research questions as shown in Sect. 3.

5.1 The Effects of Task Demand and Time Pressure on Cognitive Stress

After performing necessary data transformation, we first test the main effects of task demand (Demand) and time pressure (Timing) on *SP* by using Analysis of Variance (ANOVA) [23]. Both effects are significant (see Fig. 2). However, there is no interaction

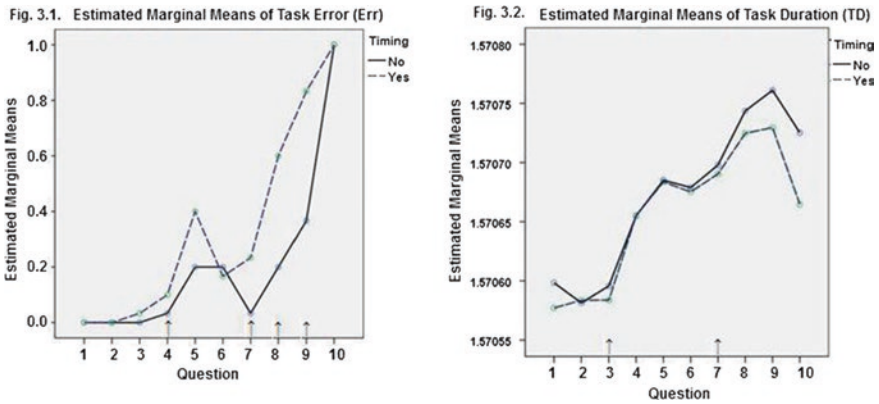


Fig. 3 Mean plots of task performance features

effect between Demand and Timing on *SP*. It is interesting to note that the participants in the control group (who are not given time pressure) in fact perceive higher stress than those in the experimental group. This could be due to some uncontrollable external environmental factors, such as tiredness after attending many classes on the same day, or intrinsic causal factors such as there are possibly more students having math anxiety in the control group than the experimental group. To examine the relationship between Demand, Timing and *SP*, Pearson correlation coefficient test shows that their correlation is significant. The significance of correlation between Demand and *SP* is given as $p = 0.02e^{-48}$, where $r = 0.56$; while the correlation significance between Timing and *SP* is given as $p = 0.0140$, where $r = -0.10$.

5.2 The Effects of Task Demand on Task Performance, Mouse Behaviour and Keystroke Behaviour

Demand significantly changes all the behaviours. Timing affects all features except *MID*. The interaction effect of Demand and Timing is only significant for *Err*, *MID*, *KS* and *KL* (see Table 2). We then performed Tukey Post Hoc Tests to analyse the variations between Demand and Timing effects to the three behaviours. The results are illustrated in Figs. 3, 4 and 5. The arrow markers in each graph indicate the significant difference between classes.

To observe the effects of Demand and Timing on *B*(*T*), we first observe the number of users who attempted to give up or could not finish the questions on time (*PA*). Table 3 shows that *PA* started to increase from Question 6 onwards. However at Question 10, *PA* dropped instead of continue to rise although the users scored even higher *SP* at Question 10. This phenomenon shows that there is an anomalous behaviour of *PA* at Question 10. We then observe the number of students who made errors in answering the questions. Figure 3 shows that the students who are not given time pressure are generally making less errors than those who are given 30 s limit. The number of students who made error started to increase from Question

Table 2 MANOVA tests of the between-subjects effects

Demand			Timing			Demand*Timing		
	Feature	*p*-value		Feature	*p*-value		Feature	*p*-value
B(T)	Err	*0.0000*	B(T)	Err	*0.0000*	B(T)	Err	*0.0000*
	TD	*0.0000*		TD	*0.0041*		TD	0.2472
	SP	*0.0000*		SP	*0.0025*		SP	0.0565
B(M)	MS	*0.0000*	B(M)	MS	*0.0133*	B(M)	MS	0.9371
	MID	*0.0000*		MID	0.3201		MID	*0.0033*
	MCL	*0.0000*		MCL	*0.0000*		MCL	0.0934
B(K)	KS	*0.0000*	B(K)	KS	*0.0036*	B(K)	KS	*0.0010*
	KL	*0.0000*		KL	*0.0031*		KL	*0.0000*

Italicized cell indicates that the difference is significant at the level of $p < 0.05$

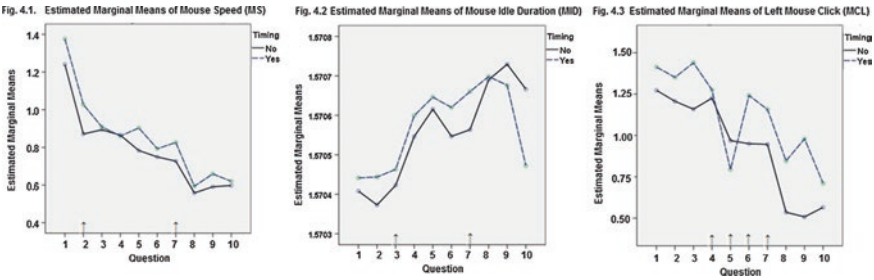


Fig. 4 Mean plots of mouse behaviour features. Generally the means of the experimental group are always higher than the control group, except *MID* (no significant difference)

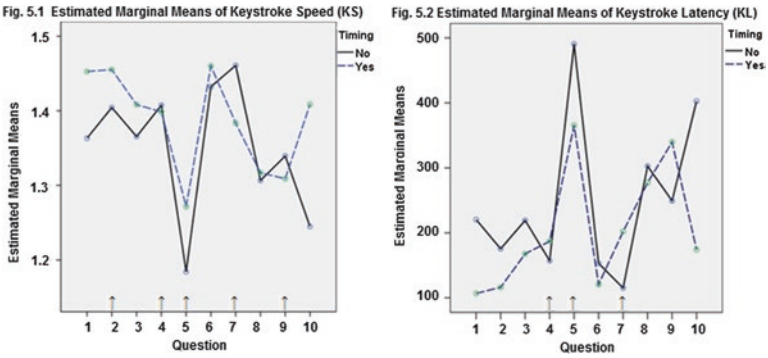


Fig. 5 Mean plot of keystroke behaviour feature. Generally the means of the experimental group are always higher than the control group

Table 3 Number of students who give up or could not finish task on time (*PA*)

Q.	Timing			Total	Q.	Timing			Total
	No	Yes				No	Yes		
	Give up	Auto submit	Give up			Give up	Auto submit	Give up	
1	0	0	0	0	6	1	0	1	2
2	0	0	0	0	7	0	1	0	1
3	0	0	0	0	8	2	6	1	9
4	0	0	0	0	9	2	14	1	17
5	0	0	0	0	10	5	5	1	11

3 onwards. Generally there are no significant differences among Question 1 to Question 4 in terms of *Err*. However, after Question 4, *Err* significantly increased at Question 5. More students made more mistakes for Question 5, 8 and 9 and all of them answered Question 10 wrongly. The fact that Question 5 achieves higher *Err* than Question 6 shows that Question 5 could be more difficult than Question 6, although they perceive Question 5 is less stressful than Question 6. In terms of

Timing, the users who were given time constraint made more mistakes than those without time pressure. For *TD*, there is no significant difference among Question 1 to Question 3, but *TD* increased significantly at Question 4. Then *TD* dropped slightly at Question 6, and gradually increased again until Question 7. The decrement of *TD* at Question 6 is consistent with the decrement of *Err* at the same question. This indicates that the users spent slightly longer time for Question 5 than Question 6. Again this shows that Question 5 could be more difficult than Question 6. Then there is a significant increment of *TD* at Question 8, but it started to drop at Question 10 instead of continues to rise. This is related to the anomaly at Question 10 as we observed in the behaviour of *PA*. In terms of Timing, those users who are given time constraint completed the task in shorter duration.

To analyze the changes of *B*(*M*) according to Demand and Timing, first we examine the effects on *MS*. Figure 4 shows that the highest *MS* falls on Question 1, which indicates that less stressful task would introduce higher mouse speed. There is a gradual decrement of *MS* from Question 3 to Question 7, followed by a significant decrement at Question 8, signify increment of stress perception would lead to lower mouse speed. However after Question 8, there is a slight increase of *MS* from Question 9 onwards. This phenomenon again indicates that the users have again behaved anomalously from Question 9 onwards. With reference to Fig. 4, the changes of MID is similar to *TD*, which it gradually increased according to the level of Demand, and it consists of 2 changeover points at Question 3 and 7. It also demonstrates anomalous pattern at Question 9 and 10 (which it started to decrease instead of rise), and a slight increment at Question 5. Figure 4 also shows that generally *MCL* decreased when stress level increased. However, it is notable that there is a drastic decrement of *MCL* at Question 5 and Question 8, but the pattern resumed to its normal behaviour in the subsequent questions. This also indicates that something has changed the pattern of *MCL* significantly at these 2 points. Similar to *MCL*, *KS* and *KL* of *B*(*K*) also demonstrate drastic changes at Question 5 and Question 8 as shown in Fig. 5 before they resumed to normal behaviour. Finally, all *MCL*, *KL* and *KS* also demonstrate incongruities between 2 groups of students at Question 10. In terms of Timing, those students in the experimental group demonstrate faster *KS* and lower *KL*, which suggest to us that if the students perceive lower stress, they would demonstrate higher *KS* but lower *KL* in general.

To explain the anomalies occurred, we review the complexity of the questions as shown in Table 1. Question 5 and Question 8 are the starting point of the increment of the digit per number in the arithmetic problems, which require more working memory to be recruited to store the information for further processing. Therefore we could predict that a change of question style, such as bigger numbers used in mental arithmetic, would lead to a temporal anomalous *MCL* pattern, i.e. a significant drop of *MCL* and *KS*, and increase of *KL* at one point. The information obtained from *MCL*, *KL* and *KS* could provide a more accurate measurement of cognitive load than subjective method, to inform the possibility that the question is more challenging than expected. To explain the anomalies happened at Question 10, which achieves the highest *Err* rate and *SP*, we strongly believe that

Table 4 Univariate tests for the effects of demand and timing on *SP*, *B*(*T*), *B*(*M*) and *B*(*K*)

Factor	Significance, *p*-value	Wilk's Lambda, λ
Demand	0.0000	0.1696
Timing	0.0000	0.8561
Demand*Timing	0.0000	0.7654

All factorial effects are significant at the level of $p < 0.5e^{-6}$

the students have reached an ultimate stress point at Question 9 (although it could be also affected by external factors such as fatigue and tiredness), which exceeds their endurance limit and makes them losing motivation to continue the task. As such, besides predicting *SP*, we are able to predict that the students may have experienced the need to cope with the change of question style at a specific point, and the point where they have lost motivation, by observing *B*(*T*), *B*(*M*) and *B*(*K*).

Lastly, we performed Multivariate Analysis of Variance (MANOVA) [22] to verify the effects of Demand and Timing on *SP*, *B*(*T*), *B*(*M*) and *B*(*K*). Table 4 shows the degree of confidence of the two factors' effects. Although both Demand and Timing give significant impacts to *B*(*T*), *B*(*M*) and *B*(*K*), however the Wilk's Lambda values for Timing and its interaction with Demand are high. This shows that the between-groups dispersions of Timing and its interaction with Demand are small. In other words, Demand is the main effect that affects stress perception and all three behaviours, but the impact of Timing is small.

5.3 Correlations Between Task Demand, Cognitive Stress, Task Performance, Mouse Behaviour and Keystroke Behaviour

To examine the relationships between Demand, *SP*, *B*(*T*), *B*(*M*) and *B*(*K*), we conducted Pearson Correlation Coefficient tests. The results are shown in Table 5. The highlighted cells indicate negative correlation coefficient. When Demand

Table 5 Correlation between features

	Demand	SP	TD	Err	MS	MID	MCL	KS	KL
Demand		✓	✓	✓	✓	✓	✓	✓	✓
SP	0.0000		✓	✓	✓	✓	✓	✓	✓
TD	0.0000	0.0000		✓	✓	✓	✓	✓	✓
Err	0.0000	0.0000	0.0000		✓	✓	✓	✓	✓
MS	0.0000	0.0000	0.0000	0.0000			✓	✓	✓
MID	0.0000	0.0000	0.0000	0.0000			✓	✓	✓
MCL	0.0000	0.0000	0.0000	0.0000	0.0000	0.0023		✓	✓
KS	0.0000	0.0002	0.0000	0.0000		0.0000	0.0000		✓
KL	0.0000	0.0000	0.0000	0.0000	0.0314	0.0000	0.0000	0.0000	

Significant correlation exists between two features at $p < 0.05$ (2-tailed) level if it is ticked (✓). Highlighted cell indicates negative correlation coefficient.

increased, *TD*, *SP*, *MID* and *KL* also increased, but *MS*, *MCL* and *KS* decreased. Therefore we could predict that if *TD*, *Err*, *MID* and *KL* increased but *MS*, *MCL* and *KS* decreased, then *SP* should increase.

6 Discussions

From the statistical analyses, task demand is the main factor that influences student's stress perception, task performance, mouse and keystroke behaviours. Although time pressure effect is also significant, however its impacts on the changes of stress perception and behaviours are relatively small. There is also an interaction effect between task demand and time pressure, but the combined effect of these 2 factors is also small. Correlation tests results suggest that prediction of cognitive stress increment is possible. When task demand increased, students' stress perceptions, duration spent to complete a task, error rate, passive attempt, mouse idle duration and key latency increased; while on flip side, mouse speed, mouse click rate and keystroke speed decreased. When task difficulty increased, but task performance, mouse and keystroke behaviours do not behave in a way that is expected, then anomaly can be detected. Anomalous behaviours indicate three possibilities: (i) there is either a wrong assumption about the independent factors (e.g. Question 5 appeared to be more challenging than Question 6 although it consists of fewer terms and operators); (ii) qualitative difference in task demands (e.g. the number of digits per number in the task increased would require more working memory to process the task), which can be observed through *MCL*, *KS* and *KL*; or (iii) the user is either understress or overstress, which is beyond their motivation limits (e.g. Question 10 contributes highest error rate and stress perception). After this ultimate stress point, prediction of *SP* using the production rules could probability become invalid, as the users have lost motivation to continue the task. Therefore it is important to activate the adaptive content to motivate the students to continue the task.

7 Conclusion

Our research shows that an automated evaluation of cognitive stress can be obtained through acquisition and processing of task performance, mouse behaviour and keystroke behaviour. When typing task demand increased, task error, task duration, passive attempt, stress perception and mouse idle duration may increase, while mouse speed, left mouse click rate and keystroke speed decreased. This is consistent with the findings by Lim et al. [13], which they found that when the users feel uncomfortable with the bad setting of menu design, generally their mouse idle duration and mouse idle occurrences would increase, but mouse speed and left mouse click rate would drop, although there is no significant correlation

between users' stress perception and search task performance. This research findings also show that task demand is the main factor that affects all three behaviours. The correlations between mouse behaviour and keystroke behaviour suggest that unifying mouse and keyboard dynamics analyses could be more useful than utilizing them separately. Anomalies of mouse and keystroke behaviours, such as mouse click, key latency, and keystroke speed, could be also be observed where there is a change of question style or the students may have lost motivation. Adaptive system can then be activated to motivate the students to continue the task. However our research has a few limitations. The results such as time pressure effects may be influenced by external environmental factors, such as external time pressure, and the attitude and motivation of the participants. It is also difficult to ensure all students are having comparable mental arithmetic skills. Therefore homogeneity among all students is not guaranteed. We also excluded mouse movement direction in the analyses, which can be an important variable that can be affected by cognitive stress. Besides, the sample size is small, and we may not be able to generalize our findings to represent the actual population. More rigorous experiments need to be conducted to validate the stress evaluation model.

References

1. Beilock, S.L., Ramirez, G.: On the interplay of emotion and cognitive control: implications for enhancing academic achievement. Psychol. Learn. Motiv. Adv. Res. Theory **55**, 137 (2011)
2. Paas, F.G.W.C., Van Merriënboer, J.J.G.: Instructional control of cognitive load in the training of complex cognitive tasks. Educ. Psychol. Rev. **6**, 351–371 (1994)
3. Kirschner, P.A.: Cognitive load theory: implications of cognitive load theory on the design of learning. Learn. Instr. **12**, 1–10 (2002)
4. Chatzara, K., Karagiannidis, C., Stamatis, D.: Students attitude and learning effectiveness of emotional agents. 2010 IEEE 10th international conference on advanced learning technologies (ICALT), pp. 558–559 (2010)
5. Selye, H.: The Stress in Life. McGraw-Hill, New Jersey (1956)
6. Setz, C., Arnrich, B., Schumm, J., La Marca, R., Troster, G., Ehlert, U.: Discriminating stress from cognitive load using a wearable EDA device. IEEE Trans. Inf. Technol. Biomed. **14**, 410–417 (2010)
7. Owen, A.M., McMillan, K.M., Laird, A.R., Bullmore, E.: N-back working memory paradigm: a meta-analysis of normative functional neuroimaging studies. Hum. Brain Mapp. **25**, 46–59 (2005)
8. Sloan, R.P., Korten, J.B., Myers, M.M.: Components of heart rate reactivity during mental arithmetic with and without speaking. Physiol. Behav. **50**, 1039–1045 (1991)
9. Imbo, I., Vandierendonck, A.: Do multiplication and division strategies rely on executive and phonological working memory resources? Mem. Cognit. **35**, 1759–1771 (2007)
10. Sweller, J., Ayres, P., Kalyuga, S.: Cognitive Load Theory. Springer, Berlin (2011)
11. Weinberg, A., Ferri, J., Hajcak, G.: Interactions between attention and emotion. Handb. Cogn. Emot. 35 (2013)
12. Hajcak, G., Dunning, J.P., Foti, D.: Neural response to emotional pictures is unaffected by concurrent task difficulty: an event-related potential study. Behav. Neurosci. **121**, 1156 (2007)

13. Lim, Y.M., Ayesh, A., Stacey, M.: The effects of menu design on users' emotions, search performance and mouse behaviour. In: Patel, S., Wang, Y., Kinsner, W., Patel, D., Fariello, G., and Zadeh, L.A. (eds.) IEEE 13th international conference on cognitive informatics & cognitive computing, pp. 541–549. IEEE, London (2014)
14. Tsoulouhas, G., Georgiou, D., Karakos, A.: Detection of learner's™ affective state based on mouse movements. J. Comput. **3**, 9–18 (2011)
15. Pusara, M., Brodley, C.E.: User Re-authentication via Mouse Movements. In: Proceedings of the 2004 ACM workshop on visualization and data mining for computer security, pp. 1–8, ACM, New York (2004)
16. Shen, C., Cai, Z., Guan, X., Du, Y., Maxion, R.: User authentication through mouse dynamics. IEEE Trans. Inf. Forensics Secur. **8**(1), 16–30 (2012)
17. Shen, C., Cai, Z., Guan, X.: Continuous authentication for mouse dynamics: a pattern-growth approach. Dependable systems and networks (DSN). 42nd annual IEEE/IFIP international conference on 2012, pp. 1–12 (2012)
18. Shen, C., Cai, Z., Guan, X., Sha, H., Du, J.: Feature Analysis of Mouse Dynamics in Identity Authentication and Monitoring. In: Proceedings of the 2009 IEEE international conference on communications, pp. 673–677, IEEE Press, Piscataway (2009)
19. Vizer, L.M.: Detecting Cognitive and Physical Stress Through Typing Behavior. In: Proceedings of the 27th international conference extended abstracts on human factors in computing systems CHI EA 09. 3113 (2009)
20. Filho, J.R.M., Freire, E.O.: On the equalization of keystroke timing histograms. Pattern Recogn. Lett. **27**, 1440–1446 (2006)
21. Cavoukian, A.: Privacy by design: origins, meaning, and prospects for assuring privacy and trust in the information era. In: Yee, G. (ed.) Privacy protection measures and technologies in business organizations: aspects and standards, pp. 170–208. IGI Global, Hershey (2012)
22. IBM: multivariate general linear modeling, http://pic.dhe.ibm.com/infocenter/spssstat/v21r0m0/index.jsp?topic=/com.ibm.spss.statistics.cs/glmm_intro.htm
23. IBM: GLM univariate analysis, http://pic.dhe.ibm.com/infocenter/spssstat/v20r0m0/index.jsp?topic=/com.ibm.spss.statistics.help/idh_glmu.htm

Towards Using Games Theory to Detect New U2R Attacks

Mokrane Kemiche and Rachid Beghdad

Abstract In this paper we focused on proving that the linear programming model to detect new user to root (U2R) attacks cited in Beghdad R (Comput Commun 32:1104–1110, 2009 [1]), can be also modeled using games theory (GT). To do that, we will transform the whole linear model Beghdad R (Comput Commun 32:1104–1110, 2009 [1]) to a game theory model, and we will use the KDD99 (http://www.kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, [2]) dataset to prove that we will obtain the same detection rates (DRs). In Beghdad R (Comput Commun 32:1104–1110, 2009 [1]), the author formulated the problem of intrusion detection as a linear programming system (LPS) to test if an unknown behavior is close enough to a known behavior (*attack* or *normal*) such as we can conclude that it belongs to its class. Simulations results show that we obtained exactly the same results as those cited in Beghdad R (Comput Commun 32:1104–1110, 2009 [1]), and that our approach outperforms a set of recent approaches focusing on U2R attacks detection.

