Editor Dana Simian

Proceedings of
Second International
Students Conference on Informatics

ICDD

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

Romania, Sibiu, April 26-28, 2012

Editor DANA SIMIAN

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

Proceedings of International Students Conference on Informatics
ICDD

April 26th – 28th, 2012
Sibiu, Romania

"Lucian Blaga" University of Sibiu

“Lucian Blaga” University, Sibiu, 2012

Editor Dana Simian

All papers in this volume were peer review by two independent reviewers

ISSN 2069-964X

Associated editor Laura Stoica

Proceedings of International Students Conference on Informatics ICDD

April 26th – 28th, 2012, Sibiu, Romania

Copyright @ 2012 All rights reserved to editors and authors

Preface

This volume contains refereed papers which were presented at the Second International Students Conference on Informatics - ICDD – 2012. The conference was held between April 26th – 28th, at Sibiu, Romania. It was organized by the Department of Informatics at the Faculty of Sciences, from ” Lucian Blaga” University of Sibiu.

The conference is addressed to bachelor and master level students. Conference aim is to bring together students from different universities from all over the world to discuss and present their researches on informatics and related fields (like computational algebra, numerical calculus, bioinformatics, etc) and their original results.

We thank all the participants for their interesting talks and discussions. We also thank the members of the scientific committee for their help in reviewing the submitted papers and for their contributions to the scientific success of the conference and to the quality of this proceedings volume.

April 2012

Dana Simian
Conference chairman

Motto:

"There are no limits, only your imagination"

Proceedings of International Students Conference on Informatics, ICDD – 2012
IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT
Sibiu, Romania

Editor *Prof. PhD. Dana Simian*

Scientific committee

- Prof. PhD. Grigore Albeanu - University of Bucharest, Romania
- Prof. PhD. Florian Boian - "Babes-Bolyai" University of Cluj-Napoca, Romania
- Prof. PhD. Oleksandr Dorokhov - Kharkiv National University of Economics, Ukraine
- Prof. PhD. Heiner Gonska - Duissburg-Essen University, Germany
- Prof. PhD. Gheorghe Grigoras - "Alexandru Ioan Cuza" University of Iasi, Romania
- Prof. PhD. Katalina Grigorova - University of Ruse, Bulgaria
- Prof. PhD. Ion Iancu - University of Craiova, Romania
- Prof. PhD. János Karsai - University of Szeged, Hungary
- Prof. PhD. Milena Lazarova - Technical University of Sofia, Bulgaria
- Prof. PhD. Daniela Marinescu - Transilvania University of Brasov, Romania
- Prof. PhD. Ioana Moisil - "Lucian Blaga University" of Sibiu, Romania
- Prof. PhD. Mariana Nagy - "Aurel Vlaicu" University of Arad, Romania
- Prof. PhD. Bazil Parv - "Babes-Bolyai" University of Cluj-Napoca, Romania
- Prof. PhD. Dana Petcu - West University of Timisoara, Romania
- Prof. PhD. Valer Rosca - "Lucian Blaga" University of Sibiu, Romania
- Prof. PhD. Ernest Scheiber - Transilvania University of Brasov, Romania
- Prof. PhD. Dana Simian - "Lucian Blaga" University of Sibiu, Romania
- Prof. PhD. Luminita State - University of Pitesti, Romania
- Prof. PhD. Mihai Talmaciu - University of Bacau, Romania
- Prof. PhD. Milan Tuba - Megatrend University of Belgrade, Serbia
- Prof. PhD. Dan Eugen Ulmet - University of Applied Sciences Esslingen, Germany
- Assoc. Prof. PhD. Kiril Alexiev - Bulgarian Academy of Sciences, Bulgaria
- Assoc. Prof. PhD. Vasile Aurel Căus - University of Oradea, Romania
- Assoc. Prof. PhD. Ioana Chiorean - "Babes-Bolyai" University of Cluj-Napoca, Romania
- Assoc. Prof. PhD. Stefka Fidanova,- Bulgarian Academy of Sciences, Bulgaria
- Assoc. Prof. PhD. Antoanela Naaji - "Vasile Goldis" Western University of Arad, Romania
- Assoc. Prof. PhD. Petrica Pop - North University of Baia Mare, Romania
- Assoc. Prof. PhD. Victoria Iordan - West University of Timisoara, Romania
- Lect. PhD. Mihaela Ciortea - "1 December 1918" University of Alba Iulia, Romania
- Lect. PhD. Ralf Fabian - "Lucian Blaga" University of Sibiu, Romania
- Lect. PhD. Daniel Hunyadi - "Lucian Blaga" University of Sibiu, Romania
- Lect. PhD. Gabriela Moise - Petroleum-Gas University of Ploiesti, Romania
- Lect. PhD. Mircea Neamțu - "Lucian Blaga" University of Sibiu, Romania
- Lect. PhD. Corina Rotar - "1 December 1918" University of Alba Iulia, Romania
- Lect. PhD. Florin Stoica - "Lucian Blaga" University of Sibiu, Romania
- Lect. PhD. Anca Vasilescu - Transilvania University of Brasov, Romania