Keywords Computer systems · U2R attacks · Games theory · Linear program system · KDD dataset

M. Kemiche · R. Beghdad (✉)
Faculty of Sciences, Abderrahmane Mira University, 06000 Béjaïa, Algeria
e-mail: rachid.beghdad@gmail.com

M. Kemiche
e-mail: k.muqran@gmail.com

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_22

1 Introduction

Intrusion detection is based on the observation of the events and their analysis. First of all, Intrusion detection systems (IDS) collect the information to be analyzed. This information comes from system logs, specific applications (such as web servers, ftp servers, mail servers, etc.) or probes introduced by tools such as intrusion detection "sniffers" network, specific modules of certain applications, or the operating system. Secondly, the collected data is then analyzed for signs of intrusion, this analysis can be done in several ways: real-time or near real time.

The aim of the IDS is to report intrusions or attacks, following the design of the IDS, to the security administrator, in order to take appropriate measures, as to restore the system in a sure state. Research in intrusion detection field is also moving towards automatic intrusion response: in addition to preventing the security administrator, the intrusion detection system (then generally speaking IPS, for Intrusion Prevention System) take measures to block the intrusion: cut the TCP connection and change the rules of the firewall to block the IP address of the attacker for a network IPS, for example.

In practice, there are two classes of intrusion detection: approach based on knowledge or signatures (misuse detection), and the behavioral approach (anomaly detection). The first is based on knowledge of the techniques used by the attackers: one draws attack scenarios and founds out in the audit trail for such events. The second is based on the assumption that we can define a "normal" user behavior, and that any deviation from this, is potentially suspicious.

Many methods and frameworks have been developed to detect intrusions. Various techniques are also employed such as decision tree [3], artificial neural networks [4], association rules [5], clustering [6], support vector machines [7], ant colony [8]. In addition, many methods have been applied to detect known and new attacks, the known attacks are those belonging to both the KDD training and testing sets. The unknown attacks also called "new" attacks are those appearing only in the KDD testing set. In addition to that, there are four main categories of attacks: Denial of Service (DoS), Remote to Local (R2L), Probing, and User to Root (U2R).

In this article we will especially focus on detecting the latter category. It is a class of attacks where the attacker tries to have root access to the system. An example of U2R attacks is rootkit, which after getting root access to the intruder, replaces the system commands so (he/she) can come back whenever he wants as root. It is one of the categories that do not contain a lot of records in the whole KDD dataset. This prevents its detection with a very good rate in learning methods. In this article we will transform the linear model to detect U2R [1] to a model of games theory. A game includes a list of players, a set of strategy for each of them, and a function that gives their gains in all possible contingencies. A game with full information is a game where each player knows all the details of the model.

The rest of this paper is organized as follows. Section 2 presents related work. In Sect. 3, we present the LPS model to detect U2R. Our approach using GT will

be detailed in Sect. 4. An example of our approach is given in Sect. 4. Section 5 presents the experiments and simulations result. Section 6 concludes the paper.

2 Related Work

In this section, some previous works in intrusion detection models are briefly presented.

The authors of [9] proposed a method of intrusion detection based on the CAC algorithm (communicating ant for clustering). Firstly, the ant is evolving on a grid that contains connections of the KDD dataset set randomly. At each iteration the ants are on a cell which contains a connection. The first ant communicates with all the others lying on a cell that also contains a connection to gather similar connections in the same heap. We get two heaps at the end. The first contains the attacks, and the other the normal behavior. Secondly, this method was hybridized with association rules using the discrete attributes of the KDD Data Set after noting the low DR of R2L. The results of the conducted experiments reached respectively 96.68 and 96.67 % DR for known and unknown U2R attacks with respectively 3.32 and 3.33 % false alarms.

In [10] the authors proposed an iterative 3-tier model for selecting a subset of features. At level 1 the analysis of the raw data set is carried out using principal component analysis (PCA). At Level 2 several technique are used to determine the number of dominant principal components that should be retained according to the result of the analysis in Level 1. The refinement of the features, the generation and verification of the model of learning is done in Level 3. After that, they used the Mahalanobis distance (Mahalanobis Distance Map) to identify patterns of payload packet and extract hidden correlations between the characteristics of the payloads of network packets, and also partially capture structural information of the payload. They tested their system by performing several experiments on the data set 99 DARPA. The result of the conducted experiments has reached 96.29 % of detection rate and 3.71 % of false alarm.

The authors of [5] proposed a hybrid model for detection by misuse detection and anomaly detection, the misuse detection is done through a sequential hierarchical model with binary decision tree classifier at all levels. The approach separates one attack at a time. The C4.5 algorithm is used. For anomaly detection, the classifier based on association rules is used. These rules are generated using the algorithm CBA. If the sample data does not satisfy any of the normal rules defined for the normal profile, it is considered anomaly. The two previous models are integrated by a classifier based on decision tree at the first level; it separates the attack traffic (known signature) from normal traffic (containing both anomalies and normal traffic). The result of the experiments conducted has reached 99.98 % of detection rate and 35.24 % for new attacks.

In [11] the authors introduce an information fusion approach for detection and prediction of multistage stealthy cyber attacks. Their approach unifies

INFERD/TANDI developed by UB team and Markov Game theoretical threat intent inference [12] developed by IAI team to provide a better solution. There are two main parts: data fusion module and dynamic/adaptive feature recognition module. Various log file entities Alters generated by Intrusion Detection Sensors (IDSs) or Intrusion Prevention Sensors (IPSs) are fed into the L1 data fusion components. The fused objects and related pedigree information are used by a feature/pattern recognition module to generate primitive prediction of intents of cyber attackers. High-level (L2 and L3) data fusion based on Markov game model, Hierarchical Entity Aggregation (HEA) are proposed to refine the primitive prediction generated in stage 1 and capture new unknown features. Markov (Stochastic) game method is used to estimate the belief of each possible cyber attack graph. The captured unknown or new cyber attack patterns will be associated to related L1 results in dynamic learning block, which takes deception reasoning, trend/variation identification, and distribution model and calculation into account.

The author of [13] uses game theory to propose a series of optimal puzzle-based strategies for handling increasingly sophisticated flooding attack scenarios. The solution concept of Nash equilibrium is used not only in a descriptive way but also in prescriptive one. In doing so, the difficulty level of puzzles, random number generators, and the other parameters of a puzzle-based defense are so adjusted that the attacker's optimum strategy, prescribed by the Nash equilibrium, does not lead to the exhaustion of defender's resources. If the defender takes his part in the Nash equilibrium prescription as his defense against flooding attacks, the best thing for the attacker to do is to be in conformity with the prescription as well. In this way, the defense mechanism is effective against the attack and provides the maximum possible payoff for the defender. In other words, the defense mechanism is optimal. This notion is applied to a series of increasingly sophisticated flooding attack scenarios, culminating in a strategy for handling a distributed attack from an unknown number of sources. It is worth noting that two-player game models can also be used in the case of distributed attacks, where the attackers are modeled as a single player with the capabilities of the attack coalition.

According to the authors of [14], game theoretic defense mechanisms against DoS attacks focus on fine tuning the parameters of the system in such a way that the server is not overloaded by the attacker. The authors work builds on the game theoretic model and defense mechanisms proposed by Fallah [13]. Indeed, In addition to the basic properties of a good puzzle [15], they introduce the following requirement: the difficulty of the puzzle should not be determined by the attacker without expending a minimal amount of computational effort. They propose three concrete puzzles that satisfy this requirement. Using game theory, they show that defense mechanisms are more effective when the puzzle difficulty is hidden from the attacker.

In [16] we present a study about the use of some supervised learning techniques to predict intrusions. The aim of the research is to analyze the performances of such techniques to determine which one best addresses the intrusion detection problem. The performances of six machine learning algorithms involving C4.5, ID3, Classification and Regression Tree (CART), Multinomial Logistic Regression

(MLR), Bayesian Networks (BN), and CN2 rule-based algorithm are investigated. The "boosting-arcing" concept was used to obtain a better prediction model while executing a machine learning method. KDD'99 data sets were used to evaluate the considered algorithms. For these evaluations, three cases were considered: the whole attacks case, the five behaviors classes' case, and the two behaviors classes' case. Nevertheless, the U2R class of attacks was still poorly detected.

In [17], the authors focused on U2R attack, which the attacker tries to access normal user account and gains root access information of the system. The U2R attacks lead to several vulnerability such as sniffing password, a dictionary attack and social engineering attacks. The authors make a comparative study analyses for U2R attacks based on several popular machine learning techniques such as naive bayes, random forest, J48, random tree, JRIP and Multilayer perceptron to achieve better accuracy and to reduce mean square error for individual attacks that belongs to user to root category.

The authors of [18] proposed a hybrid multilevel IDS which uses a combination of decision tree classifier and an ant colony clustering algorithm, the resulting IDS achieves competitive detection rates. In the first level the enhanced C4.5 algorithm is used to classify connection record into Dos, Probe and "Others". The class of "Others" contains U2R, R2L and normal connections. In the second level, ant colony algorithm splits the data into two clusters, normal and abnormal traffic. The cluster with abnormal connections is distinguished easily because it has smaller in size. Finally, on the third level, the C4.5 algorithm classifies the abnormal traffic. The disadvantage of ants approach is the long execution time while clustering large data.

2.1 Discussions

The previous works cited in this section claim that their respective approaches are efficient in detecting old and new attacks, nevertheless their techniques still suffer from some weaknesses. On one hand, some approaches lead to low detection rate of both the old and new attacks. On the other hand, even if some of them reach high detection rate, nevertheless, they suffer from their high false alarm rates (>1 %). In this article we will transform the linear model of the article [1] to a model of games theory that detects all (100 %) the old and new attacks of the U2R attacks category, without any false alarm, by selecting appropriate KDD features.

3 Modeling the U2R Attack with LPS

In article [1] the author modeled intrusion detection in the form of a linear program to detect old and new attacks of the U2R class. This approach allows finding to which class belongs an attack (*known* or *new* attack) by calculating the distance

between the characteristic vector of an attack and characteristic vectors of each type of attack in the U2R class.

Formally, the model of the article [1] can be defined as follows:

- A behavior *i* belonging to an attack class *j* is represented by x_{ij}.
- Each attack class will be represented by its features vector F_j.
- The distance λ between behavior *i* and a class *j* is represented by λ_{ij}.

$$\lambda_{ij} = \sum_{\substack{i=1,n \\ j=1,n}} |f_i - f_j| \tag{1}$$

Such that:

- f_i is an element of Fi and f_j is an element of *Fj*.
- both *Fi* and *Fj* contain *n* items.

Features in bold in the Table 1 are selected to represent the features vectors of each attack.

According to the above assumptions, we have to find if an observed behavior k belongs to one of the known attack classes by minimizing the distance between this behavior and one of the existing classes.

So, for *n* behaviors and *m* attack classes:

$$Z_{min} = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_{ij} x_{ij} \tag{2}$$

- Behavior *i* can belong or not to an attack class *j*:

$$x_{ij} \in \{0, 1\} \tag{3}$$

Table 1 The selected features for LPS model [1]

Basic features		Content features		Traffic features		Other features	
A1	**Duration**	**A10**	**hot**	**A23**	**Count**	**A32**	**dst_host_count**
A2	protocol-type	A11	num_failed_logins	A24	srv_count	A33	dst_host_srv_count
A3	service	A12	logged_in	A25	serror_rate	A34	dst_host_same_srv_rate
A4	flag	A13	num_compromised	A26	srv_serror_rate	A35	dst_host_diff_srv_rate
A5	**src_bytes**	A14	root_shell	A27	rerror_rate	A36	dst_host_same_src_port_rate
A6	dsr_bytes	A15	su_attempted	A28	srv_rerror_rate	A37	dst_host_srv_diff_host_rate
A7	land	A16	num_root	A29	same_srv_rate	A38	dst_host_serror_rate
A8	wrong_fragment	A17	num_file_creations	A30	diff_srv_rate	A39	dst_host_srv_serror_rate
A9	urgent	A18	num_shells	A31	srv_dif_host_rate	A40	dst_host_rerror_rate
		A19	num_access_files			A41	dst_host_srv_rerror_rate
		A20	num_outbound_cmd				
		A21	is_host_login				
		A22	is_guest_login				

- For all the known behaviors, a behavior i belongs to one and only one attack class j such that $i = j$. So:

$$x_{ii} = 1 \tag{4}$$

For example, the "*buffer_overflow*" represents the first class of attacks ($x_{11} = 1$), and so on …

- A given behavior i can belong to one and only one attack class among m existing classes:

$$\sum_{j=1}^{m} x_{ij} = 1, \quad i = 1, 2, \ldots, n \tag{5}$$

Each attack type contains Oj elements and ($Oj \geq 0$):

$$\sum_{i=1}^{n} x_{ij} \geq O_j, \quad j = 1, 2, \ldots, m \tag{6}$$

This leads to the following system [1]:

$$\begin{cases} Z_{min} = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_{ij} x_{ij} \\ x_{ij} \in \{0, 1\} \\ x_{ii} = 1 \\ \sum_{j=1}^{m} x_{ij} = 1 \\ \sum_{i=1}^{n} x_{ij} \geq O_j \end{cases} \tag{7}$$

This system can be resolved by using the simplex algorithm [19].

In the following, we will formulate the detection of U2R attacks in the form of a non-cooperative game with complete information.

4 Modeling the U2R Attack with Games Theory

4.1 A Game

A game is an interaction between several individuals such that the result depends on the strategies of each individual. To forecast earnings of players at the end of a game, several concepts are used in games theory as the Nash equilibrium.

A game is described by the following:

1. Set of players I = {1 … n}.
2. For each player $i \in$ I, a set of strategies $S_i = \{s_i^1, s_i^2, \ldots, s_i^{ki}\}$ which contains all possible strategies of player.

3. For each player $i \in \mathrm{I}$, a gain function, u_i, which represents the preferences of player i by assigning the value of the game result for each player i: $u_i(s)$.

Our game will represent the interaction between an *intruder* and an *intrusion detection* system.

4.2 Players

The participants in a game are called players. In our game model, there are two players: *player 1* is the intruder who attacks the computer system, and *player 2* is the intrusion detection system that tries to detect the attack.

$$\mathrm{I} = \{\text{intruder, Ids}\}$$

4.3 Strategies

A strategy is the complete specification of the behavior of a player in any situation.

Player 1 (the intruder) has a choice that his behavior is normal or a known attack such as: buffer_oveflow, loadmodule, or rootkit, or an unknown attack (new). Player 2 (intrusion detection system) has the same five strategies mentioned above: (*Normal*, *new*, *buffer_overflow*, *loadmodule*, and *rootkit*) to try to detect any attack.

So we have $\mathrm{S1} = \mathrm{S2} = \{normal, new, buffer\_overflow, loadmodule, rootkit\}$

4.4 The Utility Function

The attack of the intruder (player 1) is represented by a vector of characteristics F and each class of attack k is represented by the vector of characteristics $\mathrm{F^k}$, the function of the game utility is the distance between the vector of behavior F and the vector of the class of attack k as follows:

$$\mathrm{U_1}\left(\mathrm{s_k}, \mathrm{s_{k'}}\right) = \sum \left|\mathrm{f}_i - \mathrm{f_j^k}\right| \tag{8}$$

Such that f_i is an element of the vector F and f_j^k an element of the vector $\mathrm{F^k}$.

For player 2 (intrusion detection system), the gain function is the distance between the vector of the attack class k of player 1, and the vector of the attack class k' of player 2:

$$\mathrm{U_2}\left(\mathrm{s_k}, \mathrm{s_{k'}}\right) = \sum \left|\mathrm{f_i^k} - \mathrm{f_j^{k'}}\right| \tag{9}$$

Such that f_i^k is an element of the vector $\mathrm{F^k}$ and $f_j^{k'}$ an element of the vector $\mathrm{F^{k'}}$ (Table 2).

Table 2 The utility functions of the intruder and the IDS

		IDS				
		Buff_overflow	Loadmodule	Rootkit	Normal	New
Intruder	Buffer_ overflow	$\left(\sum \left\|f_i - f_j^1\right\|, \sum \left\|f_i^1 - f_j^1\right\|\right)$	$\left(\sum \left\|f_i - f_j^1\right\|, \sum \left\|f_i^2 - f_j^1\right\|\right)$	$\left(\sum \left\|f_i - f_j^1\right\|, \sum \left\|f_i^3 - f_j^1\right\|\right)$	$\left(\sum \left\|f_i - f_j^1\right\|, \sum \left\|f_i^4 - f_j^1\right\|\right)$	$\left(\sum \left\|f_i - f_j^1\right\|, \sum \left\|f_i^5 - f_j^1\right\|\right)$
	Loadmodule	$\left(\sum \left\|f_i - f_j^2\right\|, \sum \left\|f_i^1 - f_j^2\right\|\right)$	$\left(\sum \left\|f_i - f_j^2\right\|, \sum \left\|f_i^2 - f_j^2\right\|\right)$	$\left(\sum \left\|f_i - f_j^2\right\|, \sum \left\|f_i^3 - f_j^2\right\|\right)$	$\left(\sum \left\|f_i - f_j^2\right\|, \sum \left\|f_i^4 - f_j^2\right\|\right)$	$\left(\left(\sum \left\|f_i - f_j^2\right\|, \sum \left\|f_i^5 - f_j^2\right\|\right)\right.$
	Rootkit	$\left(\sum \left\|f_i - f_j^3\right\|, \sum \left\|f_i^1 - f_j^3\right\|\right)$	$\left(\sum \left\|f_i - f_j^3\right\|, \sum \left\|f_i^2 - f_j^3\right\|\right)$	$\left(\sum \left\|f_i - f_j^3\right\|, \sum \left\|f_i^3 - f_j^3\right\|\right)$	$\left(\sum \left\|f_i - f_j^3\right\|, \sum \left\|f_i^4 - f_j^3\right\|\right)$	$\left(\sum \left\|f_i - f_j^3\right\|, \sum \left\|f_i^5 - f_j^3\right\|\right)$
	Normal	$\left(\sum \left\|f_i - f_j^4\right\|, \sum \left\|f_i^1 - f_j^4\right\|\right)$	$\left(\sum \left\|f_i - f_j^4\right\|, \sum \left\|f_i^2 - f_j^4\right\|\right)$	$\left(\sum \left\|f_i - f_j^4\right\|, \sum \left\|f_i^3 - f_j^4\right\|\right)$	$\left(\sum \left\|f_i - f_j^4\right\|, \sum \left\|f_i^4 - f_j^4\right\|\right)$	$\left(\sum \left\|f_i - f_j^4\right\|, \sum \left\|f_i^5 - f_j^4\right\|\right)$
	New	$\left(\sum \left\|f_i - f_j^5\right\|, \sum \left\|f_i^1 - f_j^5\right\|\right)$	$\left(\sum \left\|f_i - f_j^5\right\|, \sum \left\|f_i^2 - f_j^5\right\|\right)$	$\left(\sum \left\|f_i - f_j^5\right\|, \sum \left\|f_i^3 - f_j^5\right\|\right)$	$\left(\sum \left\|f_i - f_j^5\right\|, \sum \left\|f_i^4 - f_j^5\right\|\right)$	$\left(\sum \left\|f_i - f_j^5\right\|, \left\|\sum f_i^5 - f_j^5\right\|\right)$

Table 3 The correspondence between LPS and games theory

	The LPS model	Game theory
Objective function	$Z_{min} = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_{ij} x_{ij}$	Utility function: Player1: $U_1(s_k, s_{k'}) = \sum \lvert f_i - f_j^k \rvert$ Player2 (IDS): $U_2(s_k, s_{k'}) = \sum \lvert f_i^k - f_j^{k'} \rvert$
The constraints	$\sum_{j=1}^{m} x_{ij} = 1$	In game theory, each player chooses a single strategy which is equivalent to this constraint
	$\sum_{i=1}^{n} x_{ij} \geq O_j$	There is no need to this constraint in games theory

4.5 Equivalence Between the Linear Model and Game Theory

The following Table 3 illustrates the correspondence between the linear model of the article [1] and the games theory model.

4.6 The Game Solution

The Nash equilibrium is a solution that corresponds to correct expectations of players. By choosing Nash equilibrium as a solution, each player is thinking that others will do likewise. That is to say that the combinations of strategies of players are all the better response of each player in the strategies chosen by others. Formally: a profile $p^* = (p_1^*, \ldots, p_n^*)$ $(p_i^* \in \mathrm{P_i},\ i = 1 \ldots n)$ is a Nash equilibrium if:

$$u_i(p_i^*, p_{-i}^*) \geq u_i(p_i, p_{-i}^*), \quad \forall p_i \in \mathrm{P_i},\ \forall i = 1 \ldots n. \tag{10}$$

Before seeking equilibrium of a game, we can try to simplify the game by eliminating redundant strategies and strategies that are dominated by others. This type of disposal will simplify our game and we will facilitate the search of a solution. The elimination of strategies must be done with thought and care.

Two strategies s_i and s_i' are equivalent if and only if all players get the same utility when i plays s_i or s_i'.

$$\forall j \in \mathrm{I},\ \forall s_{-i} \in S_{-i},\ u_j(s_i, s_{-i}) = u_j(s_i',\ s_{-i}) \tag{11}$$

The strategy s_i of player i is dominated by the strategy s_i' if and only if, regardless of the behavior of the other players, the player i receives with s_i a lower gain than that obtained with s_i'.

$$\forall s_{-i} \in S_{-i},\ u_i(s_i, s_{-i}) < u_i(s_i',\ s_{-i}) \tag{12}$$

In our model the two players must choose the minimum solution which corresponds to the choice of strategies which give a minimum distance, i.e. minimizing their gain.

5 Example

Let us now consider a computer system that is corrupted by some attacks of the U2R category. Some of these attacks (three known attacks of U2R) are supposed defined by the following features vectors ($F^{buffer\_overflow}$, $F^{loadmodule}$, and $F^{rootkit}$), and the normal behavior is defined by F^{normal}. In addition, we defined here a features vector F^{New} that represents the behavior of a *new* attack class. We suppose that five unknown behaviors enter the system, and we have to determine the class of each behavior.

$$F^{buffer\_overflow} = [\,1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\,]$$
$$F^{loadmodule} = [\,10\ 20\ 30\ 40\ 50\ 60\ 70\ 80\ 90\ 100\,]$$
$$F^{normal} = [\,20\ 40\ 60\ 80\ 100\ 120\ 140\ 160\ 180\ 200\,]$$
$$F^{rootkit} = [\,30\ 60\ 90\ 120\ 150\ 180\ 210\ 240\ 270\ 300\,]$$
$$F^{New} = [\,40\ 80\ 120\ 160\ 200\ 240\ 280\ 320\ 360\ 400\,]$$

$$F^{attack1} = [\,5\ 12\ 6\ 2\ 10\ 8\ 17\ 28\ 21\,]$$
$$F^{attack2} = [\,26\ 75\ 151\ 160\ 150\ 144\ 60\ 136\ 91\ 126\,]$$
$$F^{attack3} = [\,197\ 115\ 20\ 99\ 31\ 62\ 132\ 47\ 131\ 10\,]$$
$$F^{attack4} = [\,77\ 152\ 120\ 212\ 119\ 240\ 135\ 266\ 287\ 253\,]$$

We will compute the utility function for each player like this:

Player 1:

U_1 (buffer_overflow, buffer_overflow) = U_1(buffer_overflow, loadmodule)
= U_1 (buffer_overflow, rootkit) = U_1(buffer_overflow, normal)
= U_1 (buffer_overflow, new) = $\sum \left| f_i^{attack1} - f_j^{buffer\_overflow} \right| = 72$
U_1 (loadmodule, buffer_overflow) = U_1(loadmodule, loadmodule)
= U_1 (loadmodule, rootkit) = U_1(loadmodule, normal)
= U_1 (loadmodule, new) = $\sum \left| f_i^{attack1} - f_j^{loadmodule} \right| = 459$
U_1 (rootkit, buffer_overflow) = U_1(rootkit, loadmodule) = U_1(rootkit, rootkit)
= U_1 (rootkit, normal) = U_1(rootkit, new) = $\sum \left| f_i^{attack1} - f_j^{rootkit} \right| = 1559$
U_1 (normal, buffer_overflow) = U_1(normal, loadmodule) = U_1(normal, rootkit)
= U_1 (normal, normal) = U_1(normal, new) = $\sum |f_i^{attack1} - f_j^{normal}| = 1009$
U_1 (new, buffer_overflow) = U_1(new, loadmodule) = U_1(new, rootkit) = U_1(new, normal) = U_1(new, new)=$\sum |f_i^{attack1} - f_j^{new}| = 2109$

Player 2:

U_2(buffer_overflow, buffer_overflow) = $\sum \left| f_i^{buffer\_overflow} - f_j^{buffer\_overflow} \right| = 0$

U_2(buffer_overflow, loadmodule) = U_2(loadmodule, buffer_overflow)
= $\sum \left| f_i^{buffer\_overflow} - f_j^{load\_module} \right| = 495$

U_2 (buffer_overflow, rootkit) = U_2(rootkit, buffer_overflow) = $\sum \left| f_i^{\text{buffer\_overflow}} - f_j^{\text{rootkit}} \right| = 1595$

U_2 (buffer_overflow, normal) = U_2 (normal, buffer_overflow) = $\sum \left| f_i^{\text{buffer\_overflow}} - f_j^{\text{normal}} \right| = 1045$

U_2 (buffer_overflow, new) = U_2 (new, buffer_overflow) = $\sum \left| f_i^{\text{buffer\_overflow}} - f_j^{\text{new}} \right| = 2145$

U_2 (loadmodule, loadmodule) = $\sum |f_i^{\text{loadmodule}} - f_j^{\text{loadmodule}}| = 0$

U_2 (loadmodule, rootkit) = U_2(rootkit, loadmodule) = $\sum |f_i^{\text{loadmodule}} - f_j^{\text{rootkit}}| = 1100$

U_2 (loadmodule, normal) = U_2(normal, loadmodule) = $\sum |f_i^{\text{loadmodule}} - f_j^{\text{normal}}| = 550$

U_2 (loadmodule, new) = U_2 (new, loadmodule) = $\sum |f_i^{\text{loadmodule}} - f_j^{\text{new}}| = 1650$

U_2 (rootkit, rootkit) = $\sum |f_i^{\text{rootkit}} - f_j^{\text{rootkit}}| = 0$

U_2 (rootkit, normal) = U_2 (normal, rootkit) = $\sum |f_i^{\text{rootkit}} - f_j^{\text{normal}}| = 550$

U_2 (rootkit, new) = U_2 (new, rootkit) = $\sum |f_i^{\text{rootkit}} - f_j^{\text{new}}| = 550$

U_2 (normal, normal) = $\sum |f_i^{\text{normal}} - f_j^{\text{normal}}| = 0$

U_2 (normal, new) = U_2 (new, normal) = $\sum |f_i^{\text{normal}} - f_j^{\text{new}}| = 1100$

U_2 (normal, normal) = $\sum |f_i^{\text{new}} - f_j^{\text{new}}| = 0$

We note in Table 4 below, that this game contains dominated strategies for player 1 (intruder), since:

$$\forall s_{-i} \in S_2, u_1(s_1, s_{-i}) < u_1(s_i', s_{-i})$$

Table 4 Table of gain

		Ids				
		Buff_overflow	Loadmodule	Rootkit	Normal	New
Intrus	Buffer_ overflow	(72,0)	(72,495)	(72, 1595)	(72, 1045)	(72,2145)

Table 5 Elimination of dominated strategies to deduce *buffer_overflow* attack

		Ids				
		Buff_overflow	Loadmodule	Rootkit	Normal	New
Intruder	Buffer_over-flow	(72,0)	(72,495)	(72, 1595)	(72, 1045)	(72,2145)
	Loadmodule	(459,495)	(459,0)	(459,1100)	(459,550)	(459,1650)
	Rootkit	(1559,1595)	(1559,1100)	(1559,0)	(1559,550)	(1559,550)
	Normal	(1009,1045)	(1009,550)	(1009,550)	(1009,0)	(1009,1100)
	New	(2109,2145)	(2109,1650)	(2109,550)	(2109,1100)	(2109,0)

After elimination of its dominated strategies, the reduced game is obtained in Table 4. So, the attack which represented by the vector [5 12 6 2 10 8 17 28 2 1] belongs to *buffer_overflow* class (see Table 5).

The procedure is the same for other vectors of attack and the results are the following: (Tables 6, 7, 8 and 9).

So, the vector [26 75 151 160 150 144 60 136 91 126] belong to *normal* class

So, the vector [197 115 20 99 31 62 132 47 131 10] belong to *loadmodule* class

So, the vector [77 152 120 212 119 240 135 266 287 253] belong to *rootkit* class

So, the vector [105 387 251 229 252 363 338 232 131 174] belong to *new* class

We note that the results of our model are exactly the same as those of the linear programming model [1]. Indeed, in [1], the author found: ($x_{61} = 1$), and the behavior

Table 6 Elimination of dominated strategies to deduce *normal* behavior

		IDS				
		Buff_overflow	Loadmodule	Rootkit	Normal	New
Intrusder	Buffer_over-flow	(1064,0)	(1064,495)	(1064, 1595)	(1064, 1045)	(1064,2145)
	Loadmodule	(589,495)	(589,0)	(589,1100)	(589,550)	(589,1650)
	Rootkit	(763,1595)	(763,1100)	(763,0)	(763,550)	(763,550)
	Normal	(553,1045)	(553,550)	(553,550)	(553,0)	(553,1100)
	New	(1148,2145)	(1148,1650)	(1148,550)	(1148,1100)	(1148,0)

Table 7 Elimination of dominated strategies to deduce *loadmodule* attack

		Ids				
		Buff_overflow	Loadmodule	Rootkit	Normal	New
Intrusder	Buffer_over-flow	(789,0)	(789,495)	(789, 1595)	(789, 1045)	(789,2145)
	Loadmodule	(598,495)	(598,0)	(598,1100)	(598,550)	(598,1650)
	Rootkit	(1250,1595)	(1250,1100)	(1250,0)	(1250,550)	(1250,550)
	Normal	(798,1045)	(798,550)	(798,550)	(798,0)	(798,1100)
	New	(1740,2145)	(1740,1650)	(1740,550)	(1740,1100)	(1740,0)

Table 8 Elimination of dominated strategies to deduce *rootkit* attack

		IDS				
		Buff_overflow	Loadmodule	Rootkit	Normal	New
Intrusder	Buffer_over-flow	(1806,0)	(1806,495)	(1806, 1595)	(1806, 1045)	(1806,2145)
	Loadmodule	(1311,495)	(1311,0)	(1311,1100)	(1311,550)	(1311,1650)
	Rootkit	(517,1595)	(517,1100)	(517,0)	(517,550)	(517,550)
	Normal	(771,1045)	(771,550)	(771,550)	(771,0)	(771,1100)
	New	(661,2145)	(661,1650)	(661,550)	(661,1100)	(661,0)

Table 9 Elimination of dominated strategies to deduce *new* attack

		IDS				
		Buff_overflow	Loadmodule	Rootkit	Normal	New
Intrusder	Buffer_over-flow	(2407,0)	(2407,495)	(2407, 1595)	(2407, 1045)	(2407,2145)
	Loadmodule	(1912,495)	(1912,0)	(1912,1100)	(1912,550)	(1912,1650)
	Rootkit	(1358,1595)	(1358,1100)	(1358,0)	(1358,550)	(1358,550)
	Normal	(1512,1045)	(1512,550)	(1512,550)	(1512,0)	(1512,1100)
	New	(1348,2145)	(1348,1650)	(1348,550)	(1348,1100)	(1348,0)

N1 is a *buffer_overflow* attack; ($x_{73} = 1$), and the behavior N2 is a *normal* behavior; ($x_{82} = 1$), and the behavior N3 is a *loadmodule* attack; ($x_{94} = 1$), and the behavior N4 is a *rootkit* attack; ($x_{105} = 1$), and the behavior N5 is a "*New*" attack.

6 Experiments

6.1 Simulation Parameters

To achieve the same results as those of the article, we conducted experiments with exactly the same parameters as those used in [1]: they used the 11 features: A1, A10 A12 A13 A14 A16 A17 A18 A22 A23 A24 (attributes in bold described in Table 1) and ignored the following three features: A5, A32, A33 because they produce false negatives. The learning and the testing datasets used in our experiments are composed with the same sample [1] selected from KDD dataset. Each features vector used in this section is composed of mean values (each element of each vector is a mean value).

The learning dataset is composed of 78 occurrences: normal (26), buffer_overflow (30), load_module (9), perl (3), rootkit (10). The testing dataset is composed of 161 occurrences: normal (50), buffer_overflow (21), *httptunnel* (42), *loadmodule* (2), *perl* (2), *ps* (16), *rootkit* (13), *sqlattack* (2), *xterm* (13).

The different U2R attacks and the normal behavior were represented by the following vectors:

$$F^{\text{normal}} = [\,0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 15.5\ 15.5\,]$$
$$F^{\text{buffer\_overflow}} = [\,115.5\ 3\ 1\ 2\ 1\ 0\ 0\ 0\ 0\ 1\ 1\,]$$
$$F^{\text{loadmodule}} = [\,821\ 0.5\ 0.5\ 0.5\ 0\ 2\ 1\ 0\ 1\ 1\,]$$
$$F^{\text{perl}} = [\,35\ 0\ 1\ 0\ 1\ 2\ 2\ 1\ 0\ 1\ 1\,]$$
$$F^{\text{rootkit}} = [\,30\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1.5\ 1.5\,]$$
$$F^{\text{httptunnel}} = [\,115.5\ 3\ 1\ 2\ 1\ 0\ 0\ 0\ 0\ 1\ 1\,]$$
$$F^{\text{ps}} = [\,168,72\ 2.07\ 0.79\ 3.71\ 0.38\ 2.36\ 1.26\ 0.29\ 0\ 1.13\ 1.12\,]$$
$$F^{\text{sqlattack}} = [\,167.54\ 2.12\ 0.83\ 3.66\ 0.41\ 2.51\ 1.43\ 0.34\ 0\ 1.12\ 1.12\,]$$
$$F^{\text{xterm}} = [\,92.43\ 1.29\ 0.68\ 8.56\ 0.28\ 10.35\ 0.87\ 0.35\ 0\ 1.01\ 1.03\,]$$

Table 10 Simulations results

	Buffer_over-flow (%)	Loadmodule (%)	Perl (%)	Rootkit (%)	Normal (%)	New attacks (%)
Detection rate	100	100	100	100	100	100
False alarm rate	0	0	0	0	0	0

Table 11 Comparing our approach to similar approaches

	**		[9]		[10]		[5]		[16]		[17]		[18]	
	DR	FA	DR	FA	DR	FA	DR	FA	DR	FA	DR	FA	DR	FA
Normal	**100**	**0**	98.08	1.92	–	3.71	–	–	94,42	–	–	–	–	–
Old attack	**100**	**0**	96.68	3.32	96.29		99.98	–	67,11	–	88,46	–	–	–
New attack	**100**	**0**	96.67	3.33	–		35.24	–	?	–	–	–	35.94	3.37

**Our approach

6.2 Simulation Results

The results of our experiments are summarized in Table 10. In one hand, according to this result, all the tested behaviors were assigned to their respective classes, and all the ‘‘new’’ attacks were entirely detected. In the other hand, the detection rate of our method is 100 % and the false alarm rate is 0 % this is exactly what has been achieved in the article [1].

6.3 Efficiency of Our Approach

To demonstrate the efficiency of our contribution we compared both the detection rate (DR) and the false alarm rate (FA) of our approach to those of some already proposed methods. The following table summarizes the results.

According to Table 11, we can conclude that our approach highly outperforms all previous approaches used to detect old and new U2R attacks.

7 Conclusion

In this paper we have transformed a linear programming model [1] to detect U2R attacks in a model of game theory. This theory is a very important mathematical discipline that can be used to analyze systems in different domains. The game model check if a behavior in a computer system is close enough to a known behavior class (attack or normal) such as we can deduce that it belongs to it. We evaluated its performance by achieving some simulations. Even if many methods were proposed to solve the same problem, nevertheless they suffer from some

disadvantages. These methods lead to both low detection rate and high false alarm rate of the new U2R attacks. Our simulations show that our approach outperforms some already proposed approaches.

Our future work will focus on detecting attacks of the remaining attack categories (DoS, Probing, and R2L), with higher DR and lower false alarms, if any, and implementing our approach in a given system.

References

1. Beghdad, R.: Efficient deterministic method for detecting new U2R attacks. Comput. Commun. **32**(6), 1104–1110 (2009)
2. KDD Data Set: Available from http://www.kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (1999)
3. Lee, J.H., Lee, J.H., Sohn, S.G., Ryu, J.H., Chung, T.M.: Effective value of decision tree with KDD 99 intrusion detection datasets for intrusion detection system. In: 10th IEEE International Conference in Advanced Communication Technology, 2008. ICACT 2008, vol. 2, pp. 1170–1175 (2008)
4. Kukiełka P, Kotulski Z: Analysis of neural networks usage for detection of a new attack in IDS. In: Annales UMCS, Informatica, vol. 10(1), pp. 51–59. Versita, Warsaw (2010)
5. Goel, R., Sardana, A., Joshi, R.C.: Parallel misuse and anomaly detection model. Int. J. Netw. Secur. **14**(4), 211–222 (2012)
6. Horng, S.J., Su, M.Y., Chen, Y.H., Kao, T.W., Chen, R.J., Lai, J.L., Perkasa, C.D.: A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Syst. Appl. **38**(1), 306–313 (2011)
7. Feng, W., Zhang, Q., Hu, G., et al.: Mining network data for intrusion detection through combining SVMs with ant colony networks. Future Gener. Comput. Syst. **37**, 127–140 (2013)
8. Agravat, D., Vaishnav, U., Swadas, P.B.: Modified ant miner for intrusion detection. In: 2010 Second International Conference on Machine Learning and Computing (ICMLC), pp. 228–232. IEEE, (2010)
9. Kemiche, M., Beghdad, R.: CAC-UA: a communicating ant for clustering to detect unknown attacks. IEEE (SAI) 2014, pp. 515-522. London, UK, (2014)
10. Jamdagni, A., Tan, Z., He, X., Nanda, P., Liu, R.P. RePIDS: A multi tier real-time payload-based intrusion detection system. Comput. Netw. (2013)
11. Shen, D., Chen, G., Cruz, Jr, J.B., Blasch, E., Kruger, M.: RI game theoretic solutions to cyber attack and network defense problems. In: The Proceedings of the Twelfth International Command and Control Research and Technology Symposium (12th ICCRTS), Newport (2007)
12. Shen, D., Chen, G., Cruz, Jr, J.B., et al.: Game theoretic approach to threat intent prediction. In: Proceedings of the Command and Control Research and Technology Symposium CCRTS 2006. San Diego, (2006)
13. Fallah, M.S.: A puzzle-based defense strategy against flooding attacks using game theory. IEEE Trans. Dependable Secure Comput. Arch. **7**(1), 5–19 (2010)
14. Narasimhan, H., Varadarajan, V., Rangan, C.P.: Game theoretic resistance to denial of service attacks using hidden difficulty puzzles. Information Security, Practice and Experience, Lecture Notes in Computer Science, vol. 6047, pp. 359–376. Springer, Berlin (2010)
15. Smith, J.: Denial of service: prevention, modelling and detection. Ph.D. thesis, Queens land University of Technology, Brisbane, QLD 4001 Australia (2007)
16. Beghdad, R.: Critical study of supervised learning techniques in predicting attacks. Inf. Sec. J. Global Perspect. **19**(1), 22–35 (2010)

17. Revathi, S., Malathi, A.: Detecting user-to-root (U2R) attacks based on various machine learning techniques. Int. J. Adv. Res. Comput. Commun. Eng. **3**(4), 6322–6324 (2014)
18. Rajeswari, L.P., Kannan, A., Baskaran, R.: An escalated approach to ant colony clustering algorithm for intrusion detection system. Distributed Computing and Networking. Lecture Notes in Computer Science, vol. 4904, pp. 393–400. Springer, Berlin (2008)
19. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms (2 ed.) MIT Press and McGraw-Hill, ISBN 0-262-03293-7. Section 29.3: The simplex algorithm, pp. 790–804 (2001)

Development and Evaluation of Virtual Reality Medical Training System for Anatomy Education

Jannat Falah, Vassilis Charissis, Soheeb Khan, Warren Chan, Salsabeel F.M. Alfalah and David K. Harrison

Abstract Medical education is a dynamic field that witnesses continuous evolution and development. The employment of Virtual Reality (VR) based visualization and training environments in the delivery of anatomy teaching transfers the learning experience from one that involves memorising the structures without a true understanding of the 3-Dimensional (3D) relations, to a process that involves a thorough understanding of the structure based on visualisation rather than memorising, which makes the learning process more efficient and enjoyable, and less time consuming. This paper describes the development of a Virtual Reality and 3D visualisation system for anatomy teaching. The developed system offers a real-time 3D representation of the heart in an interactive VR environment that provides self-directed learning and assessment tools through a variety of interfaces and functionalities. To ensure the accuracy and precision of the developed system it was evaluated by a group of medical professionals.