Contents:

Info – Route: A graphs-based application for travel and touristic planning ... 11
Cristian-Ionel Aldulea, Bogdan-Mihai Fieraru, Vlad Vasile
Coordinator: Anca Vasilescu

Psihological tests ………………………………………………….………….... 19
Ioana Madalina Anghel, Petru Barko
Coordinator: Anca Vasilescu

RevontuLED……………………………...………………………………… 25
Ana Barar
Coordinator: Alexandru Radovici

A language for teaching kids programming ……………….…………...….... 28
Ioana Alexandra Bărbătei
Coordinator: Alexandru Radovici

Machine learning in gear changing ……………………………………..……… 32
Alexandru Petruț Becheru
Coordinator: Cătălin Stoean

Game Based Learning. Project Sotirios………………………….………….. 44
Florentin Bota
Coordinator: Dana Simian

Self growing ASP.NET site with custom CMS…………………..………….. 51
Francesco Capuzzo
Coordinator: Dana Simian

Multitouch Vector Graphics Shared Mobile Whiteboard ……………….……… 56
Ioana Culic
Coordinator: Alexandru Radovici

Graphical simulation of chemical reactions……………………….…………… 60
Farcaş Nicolae
Coordinator: Camelia-M. Pintea

Object recognition in OpenNERO ………………………………….…………. 65
Filip F. Filipov
Coordinator: Milena Lazarova

Optimization of Parallel Implementation of Quicksort ……………………..… 70
Dimitar Gadzhev
Coordinator: Milena Lazarova

Intelligent reasoning agents used in marketing strategies …………………………… 77
Valentina Lazăr
Coordinator: Florin Stoica

Traffic Sign Detection and Recognition ………………………………………………. 85
Laura Nicoleta Mamina
Coordinator: Adrian Deaconu

DCT based Digital Watermarking for Multimedia Content ……………………….. 90
Alin Manta, Arthur Mărginean
Coordinator: Ralf Fabian

Robotic Arm Playing Tic-Tac-Toe Game ……………………………………………… 97
Kaloyan Mironov
Coordinator: Katalina Grigorova

Solar Tracking System Using Computer Vision ………………………………….... 103
Kaloyan Mironov, Vladimir Kotsev
Coordinator: Katalina Grigorova

Fast Conversion Clever Interpretation ………………………………………………. 110
Diana Alina Moşneag, Diana Maria Duca, Radu Marin Roț
Coordinator: Daniel Hunyadi

Optical Data Transfer. Data Compression and Encoding Using Color …..… 116
Vasile-Nicuşor Nechifor, Daniel-Sorin Radu
Coordinator: Dana Simian

Droid monitor - Android monitoring software - …………………………………… 126
Dragoş Iulian Obancea

Simulation of Information Influence of Social Networks on Client's Loyalty by the Anylogic Package Tools 138
Svetlana A. Panasenko
Coordinator: I. A. Ushakova

Multi-User Drawing Board ……………………………………………………………... 144
Iulia Paraicu
Coordinator: Alexandru Radovici

Tank::Shooter ………………………………………………………………………….. 148
Popa Adrian, Wandschneider Josef-Alexander
Coordinator: Simian Dana

Denoising algorithms for processing images corrupted by additive noise …... 153
Nikola Rankovic
Coordinator: Milan Tuba

Image rendering using parallel ray tracing algorithm ……………………….. 166
Filip Slavov
Coordinator: Milena Lazarova

Article withdrawn at the author's request ……………………….