J. Falah · V. Charissis (✉) · S. Khan · W. Chan · D.K. Harrison
School of Engineering and Built Environment, Department of Computer, Communication and Interactive Systems, Virtual Reality and Simulation Laboratory, Glasgow Caledonian University, Glasgow G4 0BA, UK
e-mail: Vassilis.charissis@gcu.ac.uk; v.charissis@gmail.com

J. Falah
e-mail: Jannat.falah@gcu.ac.uk

S. Khan
e-mail: Soheeb.khan@gcu.ac.uk

W. Chan
e-mail: Warren.chan@gcu.ac.uk

D.K. Harrison
e-mail: D.K.Harrison@gcu.ac.uk

S.F.M. Alfalah
King Abdullah II School for Information Technology, University of Jordan, Amman, Jordan
e-mail: S.alfalah@ju.edu.jo

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_23

Keywords Medical education · Virtual reality · Anatomy · 3D · Heart

1 Introduction

The field of medical education has evolved over the last decade due to technological advances in medical visualisation and computing overall. The latter supported the development of innovative teaching methods that are as realistic as possible in resembling the human body in all its complexity and variety. These technological endeavours in medicine aim to complement current medical training methods. Yet on current and future generations of medical practitioners the educational methods could benefit dramatically from technologies that these users are already accustomed to. Therefore providing, online information through various mediums such as 3DTV, PC, tablets and smart-phones could provide a more time-flexible and customisable training conduit [1–4]

Evidently the live human body cannot directly be used to provide a teaching method for obvious safety and ethical related reasons. In turn, cadaveric material has a number of limitations based on the chemically affected tissue having reduced properties of texture, colour, thickness and flexibility [2, 3]. Additionally the cadaver has limited lifespan and has no capacity of repetition of dissections or reversing processes. These physical limitations render the traditional methods unusable out with the confinements of a laboratory or specialised facility. The representation of the human body in a digital, three-dimensional space is deemed ideal for the enhancement of the training in the complex field of medicine. The reason is that a detailed 3D model can be viewed in various devices and optimally in a Virtual Reality (VR) enabled system. The latter intrigued Medical Educationalists which have become increasingly interested in the inherent opportunities that these techniques offer [3, 4, 5].

Such systems could improve the mental 3D mapping of human organs and create a logical spatial awareness of each anatomical part [6, 7]. It has to be noted that the proposed VR system is designed to enhance the knowledge acquisition by the student and not replace completely the existing methods. If the student experiments safely on a the VR environment could interact confidently with the live body in later stages. The VR system enables the user to investigate the human body from infinite viewing points as the 3D model can be manipulated in space and alter the size (magnify specific organs or areas) and relative position anatomical information [2, 3, 6]. The system can offer also summarised or analytical information related to each selection. As such the alphanumeric information can be introduced in the interface seemingly alongside with the 3D visuals on demand.

This paper presents the design and development challenges for the proposed VR Medical System. The following section will offer a background review of emerging technologies. Section 3 will present the design and implementation of the related 3D context and will elaborate on the Human-Computer Interface (HCI) design of the VR system. Section 4 will present the system evaluation and Sect. 5 will

analyse and discuss the derived results. Finally Sect. 6 will present the conclusion of the paper and offer the tentative plan for future work aiming towards the introduction of various pathologies in the specific anatomical model.

2 Background

Historically, anatomy has been the backbone in the medical education regardless of specialty [8] as it provides the framework for any variation or pathology of the human body. Anatomy teaching depends mostly on dissection and instructional lecturers. The process of dissection provides students with significant three-dimensional views of the human anatomy structures, which cannot be gained from textbooks or 2D images [9]. However, there are some obvious issues with the traditional modalities such as the quality of training and cadavers storing amongst others [10]. Yet we anticipate that integration of new bespoke technological solutions and applications will improve students' anatomical knowledge and will overcome these issues. Furthermore the traditional modalities used in medical education are associated with numerous issues as they provide a limited spatial understanding obtained from didactic lectures and restricted anatomic dissection [8, 11]. Students assimilate knowledge more efficiently when they participate and interact with their learning context by self-directed activities [12]. Moreover, students who depend on visual aids exterior to the dissection room scored significantly higher on examination [13]. To further consolidate the learning experience, plastic 3D models, simulation, imaging, and videos were engaged in anatomy teaching.

In addition, a spatial relationship understanding from traditional modalities (textbooks and 2-Dimensional (2D) images) is not clear; requires expertise to explicate, and lacks adequate detail to demonstrate a specific teaching point [9]. Furthermore, the human cadaver which is one of the teaching anatomy methods is associated with several limitations, for instance, the rising costs, decreasing availability, and the decay in quality of useful material [13, 14, 15]. These issues are overcome by 3D modelling which has enabled visualization of the spatial relationships between structures from various viewpoints, is reusable, is of changeable size, and allows explorative details which improve understanding, [9].

To improve the realism of visualisation and spatial simulation, Virtual Reality was identified as a suitable conduit. Previous studies employed successfully Virtual Reality (VR) and high fidelity patient simulation, in order to improve and enhance the medical education and clinical training [16, 17]. The benefits of this technological adoption in the teaching field offered safer experimentation environments, reduced time and cost [2]. Furthermore the Virtual Reality facilities and systems can be extensively customised in relatively low cost and be re-used for various applications [2, 3, 6].

Virtual Reality is gradually spreading as a teaching aid due to the aforementioned benefits [18]. In medicine this technology facilitates several teaching and diagnostic activities. In turn, VR teaching methods enable the students to

investigate the human body and create a three-dimensional mental picture of the human body structures and relationships. Real-time VR applications offer a rich, interactive, and highly engaging educational context, thus supporting experimental learning-by-doing. Notably, it can, contribute to increased interest and inspiration in students and to effectively support skill development [4, 8, 13].

3 VR System Development

Adhering to the aforementioned observations and previous studies we have developed a prototype Virtual Reality system for medical training. Our research focused primarily on anatomy teaching. We identified through consultation with medical practitioners the anatomical cases which typically are amongst the most complex and difficult to teach and comprehend. As such, the case study was based on heart anatomy. Our development challenges were falling in two main categories namely HCI and 3D context. Although both elements were designed in tandem, we had to develop first the 3D content so as to enable the development and evaluation of each interface tool based on real 3D data.

3.1 3D Context Development

The 3D heart model was developed using the industry-standard Autodesk Maya 3D visualisation software. The model development was informed by ultra high-definition photographic material, CT scans, X-rays, Ultrasound images, 2D images, schematics and specialised books' documentation.

Notably the developing team avoided intentionally the use of any laser scanning of live or cadaver material as this did not offer any benefits in the accuracy or speed of the 3D visualisation process. In particular, prior experimentations with hand-held laser scanners could not scan live organs as the breathing process of patients or other pulsing motion (i.e. heart-beat) distorted the scanning process. The result of such scanning offers multiple point-clouds that could not form any type of surface. Further experimentation with live tissue and organs removed from the human body presented additional issues as the flexing of the live tissue distorts the shape of the organ and therefore any attempt to acquire the volumetric information through laser scanning is rendered useless. Finally the acquisition of 3D information from a cadaver through laser scanning is also unusable as the embalmed human organs lose significant volume, texture and colour in contrast to the live ones. Moreover the dissection process can further damage the actual 3D structure of the human anatomy and produce inaccurate and unusable material. The particular method can be used mainly for facial scanning and reconstruction which primarily depicts skin-surface information and not internal deformable organs [19].

Throughout the development of the 3D context our visualisation specialists have been closely supported by specialist consultant medical doctors. The medical practitioners provided crucial information not only for anatomical characteristics but most importantly for the teaching process and sequence that the model has to reveal to the medical students in order to comply with the current curriculum [2, 3].

The final 3D model followed a polygon modelling production methodology which provides the developer with complete control over the construction of the three-dimensional geometry. This was deemed important as the final 3D output should be optimised further for a real-time engine. Notably the VR Anatomy System required dissection of various heart structures to reveal underlying geometry and therefore multiple layers of geometry were constructed. The 3D heart model was built to allow certain interaction functionality, which included the freedom to manipulate and separate parts of the geometry. Digital sculpting and photo manipulation techniques were utilised to achieve a photorealistic result.

The complete 3D heart model has been exported to Unity 3D which offers a flexible and affordable engine for the creation of real-time Virtual Environments (VE). The interface was developed in C# and embedded in the Unity 3D engine in order to complete the system.

3.2 Human–Computer Interaction Development

Contrary to the currently used traditional modalities our proposed user interface provides direct manipulation interaction enabling the user to adapt faster on the virtual environment. The capacity to interact freely and without physical constrains in the 3D space improves users' ability to interact with the 3D model and apply their own learning preferences. The user can safely experiment and investigate the 3D objects in the VR environment in real-time. The photorealistic 3D visualisations are also responding in real-time to the digital scene's light sources offering further immersion with the system as depicted in Fig. 1. The interface is further equipped with the capacity of selective transparencies for specific heart structures to enable the students to comprehend the complex heart anatomy relationships as presented in Fig. 2.

Notably the multiple iteration through the design and development of the prototype HCI has benefited substantially by the constant consultation and reviewing from the specialised medical practitioners. The gradual refinement of the system improved the accuracy of the interactions and added purposeful tools related to real dissection processes.

During these sessions the aim was to ensure true anatomical relations of the different parts of the heart. Special emphasis was placed upon the design of the conductive system of the heart (a group of specialised nodes and fibres that generate and conduct an electric impulse that causes the myocardium to contract), as the visualisation of the conductive system are not detailed on the cadaver. The labelling process of the heart structures was revised repeatedly in order to ensure

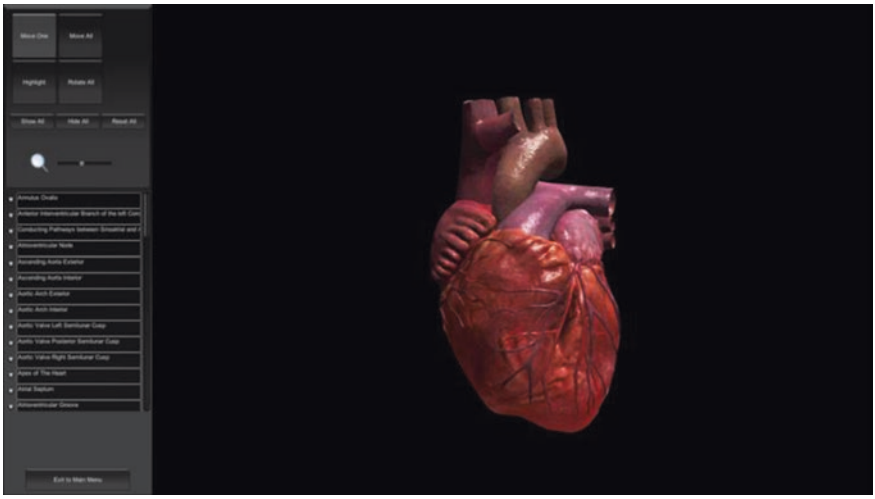


Fig. 1 The Anatomy interface in the VR heart anatomy system. Shows the wide space for interacting with the VR heart model, and the functionalities provided

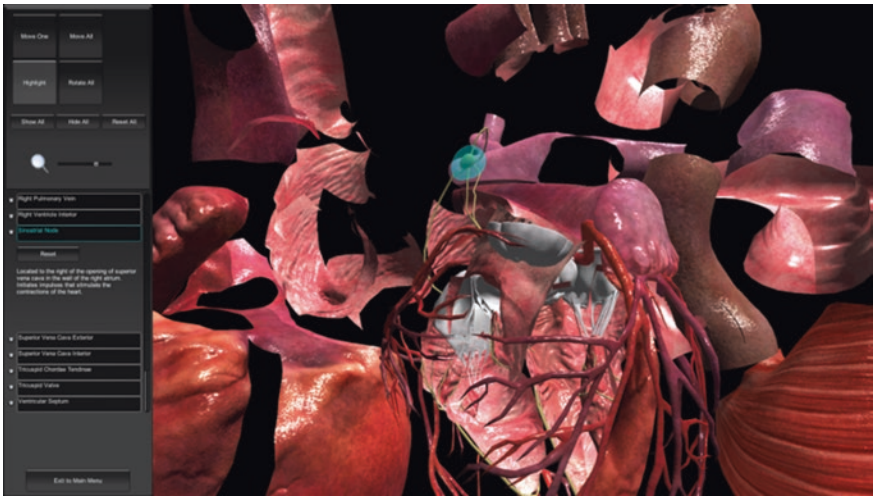


Fig. 2 Screenshot from the VR heart anatomy system shows the 'Move One' and 'Highlight' tool in use. The user dissected multiple structures from the VR heart model

precision and consistency. Finally, the position of the heart in the interface was set to the correct anatomical orientation.

Every element of the interface has designed with a view to improve the teaching engagement and knowledge acquisition in a timely manner. These principles are particularly important for a realistic medical training where performance anxiety is high due to the nature of the profession and the challenging situations. However in the VR

environment the student is allowed to fail safely, gaining from the experimentation and the learning experiences of failure with no risk to the patient.

The HCI system was presented in two main areas of the screen. The left side was occupied by a set of tools presented though icons forming the main interface, whilst the right and largest section of the screen was devoted to the 3D interactive heart model, forming the anatomy interface. The main interface aims to introduce the viewer to the system screens and facilitate the switching between information pages, and the anatomy-quiz which is linked with the 3D model. The anatomy interface aims to engage the user directly with a fully interactive 3D model of the heart. The user through interaction gains an in depth understanding of the structure of the heart, browsing through the anatomical structures by the process of dissection and manipulation to identify individual parts and their relations with others. In addition to visualising the heart in three dimensions, each structure is provided with a brief description of relevant anatomical information, which reinforces the learning experience.

The design of the anatomy interface demonstrated in Fig. 1 aimed to provide the viewer with enhanced visualisation of the 3D heart structures in addition to "easy to use" functionalities. Therefore the interaction between the user and the VR heart model is made by selecting any of the provided functionalities in the tool bar (on the left of the screen: dissect structures (Move one), restart the process (Reset all), move all the shown structures (Move all), mark structures (Highlight), rotating the structures (Rotate all), enlarging the structures (Zoom in), minimising the structures (Zoom out), show, the ability to make the structures either visible or invisible (Show all, Hide all), identify heart structures using a probe (structure list), and exit to main screen (Exit) as presented in Figs. 2 and 3. As the height of the heart is more than

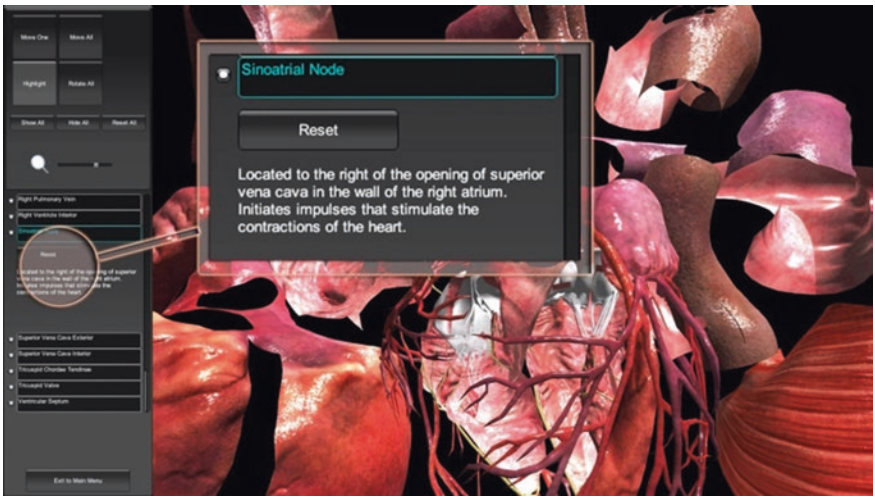


Fig. 3 Indicative screenshot shows the presented anatomical information for the selected label (Sinoatrial Node)

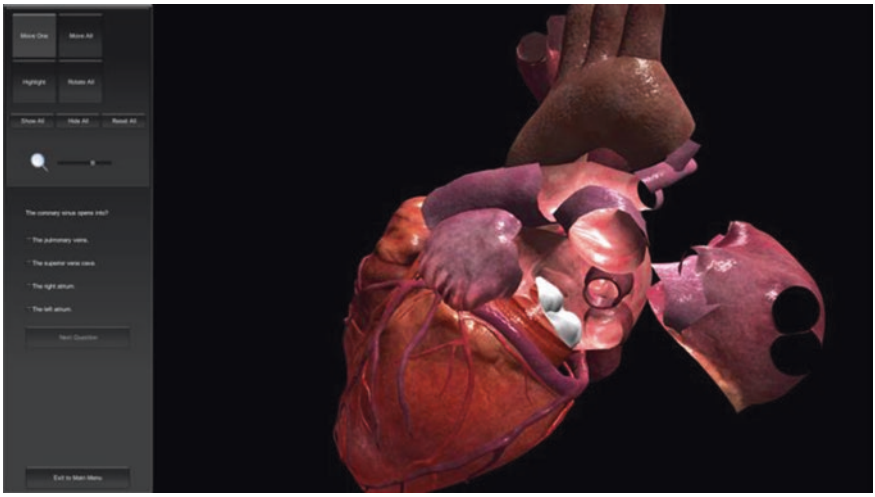


Fig. 4 Screenshot from the VR heart anatomy system, the user can strip away the various layers of the heart during the quiz

the width, the toolbar is placed on the left hand side of the screen. This provides a virtual dissecting area providing amplitude space in which the user can interact with the VR heart model as depicted in the screenshot of Fig. 4.

As stated above the main interface accommodates a section design as a knowledge Quiz. The latter is used as an assessment tool which could be utilised both by lecturers and students. In particular the lecturer can use the particular section to examine students. The system is designed in order to be fully customizable and contain an extensive number of questions linked to the 3D model. Additionally the students could use this part of the interface in order to perform a self-evaluation based on previous questions and past-papers. The current quiz was developed based on the MRCQ exams of the Royal College of Surgeons of Edinburgh in order to maintain validity and challenge the participants. The quiz consists of 25 questions added to assess students' knowledge based on the functionalities provided in the anatomy interface. The entire questions were Multiple Choice Questions (MCQs). The student answers by selecting one of the multiple choices. In the quiz interface the user is able to navigate through the model and utilise the provided tools except the structures' list as depicted in Fig. 5. The content of the questions focuses on assessing the ability to correctly identify anatomical parts of the heart, anatomical relations of different parts, and an overall understanding of the structure of this organ. The questions are suitable for exam purposes and can be used in that setting, aiding the examiner. The student can chose an answer from the multiple choice provided answers. By the completion of the examination the user can assess the outcome by revealing the correct responses. Colour-coding has also been employed in order to provide with a quick visual appraisal of the outcome. Detailed information regarding scoring and the correct answers is also provided.

Fig. 5 User interacting with the VR Heart Anatomy System, and responds to exam questions. *Left* picture depicts this interaction with 3D projection wall, and the *Right* picture presents the user interacting with the 3DTV

4 Prototype VR System Evaluation

For the evaluation of the prototype VR system we employed an evaluation process which was performed both in Europe and Middle East. The first evaluation provided a subjective feedback for the system which informed our work through minor amendments prior to the second evaluation in Jordan. Notably the proposed VR system was contrasted to the traditional teaching methods in both cases.

4.1 Evaluation Stage 1

The first stage of our evaluation involved a group of 10 Medical professionals in the United Kingdom (UK). The participants were from different medical backgrounds, such as consultant surgeons, cardiologists and general practitioners. The group had a mixed population of medical practitioners that acquired their medical degree in UK and in the Middle East, yet all had their Royal College Fellowship acquired in the UK, as fully qualified, practising consultants. The reason to select from both groups was to improve the usability and acceptability of the proposed VR system both in European and Middle East level. As such the final system could be used for teaching purposes in both regions.

On this first evaluation the system was appraised both for the HCI functionality as well as for the accuracy and realism of the 3D context. The users were asked to identify a sequence of anatomical components of the heart, and perform specific tasks. In turn the participants filled in a questionnaire that contains 10 questions using a 7-points Likert scale ranging from (1) Completely Disagree (2) Mostly Disagree (3) Slightly Disagree (4) Neutral (5) Slightly Agree (6) Mostly Agree to (7) Completely Agree.

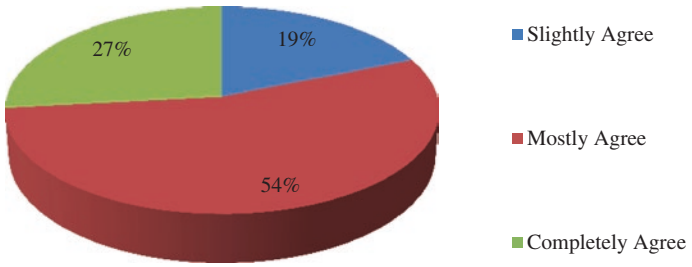


Fig. 6 Interface and VR Model optimisation results, all the answers were within 'Agree' scales

Overall, all the answers were within 'agree' scales. The summary of the results is illustrated in Fig. 6. In detail the 19 % of the answers were 'Slightly Agree', 54 % of the answers 'Mostly Agree' and 27 % of the answers were 'Completely Agree'. This means that doctors largely agreed on the provided system but they had some considerations regarding minor amendments that were suggested for the system prior to the students' cohort evaluation in the second stage. These suggestions were summarised in the following points:

- Highlight anatomical relations of the different parts of the heart.
- Improve Labelling for even smaller components and sections.
- Offer additional anatomical information in the structures' list.
- Provide visible pathways of the conductive system for the heart.

The aforementioned amendments were implemented prior to the second stage of our evaluation. Some additional suggestions were taken into consideration by the developing team, yet were covered through other parts of the interface and as such, it would have been superfluous to incorporate multiple interactions for the same interface elements. Potentially this multiplicity could have been useful for other types of interfaces (i.e. mobile phones), however for an educational system it was deemed necessary to maintain the simplicity of interactions [20, 21, 22]. Notably the users offered suggestions for a more detailed anatomical information per structure. However this has been considered during the development and as such the system already offers an 'Update' option for editing the information. This update may vary from one Faculty of Medicine to another. Therefore, this system has the major anatomical information that covers the heart anatomy, and the additional information was left up to the lecturers. Secondly the users' offered some ideas and suggestions for navigating and manipulating in the VR environment. These suggestions were conflicting variations of the same interface affected by personal preferences.

As such we maintained the simplicity of the system based on the more popular responses. In future work we will entertain the possibility to customise the interface within the limits that dictate an efficient functionality based on our multiple iterative developmental stages.

4.2 Evaluation Stage 2

The second stage of our evaluation involved a group of 40 third-year medical students beginning their fundamental skills in training in University of Jordan.

Forty students were randomly selected to participate in the study. The group was divided in two arms of 20 students. The first arm was trained through an hour long lecture in the traditional physical model prior to the examination. The second arm of 20 students received a hour long lecture through the VR heart anatomy system prior to their examination process. The examination entailed a set of 25 multiple choice questions based an adaptation of the MRCS previous examination papers. The students had 30 min to complete the test. The evaluation results were in favour of the VR system with better overall performance of the students.

The produced mean scores were compared for both groups. The quiz mean scores for the first group (traditional physical model group) was 13.05/25 while that of the second group (VR heart anatomy system) was 18.95/25 as presented in Table 1.

The comparison of the two groups' means of the examinations shows that there was difference in results between using traditional anatomy modalities versus using VR anatomy system. The primary outcome of this evaluation was the increment in the mean score quiz for the VR anatomy system as shown in Fig. 7.

Table 1 Quiz results for 2 groups (traditional physical model and VR heart anatomy system)

Groups	Number of students	Minimum quiz score	Maximum quiz score	Mean
1. Traditional physical model	20	7	20	13.05
2. VR heart anatomy system	20	12	25	18.95

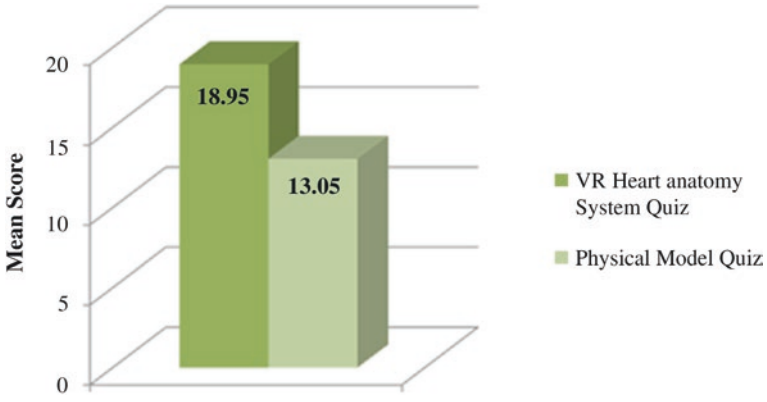


Fig. 7 Mean score for the physical model quiz and VR heart anatomy system

With this evaluation it have been demonstrated that a computer-based 3D anatomical model presented in interactive Virtual Reality environment enhances medical students' learning for heart anatomy. These results are in par with similar experiments that we have conducted previously for other medical simulation systems that we have developed for both anatomy and pathology training [2, 3, 8]. In all the comparative studies the VR systems have performed significantly better that the traditional teaching methods.

5 Discussion

The proposed VR medical training system was designed based in the hypothesis that the use of emerging technologies can improve the knowledge acquisition process in a demanding field such as medical training. Adhering to the above we designed and developed a prototype system that was exclusively produced to facilitate the teaching of a particular organ. In our prototype, the heart was identified as a suitable subject, due to the complexity of the internal and external structures, the related pulsating motion and the plurality of anatomical variations and pathologies.

It has to be noted that our previous experimentation on this field offered some valuable pathways regarding the decisions taken for improved efficiency of the interface design and the optimal 3D visualisation techniques [2, 3, 6]. The development process and iterative evaluations provided a valuable insight to the challenges related to the development of an accurate and fully functional system which could assist both lecturers and students.

The final prototype VR system has the capacity to offer real-time, 3D representations of the heart structure in an interactive VR environment. This offers simulation capacity that has the potential to enhance the heart anatomy course as the evaluation results suggest. In order to provide a complete teaching tool for the heart anatomy, it was imperative to produce photorealistic 3D model that includes all the major anatomical structures of the heart. Throughout the user trials for both evaluation stages, the users commented in favour of the realism of the 3D model which represented accurately the particular organ.

The embedded HCI functionalities have also received positive feedback as they enabled the user to interact through simple and timely actions in real-time with the 3D model. The replication of actions, such as dissection of the heart into layers, enhanced further the immersion and added the direct manipulation element which related to real-life activities of the users. Additionally the transparency of each part of the heart presented an unreal, yet very useful interaction which supported immensely the teaching process in the VR environment.

The anatomical information provided for each 3D structure, provided a concrete learning set of information that would link directly to the 3D model and highlight both the selected information and the correlated 3D section of the model. This element was highly appreciated by both professionals and trainees as it is demystifies

the spatial positioning and relation of each section to the neighbouring parts. This skill will be in turn transferable in the real-human body in contrast to traditional methods that presented partially or inaccurately this information through plastic models, 2D depictions or pre-dissected cadavers. Admittedly the professionally qualified medical practitioners recalled during their debriefing, after the first evaluation, the difficult transition between the student level medical training and the actual first year in a hospital training environment as Foundation Year doctors.

To provide the system users with a complete teaching aid for general heart anatomy structure, the system entails finally a knowledge evaluation system which retrieves questions from previous examination and past papers. This system concludes the training process either through guided teaching or self-directed studies. The system offers an analysis of the student answers and overall score which can be stored and used as a comparative background for future examinations. As the questions can be customised and edited the system can facilitate almost infinite number of examination papers.

Overall the proposed VR system has been developed with a view to be cost-effective, customisable, and offer accessible functionalities. As such we utilised off-the-shelf equipment that we have extensively tested in the Virtual Reality and Simulation (VRS Lab) laboratory in order to identify potential issues prior to deployment. From a software point of view we utilised a very common gaming engine, namely Unity 3D, which offers compatibility between platforms and ease of development and use. Consequently the aforementioned choices were approved by both user groups which tested the system effectively without hardware or software issues. The smooth operation of the developed system was favourably commented upon.

6 Conclusion

The work presented within this paper proposed the design and development for the proposed educational aid which addresses daily issues for medical students in learning anatomy. The developed system was based on users' requirements and followed closely the suggestions of specialised medical practitioners that have teaching experience in this field.

The system was developed to enhance medical education by utilising VR technology. Based on our previous research we developed a prototype VR system that could enhance and compliment contemporary teaching methods. Yet the proposed system offers customisation capacity, accessibility out-with laboratory constrains, repeatability of training sessions in contrast to cadaver training and safe experimentation. Evidently the proposed VR Anatomy Training System circumvents the limitations of the traditional methods as presented also through our evaluation.

In order to facilitate an appraisal of the system two evaluation studies have been performed. The first included only professional participants that provided subjective feedback and scored their preferences through a Likert scale. Their suggestion

have been used to fine tune the system. The second evaluation with 40 medical students presented a more realistic view of the performance with and without the proposed system. The derived results supported the VR system and offered positive feedback.

In a nutshell the proposed VR system was designed to be cost efficient and robust from a development point of view. It was also designed to encapsulate a large volume of information typically taught with time consuming and un-involving methods. In contrast, the proposed system capitalises on the technologically accustomed new generations of medical trainees that acquire knowledge significantly faster though various digital conduits. The hypothesis that an advance VR system could enhance drastically their performance and knowledge acquisition was supported through our initial evaluation.

Although the system offered promising results our future work will aim towards the development of a faster interface which could accommodate additional interaction tools. Furthermore we aim to perform further evaluation with a larger user sample in order to determine the users' intention to use such system through a Technology Acceptance Model (TAM). Finally we envisage to develop a complete gesture recognition system that will enable the user to operate the interface without the typical computing input constrains.

References

1. Oestergaard, J., Bjerrum, F., Maagaard, M., Winkel, P., Larsen, C.R., Ringsted, C., Soerensen, J.L.: Instructor feedback versus no instructor feedback on performance in a laparoscopic virtual reality simulator: a randomized educational trial. BMC Med. Educ. **1**, 7 (2012)
2. Charissis, V., Ward, B.M., Naef, M.: An enquiry into VR interface design for medical training: VR augmented anatomy tutorials for breast cancer. In: Proceedings of the: International Annual Symposium of IS&T/SPIE, The Engineering Reality of Virtual Reality, vol. 6804, pp. 19–28. San Jose (2008)
3. Ward, B.M., Charissis, V., Rowley, D., Anderson, P., Brady, L.: An evaluation of prototype VR medical training environment: applied surgical anatomy training for malignant breast disease. J. Stud. Health. Technol. Inform. **132**, 500–505 (2008)
4. Mantovani, F., Castelnuovo, G., Gaggioli, A., Riva, G.: Virtual reality training for health-care professionals. CyberPsychol. Behav **6**, 389–395 (2003)
5. Alfalah, S.F., Harisson, D.K., Charissis, V., Evans, D.: An investigation of multimodal interaction and 3d simulation environment for prototype healthcare system. in Journal of Enterprise Information Management (JEIM). N. Mustafee, and K. Katsaliaki, (eds.) vol. 26, pp 183–197 (2013). ISSN: 1741-0398
6. Falah, J., Harrison, DK., Wood, B., Evans, D.: The characterisation of an IT system to reduce the gap between information technology and medical education. In: International Conference on Manufacturing Research (ICMR), vol. 2, pp. 360–365 (2012)
7. Craig, A.B., Sherman, W.R., Will, J.D.: Developing virtual reality applications, pp. 145–189. Elsevier, Amsterdam (2009)
8. Sakellariou, S., Charissis, V., Grant, S., Turner, J., Kelly, D., Christomanos, C.: Virtual reality environment as knowledge enhancement tool for musculoskeletal pathology. In: Schumaker, R. (ed.) Human-Computer Interaction. Virtual and Mixed Reality. Lecture Notes in Computer Science, vol. 6774, pp. 54–63, Springer, Berlin (2011). ISBN 978-3-642-22023-4

9. Huang, H.M., Rauch, U., Liaw, S.S.: Investigating learners' attitudes toward virtual reality learning environments: based on a constructivist approach. Comput. Educ. **55**, 1171–1182 (2010)
10. Sugand, K., Abrahams, P., Khurana, A.: The anatomy of anatomy: a review for its modernization. Anat. Sci Educ. **3**, 83–93 (2010)
11. Snelling, J., Sahai, A., Ellis, H.: Attitudes of medical and dental students to dissection. Clin. Anat. **16**(2), 165–172 (2003). doi:10.1002/ca
12. Chien, C.H., Chen, C.H., Jeng, T.S.: An interactive augmented reality system for learning anatomy structure. In: Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 1, Hong Kong (2010)
13. Turney, B.W.: Anatomy in a modern medical curriculum. Ann. R. Coll. Surg. Engl. **89**, 104–107 (2007)
14. Schuwirth, L.W., van der Vleuten, C.P.: Medical education: challenges for educationalists. BMJ. Br. Med. J. **333**(7567), 544–546 (2006). doi:10.1136/bmj.38952.701875.94
15. McNulty, J.A., Sonntag, B., Sinacore, J.M.: Evaluation of computer aided instruction in a gross anatomy course: a six year study. Anat. Sci. Educ. **2**(1), 2–8 (2009)
16. Nicholson, D.T., Chalk, C., Funnell, W.R.J., Daniel, S.J.: Can virtual reality improve anatomy education? A randomised controlled study of a computer generated three dimensional anatomical ear model. Med. Educ. **40**, 1081–1087 (2006)
17. Vozenilek, J., Huff, J.S., Reznek, M., Gordon, J.A.: See one, do one, teach one: advanced technology in medical education. Acad. Emerg. Med. **11**(11), 1149–1154 (2004). doi:10.1197/j.aem.2004.08.003
18. Onyesolu, M.: Virtual reality laboratories: an ideal solution to the problems facing laboratory setup and management. In: Proceedings of the World Congress on Engineering and Computer Science, San Francisco, vol. 1 (2009). ISBN: 978-988-17012-6-8
19. Hennessy, R.J., McLearie, S., Kinsella, A., Waddington, J.L.: Facial surface analysis by 3D laser scanning and geometric morphometrics in relation to sexual dimorphism in cerebral–craniofacial morphogenesis and cognitive function. J. Anat. **207**(3), 283–295 (2005)
20. Brenton, H., Hernandez, J., Bello, F., Strutton, P., Purkayastha, S., Firth, T., Darzi, A.: Using multimedia and Web3D to enhance anatomy teaching. Comput. Educ. **49**, 32–53 (2007)
21. Falah, J., Harrison, D.K., Charissis, V., Wood, BM., (2013) The characterisation of a virtual reality system to improve the quality and to reduce the gap between information technology and medical education. In: Virtual, Augmented and Mixed Reality, Systems and Applications, vol. 2, pp. 122–131. Springer, Berlin (2013)
22. Petersson, H., Sinkvist, D., Wang, C., Smedby, Ö.: Web-based interactive 3D visualization as a tool for improved anatomy learning. Anat. Sci. Educ. **2**, 61–68 (2009)

Compression of ECG Signal Using Hybrid Technique

K.S. Surekha and B.P. Patil

Abstract The new method proposed here consists of a combination of two transforms. The hybrid technique consists of Discrete Cosine Transform (DCT) and 1D or 2D discrete wavelet Transform (DWT). The method is suitable for compression of ECG signals. In this research, different records of MIT–BIH data base are used. The performance measure of the technique is done with the help of Compression Ratio (CR) and Percent Root Mean Square Difference (PRD). The threshold based technique is used to achieve better CR. The threshold value is selected based on the R peak. The QRS complex is detected in order to select the R peak. The threshold level is selected as 1, 0.5, and 0.1 % of the R peak. Further improvement in the CR is achieved by the DWT decomposition method. The level of decomposition is carefully selected to achieve improved CR.

Keywords ECG · Compression · R peak · CR · PRD

1 Introduction

An Electrocardiogram (ECG) signal is used as an important signal by the doctors for diagnosis purpose. The normal ECG signal is as shown in the Fig. 1.

The ECG signal is an AC signal with a voltage level of 5 mv of peak to peak. The bandwidth of ECG signal is 0.05–100 Hz. The various methods of electrocardiography are available. In the first method [1, 4], 12 different potential differences are recorded from the body surface. The method is known as standard clinical ECG. The potential differences recorded are called as ECG leads. In the second method, body surface potentials are used as inputs to a three dimensional vector model of

K.S. Surekha (✉) · B.P. Patil
Department of Electronics and Telecommunication,
Army Institute of Technology, Dighi, Pune, India
e-mail: surekhaks@yahoo.com

B.P. Patil
e-mail: bp_patil@rediffmail.com

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_24

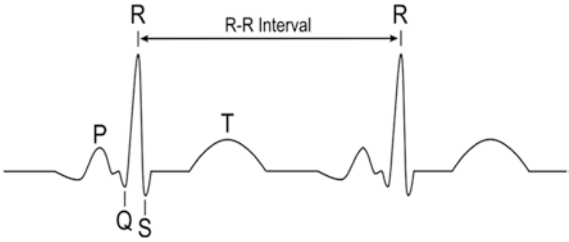


Fig. 1 ECG signal

cardiac excitation. The method is used to get a graphical view of the excitation of the heart. That is called as vector cardiogram (VCG). The third method consists of using only one or two ECG leads. Long monitoring of ECG signal is done in this method. The method is preferred in intensive care unit or in ambulatory. The sampling rate of ECG signal is 200–500 samples with 8–12 bit resolution.

The QRS complex detection in ECG plays an important role. Different methods are used for the detection of QRS complex. The signal is reduced into a set of predefined tokens. Certain shapes of the ECG waveform are represented by these tokens. Pan and Tomkins developed a real time QRS detection algorithm (1985). The signal is passed through band pass filter to attenuate noise. The band pass filter is implemented as a combination of high pass and low pass filter [1]. The remaining stages are differentiator, squaring and finally through time averaging of the signal.

The ambulatory R peak detection proposed [2] is optimized to reduce the computational complexity. A new method proposed [3] is based on the empirical mode decomposition and adaptive threshold technique. The method can detect R peak for various ECG shapes.

In the proposed method, filtering and thresholding are used for the detection of QRS complex. After detection of the QRS peak, the difference code is calculated. The R peak is detected, if the difference code detects a change from $+1$ to -1.

2 Types of Compression

The ECG signal which needs to be stored and transmitted, in a computerized medical signal processing system, the reduction of storage space is essential. This must also preserve the clinical content for signal reconstruction. After reconstruction, the signal is used as an important tool for diagnosis purpose. Various compression techniques are available for ECG signals [5–9].

- Direct Method
- Transform based method
- Parameter Extraction method

Direct methods are simpler. Some of the methods are Coordinate reduction time encoding system (CORTES), FAN, TURNING POINT (TP), Amplitude zone time epoch coding (AZTEC).

In Transform based method, the common transforms used are DCT, DWT, and DFT etc. Wavelet transform is used because of its localized and non-stationary property. In Parameter Extraction Methods a set of parameters extracted from the original signal. These parameters are used in the reconstruction process. The direct methods are less complex than transform methods. In the transform method, Energy of signal is represented by less number of samples [5, 7, 9].

3 Work Carried Out in the Field

Most of the work is carried out using transform based techniques. The transform based techniques are KL transform, wavelet transform [8], Burrows wheeler transformation [10] wavelet transform using SPIHT algorithm [6, 8], DCT [11], 2D wavelet transform [7, 12] etc. The important point in most of the compression techniques is the better compression ratio.

The researcher has presented a wavelet [6] based compression based on the set partitioning in hierarchical trees (SPIHT) coding algorithm. The results show the high efficiency.

A new method [7] using 2-D wavelet transform is presented. The method utilizes the fact that 2-D wavelet show redundancy between adjacent samples and between adjacent beats. The methodology uses four steps. The method starts with converting 1-D ECG signal into 2_D array, preprocessing. Then DWT is applied followed by thresholding and RLC. The researcher has claimed the lower calculation complexity in comparison with other methods.

The researcher [8] has proposed a wavelet ECG data codec based on the Set Partitioning in Hierarchical Trees compression algorithm. The method has achieved small percent root mean square difference (PRD) and high compression ratio with low implementation complexity. Compression Ratio achieved is 48:1.

Burrows Wheeler Transformation [10] method used a combination of move-to-front coder with the Huffman coder. The data used are of small duration. The researcher has achieved a Compression Ratio of 2.7247. The method is useful for the data transfer in an ECG event recorder.

The researcher [12] has used 2-D approach. 2D approach is based on the fact ECG signals show redundancy between adjacent beats and between adjacent samples. The method achieves high compression ratio with low distortion and low computational complexity in comparison with other methods.

4 Discrete Wavelet Transform and Discrete Cosine Transform

The different types of wavelet transform are continuous wavelet transform; a wavelet series expansion and a discrete wavelet transform (DWT) [6, 14]. The discrete wavelet transform is defined as

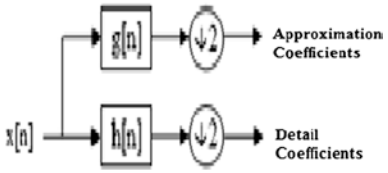


Fig. 2 Filter analysis using DWT

$$f(t) = \sum C_{m,n}\psi_{m,n(t)}$$

2D wavelet exhibits redundancy between adjacent beats and between adjacent samples [12].

The DWT of a signal x is calculated by filtering the signal through a series of filters as shown in Fig. 2. Initially the samples are passed through a low pass filter with impulse response 'g'. The signal is also decomposed simultaneously using a high-pass filter with impulse response 'h'. The outputs of the High pass filter gives detail coefficients and the outputs of Low pass filter gives approximation coefficients. Half the frequencies of the signal have been removed and half the samples can be discarded according to Nyquist's rule. The filter outputs are also sub-sampled by 2.

The DCT of an input sequence x is given by

$$y(k) = w(k)\sum_{n=1}^{N} x(n)\cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right)$$

$$k = 1, \ldots N$$

$$w(k) = \begin{cases} \frac{1}{\sqrt{\mathrm{N}}}, & \mathrm{k} = 1 \\ \sqrt{\frac{2}{\mathrm{N}}}, & 2 \leq \mathrm{k} \leq \mathrm{N} \end{cases} \tag{1}$$

The inverse discrete cosine transform is given by

$$x(n) = \sum_{k=1}^{N} w(k)y(k)x(n)\cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right), \tag{2}$$

$$n = 1, \ldots N$$

5 The Performance Measure of Compression Technique

Percent Root mean square Difference (PRD) and compression ratio (CR) [16, 17]. The distortion between the original and the reconstructed signal is known as the PRD.

$$\mathrm{PRD} = \sqrt{\left[\frac{\sum_{n=0}^{N-1}[x_0(n) - x_r(n)]^2}{\sum_{n=0}^{N-1}\left[x_o^2(n)\right]}\right]} 100\ \% \tag{3}$$

where, x_0 indicates the original data, x_r indicates the reconstructed data, and N represents the number of samples.