Cloud Collaboration - Platform for online collaboration in open source distributed cloud systems …… 179
George Suciu
Coordinator: Simona Halunga

Research and analysis of the method for the total tardiness time minimization of jobs with due dates on a computing device ………….. 188
Maksym Sukhonos, Dmytro Lienko
Coordinator: Sergey Minukhin

Sharing interactive SVG documents through XMPP ………………………… 193
Ştefan Alexandru
Coordinator: Alexandru Radovici

Bricks breaking game …………..……………………………………………. 197
Zoltan Tamasi, Zsigmond-Attila Szasz, Izabella Szornyi-Chis
Coordinator: Anca Vasilescu

Implementation of image analogies using neural networks ……………….……. 202
Jelena Tasic
Coordinator: Milan Tuba

Routemaster – A Windows Phone 7 Travel Application ……………………... 214
Tecsa Simona, Stancu Mihai, Pirvu Doru
Coordinator: Hunyadi Daniel

Multilevel image thresholding for image segmentation ……………………..... 219
Ana Tuba
Coordinator: Milan Tuba

Ant Colony Optimization in a parallel framework ……….…………………... 230
George Viorel Vişniuc, Mihai Manea, Radu Rotari, Alexandru Ioan Florea
Coordinator: Cerasela Crişan

Paradigms of Artificial Intelligence Programming: Ruby Case Studies …... 240
Volosincu Bogdan, Rusu Andrei
Coordinator: Ralf Fabian

Participants list …………………….………..……………………………..… 246

Sponsors ……………………………………………………….…..…… 253

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Info – Route:

A graphs-based application for travel and touristic planning

Cristian-Ionel Aldulea, Bogdan-Mihai Fieraru, Vlad Vasile
Teacher Coordinator: Anca Vasilescu

Abstract

This application has been developed using the Delphi programming language. On a global scale, the software makes use of multiple functions, each function using a certain algorithm. The most interesting and complex of these is, by far, a graph theory algorithm which determines the shortest distance between two cities. As a consequence, the entire software could be viewed, at its core, as a graphs-based application. The software consists of two main sections. The first section is designed to present the main Romanian touristic points of interest and to help the user to calculate the minimum distance between two important cities. After all necessary input fields have been filled, the shortest route between the two specified cities will be displayed on the map. The second section provides the user with a diverse map database which includes road maps, tourist maps and county maps for planning a certain route ahead of time.

1 Why we made our choice

Taking into account the current economical and financial circumstances, we believe that tourism is an economical branch with a lot of development potential, which might even be able to significantly improve the state budget. Our country is rich in touristic resources, like historical landmarks, cultural centres and natural landscapes. We developed the Info – Route utility software specifically in order to locate these resources and to find the most convenient ways to reach those available for tourists, guests and even locals.

2 About our application

2.1 Technology that we used

The application was developed using Borland Delphi. Originally developed by Borland as a rapid application development tool for Windows, and as the successor of Borland Pascal, Delphi uses its own Object Pascal dialect of Pascal to generate native code for 32 and 64 bit Windows OS, as well as for 32-bit Max OS X and iOS. Numerous versions of Delphi have been in use from 1995 until today, such as Borland Delphi 1 to 8, 2005, 2006, 2007 CodeGear, Delphi Prism, Embarcadero *(in 2006, Borland's developer tools section was transferred to a subsidiary known as CodeGear, which in turn was sold to Embarcadero Technologies in 2008, thus the name changes),* Delphi 2009, 2010, XE2. [5]

2.2 System requirements

The system requirements for this software are low level:

- Compatible Operating System (Windows XP/Vista/7)
- 1.8 GHz Processor
- 256Mb RAM
- 150 Mb free on HDD

2.3 How it works

The software consists of two main sections. The first section is called “Excursie” (*Trip*) and is designed to present the main Romanian attractions and to help the user to calculate the minimum distance between two important cities in the country. The distance between the two cities and the corresponding shortest route between them will be displayed on the map. Moreover, this application can provide additional data about the chosen route for example, the estimated time and expenses expressed in fuel and money for an average speed of 55 km/h. To receive this information, the user must fill two fields: fuel price and vehicle consumption rate. The second section is dedicated to maps. By accessing this section, the user may view various map types such as road map, tourist map, counties map, which may help him in establishing an itinerary.