The Compression Ratio (CR) is defined as the ratio of number of bits in the original signal to the number of bits in the compressed signal

$$CR = \frac{\text{Number of bits in the original signal}}{\text{Number of bits in the compressed signal}} \tag{4}$$

6 The Proposed Method

The different components of ECG signals are P, T and QRS complex. The ECG signal used is taken from MIT–BIH data base [13]. The software is developed using MATLAB tool. The proposed block diagram is shown in Fig. 3.

The procedure starts with the preprocessing stage as the signal quality may degrade due to noise and distortion. The first stage in this process is high pass filter. This filter removes the base line wander noise. The power line interference is removed using stop band filter. The extraction of QRS wave starts with the Band Pass filter [15]. After this stage, threshold based technique is used for extracting QRS wave. Once the QRS complex is extracted, the R-peak is detected.

The signal is passed through two stages of transformation. In the first stage, DCT is used. The second stage consists of wavelet transformation. The 1-D and 2-D wavelet transforms are used in the second stage of transformation. The Wavelet transform is further decomposed to higher levels. This achieves improved CR. The level of decomposition is carefully selected. The decomposition methodology is applied to both 1-D and 2-D wavelet transform.

The procedure used in the compression process:

- After reading the signal from MIT–BIH data base, do the preprocessing
- Extract the QRS complex
- Detect the R peak
- Pass the preprocessed signal through DCT
- Threshold based technique for removing extra coefficients
- Reconstruct the signal using IDCT
- Pass the reconstructed signal through DWT

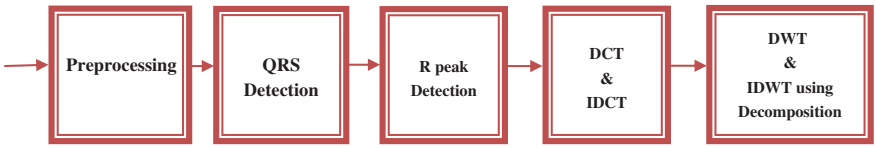


Fig. 3 The block diagram

- Pass the signal through various levels of DWT decomposition
- Reconstruct the signal using IDWT.

7 Results

The MIT-BIH data base is used for the testing purpose. The sampling frequency used here is 330 Hz. The ECG signal used is .dat signal. The total number of samples used is 256 and 512. After the QRS detection, the R peak is calculated. The threshold is calculated as percentage value based on the R peak.

The original ECG signal with 256 samples used for testing purpose is shown in Fig. 4. And the signal which is obtained after the removal of baseline wander noise using High pass filter is also shown in Fig. 4. The signal also passes through other stages of pre-processing. The QRS complex is extracted from the preprocessed ECG signal. The R peak is extracted from the QRS complex.

The signal then passes through the compression stages. The reconstructed signal after the IDCT is shown in Fig. 5.

The reconstructed signal after the IDCT stage passes through DWT and IDWT. The threshold based technique is used in the DCT and DWT stages. The reconstructed signal after IDWT is shown in Fig. 6.

The results are tabulated for the hybrid stage. Tables 1 and 2 show the results for DCT and 1-D DWT hybrid unit using 256 and 512 samples respectively (Table 1).

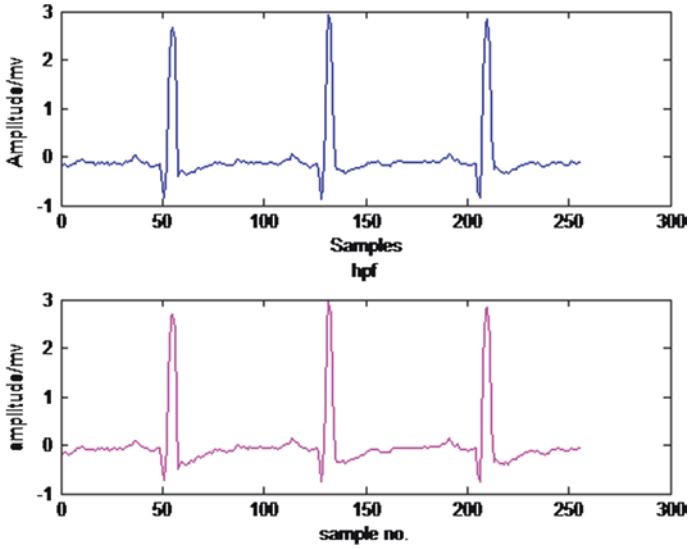


Fig. 4 The input signal and high pass filtering

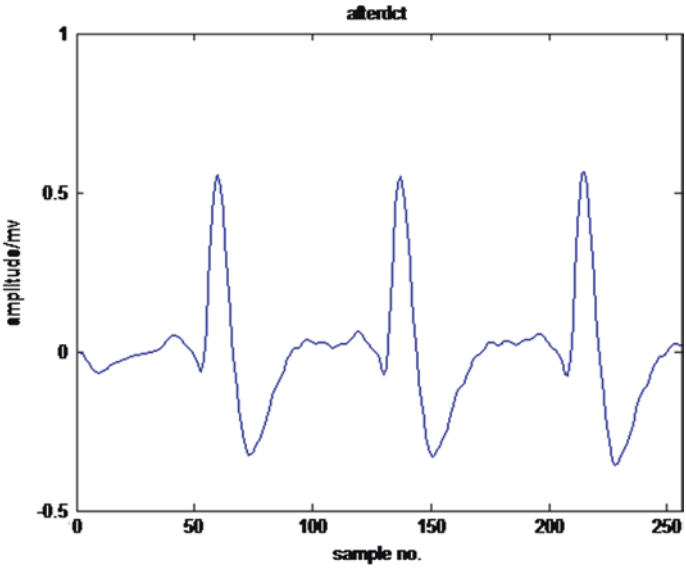


Fig. 5 The reconstructed signal after IDCT

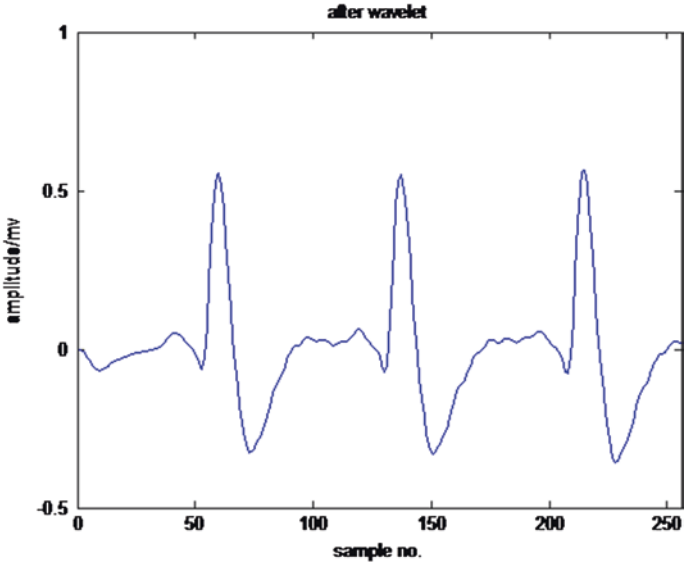


Fig. 6 The reconstructed signal after IDWT

Similarly Tables 3 and 4 show the results for DCT and 2-D DWT hybrid stage for 256 and 512 samples respectively.

Table 1 Hybrid transform technique using 1_D DWT for 256 samples for Db1 wavelet and Sym3 wavelet

% Threshold	Type of wavelet	CR		PRD	
		DCT	DWT	DCT	DWT
0.01 % of R peak	Db1	35.54	0.78	0.01	0.01
	Sym3	35.54	4.68	0.01	0.01
0.1 % of R peak	Db1	58.2	3.9	0.14	0.16
	Sym3	58.2	25.39	0.14	0.21
0.5 % of R peak	Db1	72.65	35.15	1.3	1.88
	Sym3	72.65	59.37	1.3	1.63
1 % of R peak	Db1	84.37	68.75	7.03	9.8
	Sym3	84.37	80.46	7.03	9.16

Table 2 Hybrid transform technique using 1-D DWT for 512 samples for Db1 wavelet & Sym3 wavelet

% Threshold	Type of wavelet	CR		PRD	
		DCT	DWT	DCT	DWT
0.01 % of R peak	Db1	37.5	0.39	0.02	0.02
	Sym3	37.5	3.12	0.02	0.02
0.1 % of R peak	Db1	59.17	4.29	0.19	0.20
	Sym3	59.17	20.11	0.19	0.23
0.5 % of R peak	Db1	74.21	34.76	1.66	2.21
	Sym3	74.21	59.57	1.66	2.04
1 % of R peak	Db1	89.64	69.72	10.13	12.65
	Sym3	89.64	78.9	10.13	11.86

Table 3 Hybrid transform technique using 2-D DWT for 256 samples

% Threshold	Type of wavelet	CR		PRD	
		DCT	DWT	DCT	DWT
0.1 % of R peak	Db1	58.20	0.78	0.14	0.14
	Sym3	58	0	0.14	0.14
0.5 % of R peak	Db1	68.75	10.93	0.76	0.93
	Sym3	68.75	4.68	0.76	0.77
1 % of R peak	Db1	84	51.56	7.03	8.96
	Sym3	84.37	34.76	7.0	8.4

Table 4 Hybrid Transform technique using 2-D DWT for 512 samples

% Threshold	Type of wavelet	CR		PRD	
		DCT	DWT	DCT	DWT
0.1 % of R peak	Db1	58.2	0.71	0.14	0.14
	Sym3	59	0	0.19	0.19
0. 5 % of R peak	Db1	69.92	7.42	0.87	0.97
	Sym3	69.92	2.34	0.87	0.88
1 % of R peak	Db1	89	51.17	10.13	11.81
	Sym3	89.64	26.17	10.13	11.44

The results are also tabulated using different levels of DWT decomposition for both 1-D and 2-D wavelets. This gives further improvement in the CR and are shown in Tables 5 and 6.

Table 5 Hybrid Transform technique using 1-D DWT & different levels of decomposition for 256 samples

% Threshold	Type of wavelet	CR		PRD	
		DCT	DWT	DCT	DWT
0.1 % of R peak	*Level 3 DWT decomposition*				
	Db1	58.2	4.29	0.14	0.16
	Sym3	58.2	26.95	0.14	0.24
	Level 4 DWT decomposition				
	Db1	58.2	5.27	0.14	0.15
	Sym3	58.2	25	0.14	0.22
0. 5 % of R peak	*Level 3 DWT decomposition*				
	Db1	72.65	39.06	1.3	2.17
	Sym3	72.65	62.5	1.3	1.81
	Level 4 DWT decomposition				
	Db1	72.65	34.76	1.3	1.9
	Sym3	72.65	59.37	1.3	1.69
1 % of R peak	*Level 3 DWT decomposition*				
	Db1	84.37	74.21	7.03	11.50
	Sym3	84.37	83.98	7.03	8.65

Table 6 Hybrid transform technique using 2-D DWT and different levels of decomposition for 256 samples

% Threshold	Type of wavelet	CR		PRD	
		DCT	DWT	DCT	DWT
0.1 % of R peak	*Level 3 DWT decomposition*				
	Db1	58.2	0.78	0.14	0.14
	Sym3	58.2	3.51	0.14	0.14
	Level 6 DWT decomposition				
	Db1	58.2	5.46	0.14	0.29
	Sym3	58.2	57.16	0.14	0.14
0. 5 % of R peak	*Level 3 DWT decomposition*				
	Db1	72.65	24.2	1.3	2.01
	Sym3	72.65	16.01	1.3	1.34
	Level 6 DWT decomposition				
	Db1	72.65	39.84	1.3	3.71
	Sym3	72.65	60.95	1.3	1.31
1 % of R peak	*Level 3 DWT decomposition*				
	Db1	84.37	60.93	7.03	10.61
	Sym3	84.37	49.21	7.03	7.51
	Level 6 DWT decomposition				
	Db1	84.37	75	7.03	21.67
	Sym3	84.37	67.18	7.03	7.18

Table 7 Comparison chart

Method	CR	PRD (%)
Wavelet Compression using SPIHT [6]	21.4	3.1
2D Wavelet Transform coefficients and run length coding [7]	8	0.71
Compression using Wavelet Transform and SPIHT [8]	45	1.06
Burrows Wheeler Tr. and Move to front coder [10]	2.72	*
Beta Wavelet based ECG signal compression using lossless encoding [14]	6.06	1.11
Compression based on Discrete Wavelet transform and QRS complex estimation [15]	25.15	0.7
DWT [16]	15.6	2.81
Compression based on Mother wavelet parameterization and best threshold level selection [17]	23.1	1.60
Proposed method (DCT/1D DWT)	74.21/59.57	1.66/2.04

*not mentioned

A comparison chart is shown in Table 7. The proposed method is compared with the methods developed by the researchers. The various methods listed are wavelet using SPIHT [6], 2-D Wavelet Transform Transform and run length coding [7], Burrows Wheeler and move to front coder [10], Beta Wavelet based method [14], DWT and QRS complex [15], DWT [16], and Mother wavelet parameterization. The proposed method shows better results than the existing methods (Table 7).

The performance measure of the compression technique is done with the help of CR and PRD. The graph showing the CR versus % threshold is shown in Fig. 7. The figure shows the graph for both 1-D and 2-D DWT.

8 Conclusion

The proposed method of compression is tested for both 1-D and 2-D wavelet transform. The results are tabulated above. It is clear from the result table that 1-D DWT shows better results than 2-D DWT. The threshold value improves the CR. As we increase the threshold, the CR and PRD also increases. The increase in PRD beyond the value 10 is not desirable. This may deteriorate the clinically significant details of signal. Hence the value of threshold must be selected carefully. Further improvement in the CR is achieved by selecting higher levels of decomposition in both 1-D and 2-D wavelet transform. The improvement in CR is achieved in the 3rd level of decomposition in case of 1-D DWT and in the 6th level of decomposition for 2_D DWT. Further decomposition of DWT reduces the CR value. The Symlet DWT shows better results than Db1. This clearly shows that the overall improvemet in CR is achieved by suitable threshold value and DWT decomposition. Further improvement in CR is possible by suitable encoding techniques.

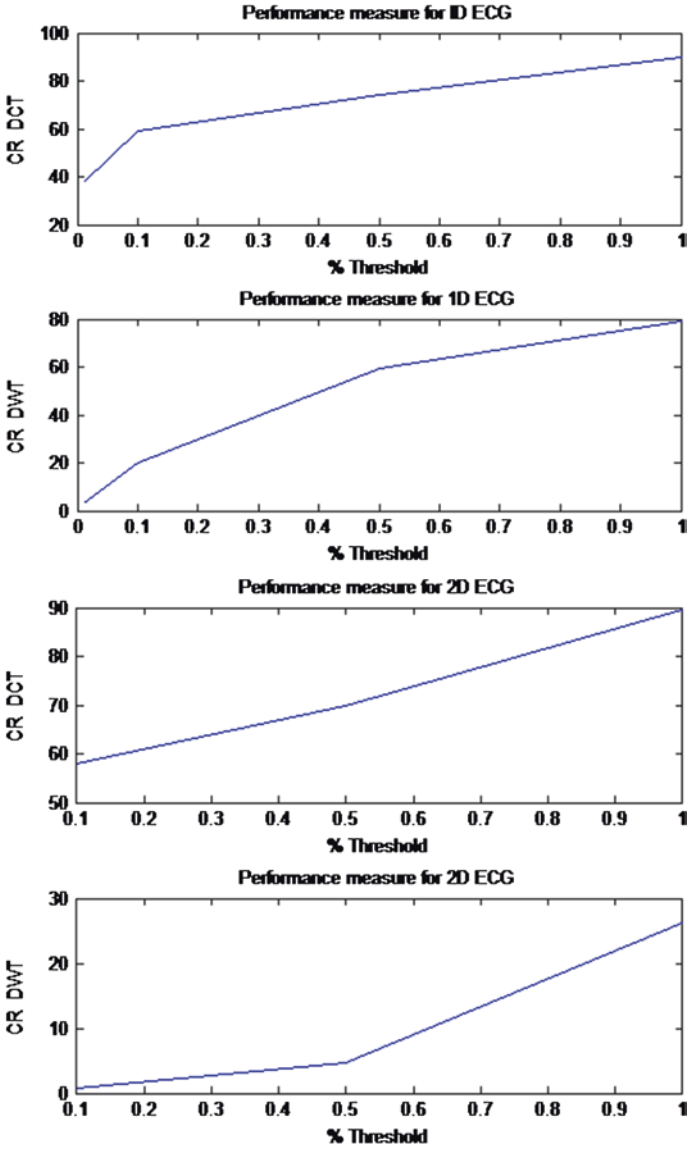


Fig. 7 The Graph of CR versus % threshold for 1D and 2D DWT

Acknowledgments The authors would like to thank the Principal, Head of Department & PhD coordinator Sinhgad college of Engineering, for their support & guidance to publish this paper. Also we would like to thank Director and Principal, AIT for the constant support and guidance to publish this research work.

References

1. Tomkins, W.J.: Biomedical Digital Signal Processing, 1st edn. PHI, USA
2. Rooijakkers M.J., et.al.: Low complexity R peak detection in ECG signals: a preliminary step towards fetal monitoring. In: IEEE Annual International Conference of the Engineering in Medicine and Biology society, pp. 1761–1764 (2011)
3. Tang J-h. et.al.: An algorithm of R-peak detection in ECG based on Empirical mode decomposition. In: IEEE Fourth International Conference on Natural Computation, vol. 5, pp. 624–627 (2008)
4. Khandpur, R.S: Handbook of Biomedical Instrumentation, 6th edn, TMH, New York
5. Jalaleddine, S.S., Hutchens, C., Strattan, R., Coberly, W.: ECG data compression techniques—a unified approach. IEEE Trans. Biomed. Eng. **37**, 329–343 (1990)
6. Pooyan, M., Taheri, A., Moazami-Goudarzi, M., Saboori, I.: Wavelet compression of ECG signals using SPIHT algorithm. Int. J. Inf. Commun. Eng. **1**(4), 219–225 (2005)
7. Mohammadpour, T.I., Mollaei, M.R.K.: ECG compression with thresholding of 2D—wavelet transform coefficients and run length coding. Euro. J. Sci. Res., ISSN 1450 **27**(2), 248–257 (2009)
8. Ktata, S., Ouni, K., Ellouze, N.: A novel compression algorithm for electrocardiogram signals based on wavelet transform and SPIHT. Int J Signal Proc. **5**(4), 253–258 (2009)
9. Moazami-Goudarzi, M., Moradi, M.H.: Electrocardiogram signal compression using multi wavelet transform. Trans. Eng. Comput. Technol. **6**, 332–336 (2005)
10. Dakua1, S.P., Sahambi, J.S.: Lossless ECG compression for event recorder based on burrows-wheeler transformation and move-to-front coder. Int. J. Recent Trends Eng. **1**(3), 120–123 (2009)
11. Aggarwal, V., Patterh, M.S.: Quality controlled ECG compression using discrete cosine transform (DCT) and laplacian pyramid (LP), Multimedia Signal Processing and Communication Technologies, IMPACT 09, 12–15 (2009)
12. Abu-Zahhad, M., et al.: An efficient technique for compressing ECG signals using QRS detection, estimation, and 2D DWT coefficients thresholding (2012). doi:10.1155/2012/742786, Article ID742786, Hindawi Publishing Corporation (Research Article)
13. http://ecg.mit.edu/
14. Kumar, R. et.al: Beta wavelet based ECG signal compression using lossless encoding with modified thresholding. Comput. Electr. Eng. J. **39**(1): 130–140
15. Zahhad, M.A., et al.: ECG signal compression technique based on discrete wavelet transform and QRS complex estimation. Sig. Process. Int. J. (SPIJ) **4**(2), 138–160 (2010)
16. Zahhad, M.A. et.al.: Compression of ECG signals based on DWT and exploiting the correlation between ECG signal samples. Int. J. Commun. Netw. Syst. Sci. (Scientific Research), **1**, 53–70 (2014)
17. Abo-Zahhad, M., et al.: A new algorithm for the compression of ECG signals based on mother wavelet parametrization and best threshold level selection. Dig. Sig. Process. J. **23**(3), 1002–1011 (2013)

Performance Analysis of MATLAB Parallel Computing Approaches to Implement Genetic Algorithm for Image Compression

Omaima N. Ahmad AL-Allaf

Abstract This chapter presents how to use parallel computing approaches from MATLAB Parallel Computing Toolbox to implement genetic algorithm for fractal image compression. These approaches are: ParFor, CoDistributor and Parallel Cluster. This is done to decrease processing time as possible as and maintaining reconstructed image quality. Many experiments were executed with comparisons between the three approaches. The experimental results showed that decreasing the GA population size and increasing number of workers used for the three parallel computing approaches can reduce the compression time. Best results obtained from implementing parallel approaches with 6 workers and 150 population size. The execution speed reached 4, CR reached 90.97 % and PSNR reached 34.98 db. At the same time, best results obtained from Parallel Cluster approach and then from CoDistributor approach.

Keywords Parallel computing approaches · Genetic algorithm (GA) · Fractal image compression (FIC) · Parallel cluster · ParFor · CoDistributor · Single program multiple data (SPMD)

1 Introduction

Image compression plays an important role in many research fields and can be regarded as one of the most important topics that were discussed in the literature studies. Fractal image compression (FIC) is a new compression techniques recently developed to compress images. From the literature, FIC can achieve high

O.N.A. AL-Allaf (✉)
Faculty of Sciences and Information Technology, AL-Zaytoonah University of Jordan,
P.O. Box 130, Amman 11733, Jordan
e-mail: omaimaalallaf@zuj.edu.jo

K. Arai et al. (eds.), *Intelligent Systems in Science and Information 2014*,
Studies in Computational Intelligence 591, DOI 10.1007/978-3-319-14654-6_25

reconstructed image quality, high compression ratio (CR), and fast decompression. At the same time, FIC requires huge time for executing large number of computations to find best matched domain block during compression process. Whereas, the FIC decompression process is fast [1, 2].

Many literatures were discussed FIC to reduce time and increase CR [2–7]. And different algorithms has been proposed and discussed to implement FIC based on partitioned iterated function systems. Each of these algorithms follows the same steps but the difference is in the way of partitioning images or in the metric used to compare domain and range blocks [8].

Genetic algorithm (GA) can be regarded as is a search and optimization algorithm that represents the biological evolutionary principles and chromosomes of natural genetics. GA depends on population that includes a set of chromosomes where each chromosome is a string of ones and zeros. A new population can be generated from the previous population by applying selection, crossover and mutation processes. This process is repeated until either reaching maximum number of generations or finding optimal solution [9, 10].

Many researches in the literature [11–17] were adopted GA for implementing FIC to reduce compression time. Most of these researches were achieved bad CR and peak signal to noise ratio (PSNR) and sometimes required large number of computations. Whereas, few literature researches [18–20] were implemented parallel computing to solve different problems to reduce computation time as possible as. Whereas other researches [21–24] were implemented parallel computing for FIC. Finally, very few researches [25, 26] were implemented GA for FIC using parallel computing. Therefore, in this chapter, parallel computing approaches were discussed to implement GA for FIC.

MATLAB has the ability to control programming environment. MATLAB can automates and make you send jobs to a remote computer without the difficulty of logging in, running program and take results, and transferring files [27, 28]. We can solve computational and data-intensive problems using MATLAB and Simulink on multiprocessor computers using parallel computing toolbox and MATLAB distributed computing server software. MATLAB Parallel computing constructs: parallel for-loops (ParFor), distributed arrays, parallel numerical algorithms, and message-passing functions to implement task parallel and data parallel algorithms at high level without programming for specific hardware and network architectures [27].

For this reason, three MATLAB parallel computing approaches were discussed and executed (ParFor, CoDistributor, and Parallel Cluster) in this chapter for implementing GA to compress images using FIC. This is done to achieve high CR and PSNR and also reduce the compression time as possible as. Experimental results showed good values of CR, PSNR and compression time. This chapter described also, the main principles of GA, FIC, GA for FIC and many of the MATLAB parallel computing approaches.

2 Genetic Algorithms Based Parallel Computing

Traditional Genetic algorithms can be summarized by the following steps [9, 10]:

1. Initialization: to generate first population of chromosomes that represents problem solutions. Each chromosome is represented as a string with zeros and ones.
2. Determine: Fitness function; number of generations; crossover and mutation probabilities.
3. Find corresponding values of created chromosomes in population using mapping rule. The fitness function is calculated for each chromosome using these values.
4. Selection: Each chromosome is evaluated and assigned a probability according to its fitness value.
5. Each pairs of chromosomes can be selected for reproduction.
6. Crossover (pc): each two selected chromosomes are combined to form new two chromosomes.
7. Mutation (pm): flip a single bit (1–0 or 0–1) in each selected chromosome.
8. Repeat selection, crossover and mutation until getting chromosome with minimum fitness function [9, 10].

The parallel computing can improve the calculations speed of GA processes (selection, crossover and mutation) because these processes will be executed in parallel way. The parallel computing depends on dividing complex population into many small populations and distributing them to be executed on connected parallel computers. Four types of parallel computing to implement GA: global parallel genetic algorithm (PGA); coarse-grained PGA; fine-grained PGA; hybrid PGA. They can be implemented using MATLAB. The global PGA is implemented using master-slaver programming on computers. This algorithm holds one population. The selection, crossover and mutation processes are applied to each chromosome in this population. The selection operation is executed globally on whole population (on master device) rather than locally (a sub-population on slave device). GA can be implement using one of the two parallel computing approaches: execute GA processes including fitness function in parallel; or parallelize the calculation of individual's fitness [29].

3 Fractal Image Compression Algorithm

Now a days fractal image compression, utilized in many applications and research fields to compress the image. Different algorithms has been proposed to implement the fractal image compression, based on partitioned iterated function systems, the difference is generally in the way of partitioning images, or in the metric used to compare domain and range blocks, but each algorithm follow the same steps[MM]. FIC is based on iterated function system (IFS). FIC includes the following steps [11, 12, 30, 31]:

1. An image (I) is divided into non-overlapping M (B × B) range blocks and into N arbitrarily located (2B × 2B) domain blocks. Range blocks are represented by R_i ($1 \leq i \leq M$). Domain blocks are represented by D_j ($1 \leq j \leq N$).
2. Best matched domain D_k ($1 \leq K \leq N$) and contractive affine transformation T_{ik} are calculated for each range block Ri to satisfy Eq. 1:

$$d(R_i, T_{ik}(D_k)) = \min.d(R_i, T_{ij}(D_j)) \tag{1}$$

where:

T_{ik}: contractive affine transformation from block D_j to block R_i.
$d(Ri; T_{ik}(Dj))$: distortion measure is mean square error (MSE) between range block R_i and contractive domain block $T_{ik}(Dj)$.

3. Contractive affine transformation T_{ik} is composed of two mappings Φ_j and Θ_{ij} as Eq. 2:

$$T_{ij} = \theta_{ij} \circ \phi_j \tag{2}$$

where,
Φ_j: Transformation of domain block size to same size as range block.

4. Dj is divided into non overlapping (2 × 2) unit blocks. Each pixel of transformed block $\Phi_j(D_j)$ is an average value of 4 pixels in each unit block in Dj.
5. Θ_{ij} includes transformation block $\Theta_{ij}(D_j)$ by one of 8 transformations (isometries): rotation around center of block $\Theta_{ij}(Dj)$ by (0°, 90°, 180°, 270°) and each rotation after orthogonal reflection about mid-vertical axis of block $\Theta_{ij}(D_j)$.
6. Transformation pij of pixel values of block obtained by first step as Eq. 3.

$$P_{ij}(v) = a_{ij}v + g_{ij} \tag{3}$$

where,

v: pixel value of block obtained by first step.
a_{ij} (scaling coefficient) and g_{ij} (offset):computed by least square analysis of pixel values of block R_i and block obtained by first step.

4 FIC Using Genetic Algorithm Based Parallel Computing

GA is used for FIC to reduce computation time instead of using traditional FIC to find near optimal solution. GA can be used for FIC as a simple classifier for range blocks to find appropriate domain block and transformation for each range block. At the same time, instead of searching only one point at a time to find optimal solutions, GA use many search points. Therefore, GA has advantage of time and search space reduction [32]. The detailed steps of FIC using GA were described in researches [11, 12, 14, 25] and can be summarized as following steps [11, 30]:

1. Initialization by determining population size (number of chromosomes in population) and number of generations. Increasing population size will decrease the results errors.

2. First generation is a set of random chromosomes that is created to represent the solution area. The initial population is created randomly from images pixels using chromosomes (strings with 0's and 1's).
3. Local iterated function system (LIFS) parameters are composed of {x, y: location of best matched domain block, z: isometry type of best matched domain block, a: scaling coefficient, g: offset}.
4. Search of domain block is completed in encoding space (one dimension). Whereas evaluation and selection of domain block is completed in decoding space (two dimension). The image I with dimension w × w is divided into (b × b) blocks. Search of domain block position (top-left coordinate) is coded by color code g_i (i = 1; 2 … n).
5. Determination of fitness function for FIC is defined by Eq. 4 [11]:

$$Fitness(x) = \begin{Bmatrix} \frac{1}{mse(x)+1} & mse(x) > \varepsilon \\ \infty & mse(x) \leq \varepsilon \end{Bmatrix} \tag{4}$$

where mse (x) represents MSE between a given range block and domain.

6. Selection to combine proportion choice with protection of best chromosome. The probability of a given chromosome is given by Eq. 5 [11].

$$Pis = Fi/\left(\sum_{i=1}^{m} Fi\right), \quad (i = 1, 2, \ldots, M) \tag{5}$$

where, M is population number and Fi is fitness of chromosome i.

7. Crossover (Pc) to combine the selected two chromosomes in current population to form new two chromosomes.
8. Mutation (Pm) changes the two selected chromosomes with a mutation probability and flips a bit (1–0 or 0–1).
9. New generation of new population is created randomly from evaluation each chromosome in previous generation according to fitness function to get new chromosomes from old chromosomes.
10. Steps (5–9) are repeated to find the final generation of chromosomes. The best chromosome represents the optimal solution with lowest fitness function value.

Finally, one of the parallel computing approaches can be used to implement GA for FIC to reduce the execution time required by image encoding process.

5 MATLAB Parallel Computing Approaches

Parallel computing includes exchanging information between many connected computers to increase speed of computations needed by single computer and provide large amounts of memory for program execution [33]. Parallel MATLAB is an extension of MATLAB that takes advantage of multicore desktop machines and clusters. Parallel Computing Toolbox runs on a desktop and can take specifications of up to 8 cores there. Parallel programs can be run interactively or in batch [27].

The Distributed Computing Server controls the parallel execution of MATLAB on a cluster with tens or hundreds of cores. There are several ways to execute the parallel MATLAB program: interactive local (matlabpool); indirect local, (batch); and finally, indirect remote (batch) [27].

The general standard used for implementing parallel programs on many processors in parallel computing is message passing interface (MPI) [34]. To implement MPI concepts and allow program based on Mathlab to be run on a parallel computer, a set of Matlab scripts (MatlabMPI) [35] is used MATLAB supported built-in data types and data structures that supported in parallel programming environment.

5.1 Co-distributor Approach

Co-Distributed arrays are special arrays that store segments of data on Matlab workers that are participating in a parallel computing problem. Distributed arrays can handle larger data sets and can be constructed in several ways: using constructor functions such as rand, ones, and zeros; and concatenating arrays with same name but different data on different labs [27].

Full content of array is stored in workspace of each lab when using normal arrays. Whereas Co-Distributed arrays are partitioned into segments and store each segment in the workspace of different lab. Each lab has its own array segment to work with. Reducing the size of array that each lab has to store and process means a more efficient use of memory and faster processing [36]. The MATLAB software partitions the array into segments and assigns one segment to each lab when an array is distributed to a number of labs. Co-Distributed is an approach in which we can access data of arrays distributed among workers in parallel pool. This approach partition arrays across MATLAB workers for data-parallel computing. There is one manager process to supervise workers who cooperate on a single program. Each worker: has an identifier; knows how many other workers; and determine its behavior based on ID. Each worker runs on a separate core and there is separate workspace used for each worker. The client program modifies data on worker [27].

In this study, the population size is determined with number of chromosomes that comprise the whole population. Then, Co-Distributed arrays are used to distribute chromosomes to labs to improve the speed and evaluating fitness function.

5.2 ParFor Approach

The concept of Parallel for loop (ParFor) in MATLAB is the same as the standard for-loop. MATLAB executes a set of statements over range of values. Part of ParFor body is executed on MATLAB client (manager) and other part is executed in parallel on labs (workers). Labs evaluate iterations in no particular order and

independently of each other. The data on which ParFor operates is sent from the manager to workers, and the results are sent back to manager and pieced together.

ParFor-loop is useful in situations where you need many loop iterations of a simple calculation. ParFor divides the loop iterations into groups so that each lab executes some portion of total number of iterations. ParFor are useful when we have loop iterations that take a long time to execute because the lab can execute iterations simultaneously [29, 36].

The ParFor loop is easy to use but it only executes parallelism in terms of loops. Each execution of ParFor-loop body is iteration. The MATLAB workers evaluate iterations in no particular order and independent of each other. When number of workers is equal to the number of loop iterations, then each worker performs only one loop iteration. But when there are more iterations than workers, then many workers perform more than one loop iteration. Using ParFor, the workers are anonymous and the memory is shared/copied/returned.

ParFor changes how any program do calculations. It asserts that all iterations of loop are independent and can be done in any order or in parallel. The execution begins with single processor (client). When ParFor loop is encountered, the client is helped by pool of workers. Each worker is assigned some iterations of the loop. The client resumes control of execution when the loop is completed. MATLAB ensures that the results are the same whether the program is executed sequentially or with help of workers. We were only able to parallelize the loop because the iterations were independent, that is, the results did not depend on the order in which the iterations were carried out [27]. Therefore, ParFor-loop can be used in this research to speed up GA selection, crossover and mutation. This is done because loops in these operations are independent of each other.

5.3 Single Program Multiple Data Approach

In SPMD, single program runs simultaneously across all workers and enables easy writing and debugging whereas multiple data can spread across workers and runs serially if no workers are available. Block of code executes simultaneously on multiple labs in a MATLAB pool. Each lab can operate on a different distributed data set and communicate with other participating labs while performing parallel computations [37]. SPMD can be used to execute code in parallel on workers of parallel pool.

SPMD command can be regarded as a very simplest version of MPI. There is only one client process and supervising workers who cooperate on a single program. Each worker has its own identifier (ID), runs on a separate core, uses separate workspace, and knows how many other workers. A common program is used and all workers meet at synchronization points. The client program can examine/modify data on any worker and any two workers can communicate using messages. SPMD similar to MPI by allowing design any kind of parallel computation but requires re-arranging program and data.

In SPMD, client and workers share a single program in which some commands are delimited within blocks opening with SPMD and closing with end. The client executes commands up to first SPMD block when it pauses but the workers execute code in block. The client resumes execution when workers finish. The variables' values defined in client program can be referenced by workers but not changed. Whereas variables defined by workers can be referenced/changed by client. A program can contain several SPMD blocks. The workers pause when execution of one block is completed but they do not disappear and their workspace remains intact. A variable set in one SPMD block will still have that value if another SPMD block is encountered. In MATLAB, variables defined in a function disappear once the function is exited. The same thing is true for MATLAB program that calls a function containing SPMD blocks. The worker data is preserved from one block to another inside the function. The worker data is disappears when the function is completed [27].

5.4 Parallel Cluster

Parallel Computing Toolbox software has the ability to run a local cluster of workers on client machine to run jobs without requiring MATLAB Distributed Computing Server software. For this reason, all processing required for client, scheduling, and task evaluation is performed on the same computer. This give opportunity to develop, test, and debug parallel applications before running them on network cluster. Many of functions that support the use of cluster profiles are: batch; parpool; and parcluster. Any job can be programmed on local cluster when programmer needs to tasks of job to evaluate functions. Using parallel computing toolbox client session, the basic steps in creating and running a job that contains simple tasks are [27]:

1. Identify a cluster by using parallel.defaultClusterProfile and use parcluster to create object c to represent this cluster.

   parallel.defaultClusterProfile('local');
   c = parcluster();

2. Create job j on the cluster by executing this line: j = createJob(c)

   1. Create tasks within job j. Each task evaluates mathematical operations that are passed as inputs argument.

   createTask(j, @ operation, 1, {[1 1]});
   createTask(j, @ operation, 1, {[2 2]});
   createTask(j, @ operation, 1, {[n n]});

3. Submit job to queue for evaluation. The scheduler distributes job's tasks to MATLAB workers. The local cluster start MATLAB worker sessions by: submit(j);

4. Wait for job to complete, then get results from all tasks of job.
5. Remove job from scheduler's storage location when obtaining results.

5.5 Batch File Approach

The batch command can send the specified job anywhere and get results back. The programmer needs to set up an account on the remote machine and needs to define a configuration on his desktop that tells him how to access the remote machine. The batch command is used to run a Batch Job to offload work from MATLAB session to run in the background in another session. Batch runs code on a local worker or a cluster worker but does not require a parallel pool. You can use batch to run either scripts. The batch approach can be summarized by the following steps [27]:

1. To create the script, type: edit mywave
2. Write MATLAB code in MATLAB Editor, and save the file and close editor.
3. Use batch command in MATLAB Command Window to run the script on a separate MATLAB worker: job = batch('mywave')
4. The batch command does not block MATLAB, so you must wait for the job to finish before you can retrieve and view its results: wait(job)
5. The load command transfers variables created on worker to client workspace, where you can view the results: load(job, 'A'), plot(A)
6. When job is complete, permanently delete its data and reference from workspace.

6 Research Methodology

In this chapter, three MATLAB parallel approaches from MATLAB Parallel Computing Toolbox such as ParFor, Co-Distributor and Parallel Cluster were used to implement GA for FIC. This is done according to manager/worker model of parallel programming to reduce computation time and increase compression performance (PSNR and CR). Seven computers (1 manager and 6 workers) were used.

The execution environment of MATLAB needs to be configured before carrying out the parallel programming. This procedure has been introduced in MATLAB toolbox' user guide. MATLAB Distributed Computing Server (MDCS) [37] is adopted for the server and each worker. This MDCS is downloaded and installed on all used computers to ensure the parallel programming service. Manager/Worker technique was selected from MDCS. GA parameters can be set by manager as follows: 50 generation, two values (240, 150) are selected for population size for the main computer (manager). And two other values (40, 25) are used as population size in worker. The value of crossover (pc) is equal 1 and mutation (pm) is equal 0.001. The size of range block is equal 4×4. The used Fitness function is represented by Eq. 4. The suggested parallel computing approach for FIC based GA can be summarized in Fig. 1.

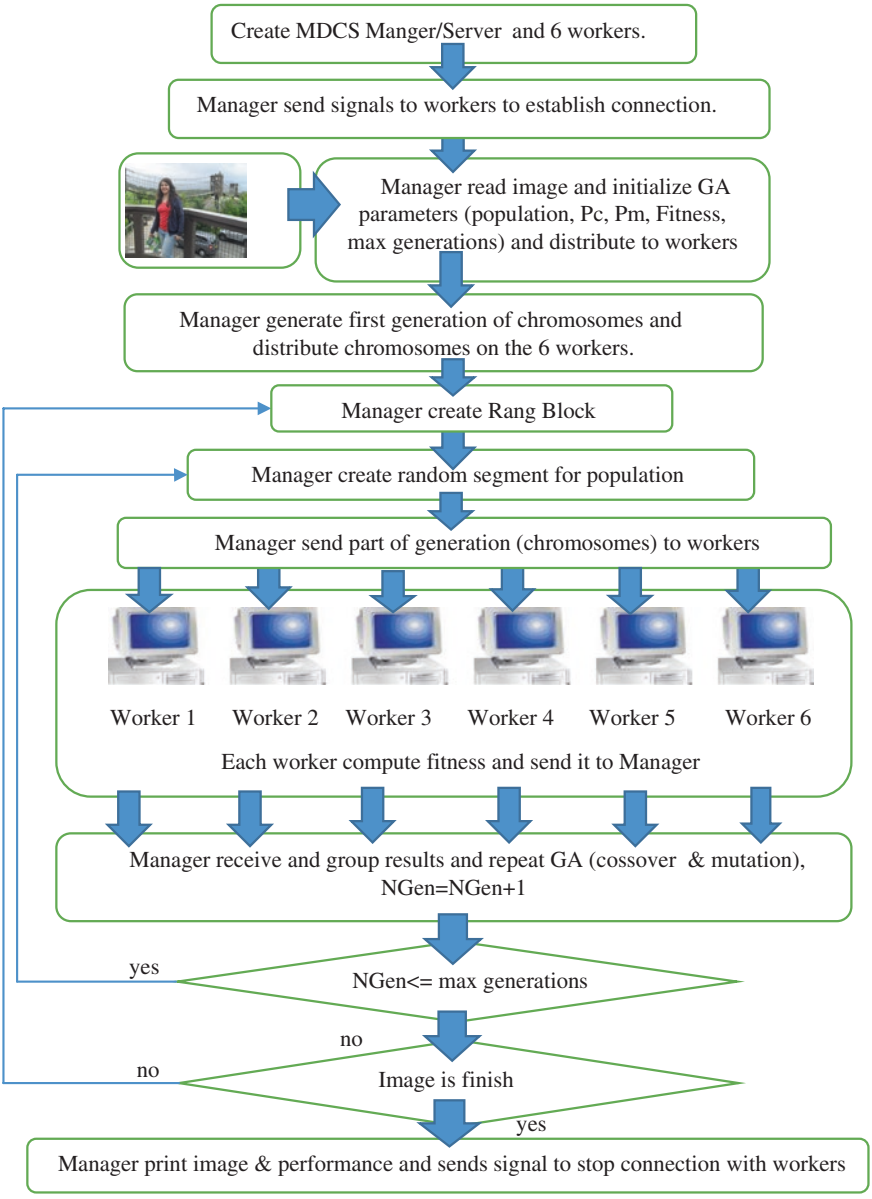


Fig. 1 The suggested parallel computing approach for FIC based GA

7 Experiments

Many experiments were conducted based on three parallel approaches from MATLAB Parallel Computing Toolbox. These approaches are: ParFor, CoDistributor, and Parallel Cluster. This is done using 5 colored image (DaliaPhoto1.JPEG,

DaliaPhoto2.JPEG, DaliaPhoto3.JPEG, DaliaPhoto4.JPEG and DaliaPhoto5.JPEG) with different sizes and dimension as shown in Fig. 2. This section includes the evaluation of using these three MATLAB parallel approaches in implementing GA for FIC.