2.3.1 The shortest route function

In order to use this function, the user has to open the “Excursie” window by clicking on its respective button. Next, he must set the departure and destination cities; our software will calculate and display the shortest route for viewing. Completing the vehicle fuel consumption rate and gas price fields and clicking on “Detalii” (*Details*) button will give additional information about time and expenses. The selected route can then be viewed on the map using the “Harta Traseu” (*Route Map*) function from the “Utile” *(Utilities*) taskbar. The route display map is reloaded every time the "Afiseaza Drumul" (*Show Route)* button on the "Actiuni" (*Actions)* taskbar menu is used. Each time the "Harta Traseu" window is opened, an image representing the Romanian borders is loaded. Cities are dynamically displayed on the map by utilizing their respective coordinates saved in "coordonate.txt" file. The shortest route algorithm is then used to determine what cities does the route pass through; they are then visually connected on the map by a line (*the line is formed by making use of the same coordinates that were used to display cities)* and thus the route is drawn on the map.

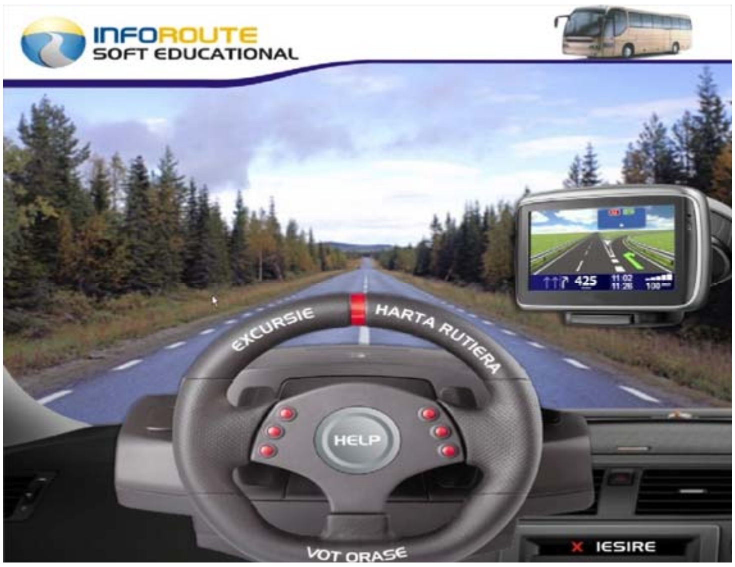
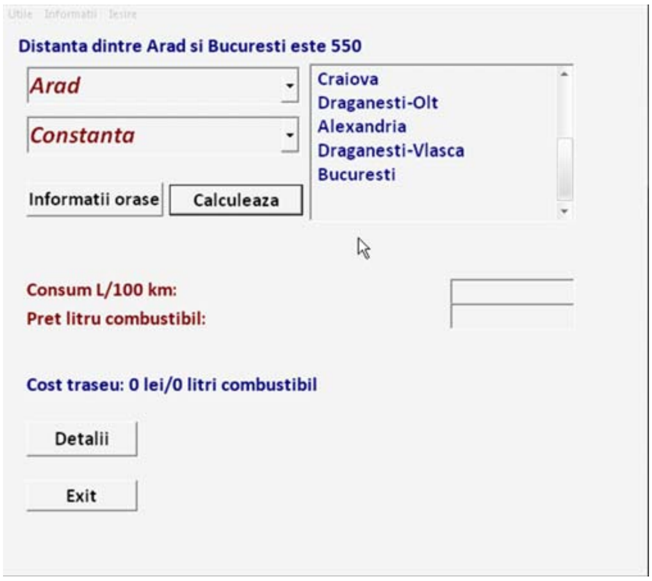