The first three experiments were accomplished by setting GA iterations to 50 and population size equal 240. The first experiment is related to compress DaliaPhoto1 using FIC PGA [25] approach. The second experiment is related to compress "DaliaPhoto1" using FIC PGA with ParFor approach. Whereas the third experiment is related to compress DaliaPhoto1 using FIC PGA with

Fig. 2 The images. **a** Image: DaliaPhoto1.JPEG. **b** Image: DaliaPhoto2.JPEG. **c** Image: DaliaPhoto5.JPEG. **d** Image: DaliaPhoto3.JPEG. **e** Image: DaliaPhoto4.JPEG

Co-Distributor approach. Table 1 shows results (CR, PSNR and computation time (in seconds)) for of the first, second and third experiments respectively when using same values of parameters (population size equal 240, and 50 iterations and same number of workers. We can note from Table 1 that, the computation time is decreased when increasing the number of workers.

Another 3 experiments (4, 5 and 6) were accomblished by setting GA iterations to 50 and population size equal 150. Each of experiment 4, experiment 5, and experiment 6 are related to compress image DaliaPhoto1 using FIC PGA [25], ParFor and Co-Distributor approaches respectivelly. Table 2 shows results (CR, PSNR and computation time (in seconds)) for of fourth, fifth and sixth experiments respectively with population size equal 150. We can note from Table 2 that, the computation time is decreased when increasing the number of workers.

At the same time, we can note from Tables 1 and 2 that computation time is decreased when decreasing the population size. From Table 1, the execution speed of FIC PGA by Co-distributor has reached 51 s with PSNR equal 32.29 and CR equal 87.88 % when conducting two workers with population size equal 240. While the time required by FIC PGA [25] is 67 s and the time required by ParFor is 60 s. This means that the Co-Distributor approach lead to better results related

Table 1 FIC PGA [25], ParFor, and co-distributor for image: daliaphoto1, when pop size = 240

Iterations = 50, population size = 240									
	FIC PGA [25]			ParFor			Co-distributor		
Workers	Time (sec)	PSNR	CR (%)	Time (sec)	PSNR	CR (%)	Time (sec)	PSNR	CR (%)
1	75	30.19	86.30	69	31.44	87.44	60	31.64	87.64
2	67	30.66	86.70	60	31.99	87.89	51	32.29	87.88
3	54	31.47	87.31	46	32.44	88.22	35	32.74	88.56
4	47	32.17	87.79	35	32.89	88.78	22	32.99	88.97
5	30	33.20	88.19	22	33.25	89.21	15	33.75	89.75
6	26	33.29	89.11	18	33.72	89.77	10	33.92	90.65

Table 2 FIC PGA [25], ParFor, and co-distributor for image: daliaphoto1, when pop size = 150

Iterations = 50, population size = 150									
	FIC PGA [25]			ParFor			Co-distributor		
Workers	Time (sec)	PSNR	CR (%)	Time (sec)	PSNR	CR (%)	Time (sec)	PSNR	CR (%)
1	74	32.66	88.17	66	32.88	88.44	54	32.92	88.64
2	61	32.76	88.31	53	33.22	88.87	41	33.32	88.90
3	50	33.18	89.35	36	33.71	89.66	25	33.87	89.75
4	37	33.30	89.49	28	33.91	89.78	17	34.11	89.91
5	24	33.78	89.59	19	34.14	89.92	10	34.54	90.32
6	20	34.00	89.83	15	34.44	90.21	6	34.74	90.45

to processing time, PSNR and CR. Table 1 shows also reducing the execution time when adopting Co-Distributor with the use of 2–6 workers and reaching (10 s) when using 6 workers with improving in PSNR (33.92 db) and CR equal 90.65 %. This means that the execution speed can be decreased by increasing the number of workers. Acceleration in execution time of FIC PGA has been noticed in Table 2 while decreasing the generation size from 240 to 150. The execution time of FIC PGA by Co-distributor ranged between 54 and 6 s with maintaining increasing in CR to reach 90.45 % and PSNR reached 34.74 db for 6 workers. Figure 3 shows the differences in computation time required by FIC PGA [25], ParFor and Co-distributor (for population size = 240). Whereas Fig. 4 shows the differences in computation time required by FIC PGA [25], ParFor and Co-distributor

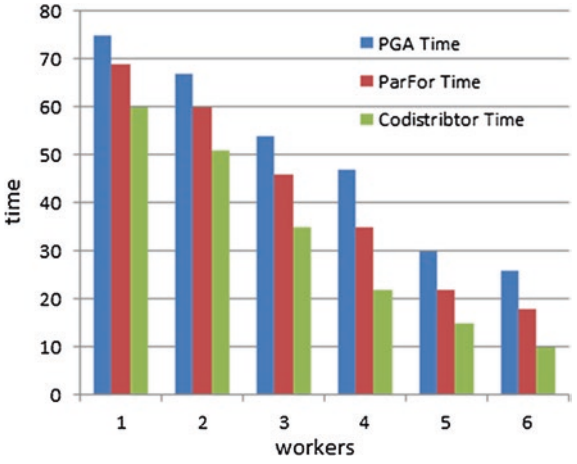


Fig. 3 Time requird for FIC PGA [25], ParFor and Co-distribtor (population size = 240)

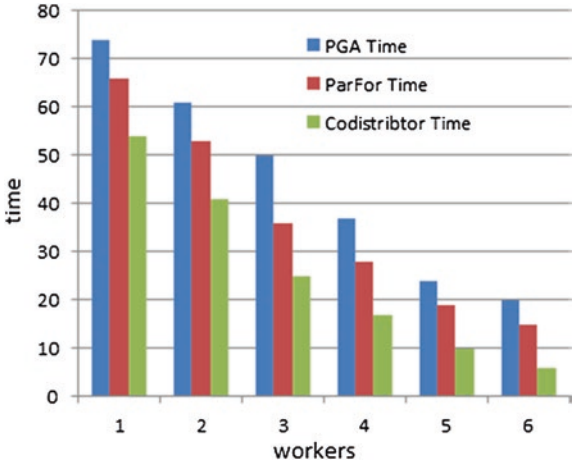


Fig. 4 Time requird for FIC PGA [25], ParFor and Co-distributor when population size = 150

(for population size = 150). Each of Figs. 2 and 3 shows that the Co-distributor approach requires less computation time for compression.

Many other experiments were conducted to determine the efficiency of ParFor and CoDistributer parallel techniques on different images (DaliaPhoto2 to DaliaPhoto5). Table 3 shows the results of these experiments when use population size equal 240 and number of iterations equal 50.

We can note from Table 3 in the section related to population size equal 240 for ParFor that the average value of time when using one worker is 61.75 whereas the average value when using 6 workers is 17.5 and this is big difference. At the same time the average value of PSNR related to one worker is 31.46 and average value of PSNR related to 6 workers is 33.74. Also the average value of CR related to one worker is 87.33 and average value related to 6 workers is 89.815. This means that increase the number of workers will decrease the required time; increase the PSNR and increase the CR. At the same time, Table 4 shows the results of these

Table 3 ParFor, and co-distributor for images: daliaphoto2 to daliaphoto5 (pop size = 240)

DaliaPhoto2.JPEG	Workers	Population size = 240, Iterations = 50					
		ParFor			Co-distributor		
		Time (sec)	PSNR	CR (%)	Time (sec)	PSNR	CR (%)
	1	68	31.48	87.48	52	31.69	87.77
	2	62	32.11	87.85	47	32.35	87.85
	3	44	32.47	88.27	33	32.88	88.71
	4	36	32.87	88.75	22	32.95	88.99
	5	24	33.27	89.27	17	33.87	89.94
	6	19	33.75	89.84	9	33.98	90.74
DaliaPhoto3	1	66	31.55	87.22	51	31.61	87.72
	2	64	32.22	87.81	43	32.32	87.81
	3	46	32.51	88.28	35	32.84	88.79
	4	32	32.85	88.72	24	32.92	88.95
	5	27	33.25	89.31	19	33.85	89.97
	6	18	33.75	89.84	10	33.94	90.77
DaliaPhoto4	1	58	31.44	87.33	47	31.65	87.66
	2	61	32.25	87.87	38	32.27	87.78
	3	42	32.55	88.26	30	32.80	88.75
	4	29	32.78	88.78	23	32.88	88.95
	5	25	33.27	89.35	17	33.78	89.95
	6	17	33.76	89.87	11	33.90	90.71
DaliaPhoto5	1	55	31.40	87.31	49	31.62	87.53
	2	63	32.28	87.76	36	32.25	87.63
	3	39	32.61	88.31	29	32.76	88.64
	4	27	32.72	88.55	21	32.74	88.80
	5	22	33.29	89.25	19	33.82	89.76
	6	16	33.71	89.71	9	33.82	90.66

Table 4 ParFor, and co-distributor for images: daliaphoto2 to daliaphoto5 (pop size = 150)

DaliaPhoto2.JPEG	Workers	Population size = 150, Iterations = 50					
		ParFor			Co-distributor		
		Time (sec)	PSNR	CR (%)	Time (sec)	PSNR	CR (%)
	1	65	32.88	88.44	51	32.95	88.75
	2	50	33.22	88.87	39	33.45	88.95
	3	35	33.71	89.66	22	33.88	89.77
	4	28	33.91	89.78	11	34.32	90.22
	5	17	34.14	89.92	7	34.65	90.43
	6	11	34.44	90.21	3	34.89	90.65
DaliaPhoto3	1	67	32.85	88.41	49	32.91	88.68
	2	51	33.25	88.82	38	33.39	88.92
	3	37	33.73	89.62	21	33.86	89.70
	4	25	33.89	89.74	13	34.28	90.19
	5	19	34.16	89.89	9	34.58	90.37
	6	12	34.47	90.19	4	34.79	90.57
DaliaPhoto4	1	55	32.79	88.42	45	32.88	88.60
	2	46	33.22	88.80	31	33.29	88.94
	3	32	33.70	89.64	19	33.78	89.74
	4	23	33.87	89.70	11	34.30	90.17
	5	17	34.14	89.82	8	34.51	90.34
	6	11	34.42	90.22	2	34.72	90.51
DaliaPhoto5	1	49	32.66	88.22	42	32.66	88.51
	2	41	33.12	88.65	26	33.21	88.22
	3	29	33.22	89.43	17	33.65	89.13
	4	25	33.61	89.32	9	34.32	90.12
	5	16	34.11	89.76	5	34.45	90.21
	6	9	34.22	90.11	1	34.61	90.34

experiments when use population size equal 150 and number of iterations equal 50. We can note from this table that decrease the population size to 150 chromosomes will decrease the time of 6 workers to 9 s when implementing GA for FIC using MATLAB ParFor. Also decrease the time of 6 workers to 1 s when implementing GA for FIC using MATLAB CoDistributor.

As an average performance of the parallel computing results, Table 5 shows the average values of (Time, PSNR and CR) when applying ParFor and CoDistributor to compress these four images when use population size equal 240 and number of iterations equal 50. At the same time, Table 6 shows the average values of (Time, PSNR and CR) when applying ParFor and CoDistributor to compress the same images when use population size equal 150 and 50 iterations.

We can note from Tables 5 and 6 that the average values of time related to Co-Distributor are less than the average values of time related to ParFor. The final experiment was conducted based on other MATLAB parallel approach named

Table 5 Averages when applying ParFor and co-distributor, pop size = 240

Workers	Population size = 240, Iterations = 50					
	ParFor			Co-distributor		
	Time (sec)	PSNR	CR (%)	Time (sec)	PSNR	CR (%)
1	61.75	31.467	87.335	49.75	31.642	87.67
2	62.5	32.215	87.822	41	32.297	87.7675
3	42.75	32.535	88.28	31.75	32.82	88.7225
4	31	32.805	88.7	22.5	32.872	88.9225
5	24.5	33.27	89.295	18	33.83	89.905
6	17.5	33.742	89.815	9.75	33.91	90.72

Table 6 Averages when applying ParFor and co-distributor, pop size = 150

workers	Population size = 150 Iterations = 50					
	ParFor			Co-distributor		
	Time (sec)	PSNR	CR (%)	Time (sec)	PSNR	CR (%)
1	59	32.795	88.3725	46.75	32.85	88.635
2	47	33.2025	88.785	33.5	33.335	88.7575
3	33.25	33.59	89.5875	19.75	33.7925	89.585
4	25.25	33.82	89.635	11	34.305	90.175
5	17.25	34.1375	89.8475	7.25	34.5475	90.3375
6	10.75	34.3875	90.1825	2.5	34.7525	90.5175

"Parallel Cluster" approach. This is done to compare the performance of the two MATLAB approaches ParFor and CoDistributor with other approach. Table 7 shows the results of the three approaches when use population size equal 150.

We can note from Table 7 that lowest computation times were obtained from parallel Cluster approach.

Table 7 ParFor, co-distribtor and parallel cluster for image: daliaphoto1

Iterations = 50, population size = 150									
	ParFor			Co-distributor			Parallel cluster		
Workers	Time (sec)	PSNR	CR (%)	Time (sec)	PSNR	CR (%)	Time (sec)	PSNR	CR (%)
1	66	32.88	88.44	54	32.92	88.64	47	32.99	88.88
2	53	33.22	88.87	41	33.32	88.90	38	33.50	88.97
3	36	33.71	89.66	25	33.87	89.75	22	33.92	89.89
4	28	33.91	89.78	17	34.11	89.91	14	34.36	90.01
5	19	34.14	89.92	10	34.54	90.32	7	34.73	90.58
6	15	34.44	90.21	6	34.74	90.45	4	34.98	90.97

8 Conclusion

Three MATLAB parallel approaches (ParFor, Co-Distributor and Parallel Cluster) were adopted in this research to decrease the computation time of FIC based GA. Many experiments were conducted with using: five images (DaliaPhoto1.JPEG to DaliaPhoto5.JPEG); different population size (240 and 150); controlling GA parameters (crossover and mutation). This is done to improve PSNR and CR while decreasing the computation time. The results showed that increasing the number of workers in parallel programming will decrease the computation time required for FIC based GA. Also this time can be decreased when we decrease the population size from 240 to 150 chromosomes with maintaining increasing the decompressed image quality (PSNR) and CR. The experimental results showed that, the Co-Distributor parallel approach requires less computation time than ParFor approach. At the same time, best quality of the decompressed image (PSNR) and CR can be increased when adopting Co-Distributor. At the end, the results of MATLAB parallel computing approach "Parallel Cluster" are better than the results of Co-Distributor approach. In the future, an analytical survey will be conducted to give a detailed description about the literature studies related to implementing GA for FIC based on different parallel computing approaches. Comparisons between the results of this study with the results of the related studies will be conducted.

References

1. Lakshmi, G., Ramamohana Rao, S.: A novel algorithm for image compression based on fractal. Eur. J. Sci. Res. **85**(4), 486–499. ISSN: 1450-216X (2012)
2. William, R.S., Helio, P.: Improved fractal image compression based on robust feature descriptors. Int. J. Image Graph. **11**(4), 571–587 (2011)
3. Hitashi, G.K., Sugandha, S.: Fractal image compression—a review. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **2**(2) (2012)
4. Nileshsingh, V.T., Kakde, O.G.: Color image compression with modified fractal coding on spiral architecture. J. Multimedia **2**(4), 55–66 (2007)
5. Poobal, S., Ravindran, G.: Arriving at an optimum value of tolerance factor for compressing medical images. World Acad. Sci. Eng. Technol. **17**, 997–1001 (2008)
6. Sankaragomathi, B. et al.: Encoding video sequences in fractal-based compression. Fractals **15**(4), 365–378 (2007)
7. Venkatasekhar, D., Aruna, P., Parthiban, B.: Fast search strategies using optimization for fractal image compression. Int. J. Comput. IT **2**(3), 437–441 (2013)
8. Seeli, D., Jeyakumar, M.K.: A study on fractal image compression using soft computing techniques. IJCSI Int. J. Comput. Sci. Issues **9**(6), No. 2, 420–430 (2012)
9. Mitchell, M.: An Introduction to Genetic Algorithms, A Bradford Book. The MIT Press, Cambridge (1999)
10. Sivanandam, S.N., Deepa, S.N.: Introduction to Genetic Algorithms. Springer, Berlin. LCCN: 2007930221, ISBN 978-3-540-73189-4 (2008)

11. Xi, L., Liangbin, Z.: A study of fractal image compression based on an improved genetic algorithm. Int. J. Nonlinear Sci. **3**(2), 116–124 (2007)
12. Chakrapani, Y., Soundara, K.R.: Genetic algorithm applied to fractal image compression. ARPN J. Eng. Appl. Sci. **4**(1), 53–58 (2009)
13. Gaona, M., Walter, S.K.: Genetic adaptive coding optimization applied to fractal image compression. Int. J. Imaging Syst. Technol. **10**, 369–378 (1999)
14. Mitra, S.K., et al.: Technique for fractal image compression using genetic algorithm. IEEE Trans. Image Process. **7**(4), 586–593 (1998)
15. Nadira, B., et al.: Iteration-free fractal coding for image compression using genetic algorithm. Int. J. Comput. Intell. Appl. **7**(4), 429–446 (2008)
16. Faraoun, K.M.: Optimization of fractal image compression based on genetic algorithms. In: SETIT 2005, 3rd International Conference on Sciences of Electronic, Technologies of Information and Telecommunications, 17–21 March 2005
17. Zheng, Y., Liu, G., Niu, X.: An improved fractal image compression approach by using iterated function system and genetic algorithm. Comput. Math. Appl. **51**(11), 1727–1740 (2006)
18. Nowostawski, M., Poli, R.: Parallel genetic algorithm taxonomy. In: Proceeding of 3rd International Conference on Knowledge-Based Intelligent Information Engineering Systems, 1999. IEEE, Dunedin (2000)
19. Plamenka, B.: Solving the travelling salesman problem in parallel by genetic algorithm on multicomputer cluster. In: International Conference on Computer Systems and Technologies (2006)
20. Shisanu, T., Prabhas, C.: Parallel genetic algorithm for finite-state machine synthesis from input/output sequences. In: Conference on Evolutionary Computation and Parallel Processing Workshop, Genetic and Evolutionary Computation, Las Vegas, Nevada, USA, pp. 20–24 (2000)
21. Peng, H., et al.: Design of parallel algorithms for fractal video compression. Int. J. Comput. Math. **84**(2), 193–202 (2007)
22. Hammerle, J., Uhl, A.: Fractal image compression on MIMD architectures II: classification based speed-up methods. J. Comput. Inf. Technol. pp. 71–82 (2000)
23. Yunda, S., Zhao, Y., Yuan, B.: A parallel implementation of improved fractal image coding based on tree topology. Chin. J. Electron. **12**(2) (2003)
24. Peter, B.: Maximal processor utilization in parallel quadtree-based fractal image compression on mimd architectures. Studia Univ. Babes-Bolyai, Informatica **ix**(2) (2004)
25. AL-Allaf, O.N.A., AbdAlKader, S.A.: Genetic algorithm based on parallel computing to improve the performance of fractal image compression system. Eur. J. Sci. Res. **92**(2), 172–183 (2012)
26. Nadira Banu Kamal, A.R., Priyanga, P.: Parallel fractal coding for color image compression using genetic algorithm and simulated annealing. Int. J. Comput. Sci. Inf. Technol. **4**(6), 1017–1022 (2013)
27. MATLAB Parallel Computing Toolbox User's Guide 4.3, 2014b (2014)
28. MATLAB, Optimization Toolbox™ User's Guide, R2014b (2014)
29. Li, N., Gao, P., Lu, Y., Yu W.: The implementation and comparison of two kinds of parallel genetic algorithm using matlab. In: 9th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, IEEE Computer Society (2010)
30. Shouji, C., Liming, Z.: Fractal and image compression. Shanghai Science and Technology Education Publishing House (1998)
31. Fisher, Y.: Fractal image compression. Fractals **2**(3), 25–36 (1994)
32. Uma, K., et al.: Image compression using optimization techniques. Int. J. Eng. Res. Dev. **5**(5), 1–7. e-ISSN: 2278-067X (2012)
33. Kim, H., et al.: Introduction to Parallel Programming and pMatlab v2.0. Mathworks Inc. (2009). http://www.ll.mit.edu/mission/isr/pmatlab/pMatlab_intro.pdf
34. MPI, A Message-Passing Interface Standard, Version 2.1, University of Tennessee, Knoxville, Tennessee 23 June 2008. http://www.mpi-forum.org/docs/mpi21-report.pdf
35. Kepner, J.: Parallel programming with MatlabMPI. In: 5th High Performance Embedded Computing (HPEC2001) workshop, MIT Lincoln Laboratory, Lexington, MA (2002)

36. MATLAB 7, Programming Fundamentals. The MathWorks™, Inc. (2008). www.mathworks.com
37. MATLAB: Distributed Computing Server™ 5 System Administrator's Guide. The MathWorks, Inc. 3, Apple Hill Drive, Natick, MA 01760-2098, MathWorks, Inc. (2010).www.mathworks.com