Fig. 1. Main Form.

```
val(edit1.Text,consum,ok);
 val(edit2.Text,pret,ok2);
 if(edit1.text='') then showmessage('!!!Atentie!!! Nu ati introdus consumul de combustibil la suta de kilometri')
  else if(edit2.text='') then showmessage('!!!Atentie!!! Nu ati introdus pretul combustibilului')
      else if(ok<>0) then showmessage('!!!Atentie!!! Ati introdus si litere in campul consumului de combustibil
la suta de kilometri')
             else if(ok2<>0) then showmessage('!!!Atentie!!! Ati introdus si litere in campul pretului
combustibilului')
                 else
                   begin
                      label1.Visible:=true;
                      label2.Visible:=true;
                      label5.Visible:=true;
                      ti:=round(distanta/55*3600);
                      label2.Caption:='Durata traseu: ~'+inttostr(ti div 3600)+' ore, '+inttostr(((ti mod 3600) div
60)+1)+' minute';
                      label5.Caption:='Cost traseu: '+formatfloat('0.00',distanta/100*consum*pret)+' lei /
'+formatfloat('0.00',distanta/100*consum)+' litri combustibil';
                   end;
```



Distanta dintre Arad si Bucuresti este 550

Arad

Constanta

Craiova
Draganesti-Olt
Alexandria
Draganesti-Vlasca
Bucuresti

Consum L/100 km:

Pret litru combustibil:

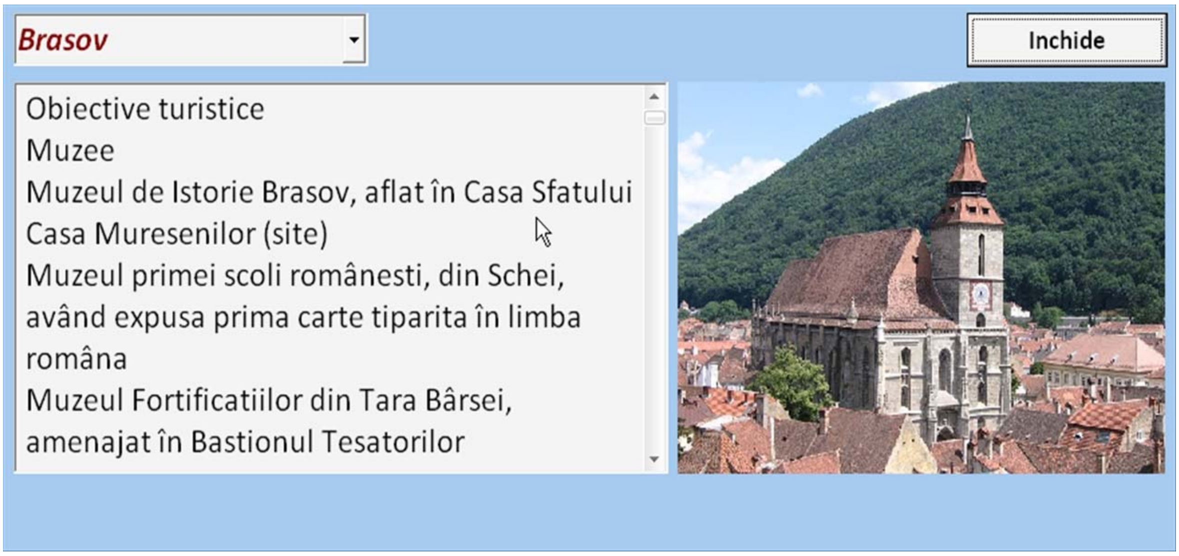
Cost traseu: 0 lei/0 litri combustibil

Fig. 2. Shortest Route.

2.3.2 “Informatii Orase” (*City Details*) function

Another important feature of this application is the “Informatii Orase” function. It provides the user access to text details and images regarding the biggest cities in Romania. Access is made by opening the “Excursie” window, then either clicking on the “Informatii Orase” button or using the “Informatii” taskbar menu and selecting “Informatii Orase”, followed by selecting the desired city choice.

```
If  FileExists('poze/'+combobox1.Text+'.jpg') then
    Form4.Image1.Picture.LoadFromFile('poze/'+combobox1.Text+'.jpg')
  else Form4.Image1.Picture.LoadFromFile('poze/poza.bmp');
memo1.Clear;
  if FileExists ('informatii/'+combobox1.Text+'.txt') then
    memo1.lines.LoadFromFile('informatii/'+combobox1.Text+'.txt')
  else memo1.lines.add('Nu exista informatii!');
```



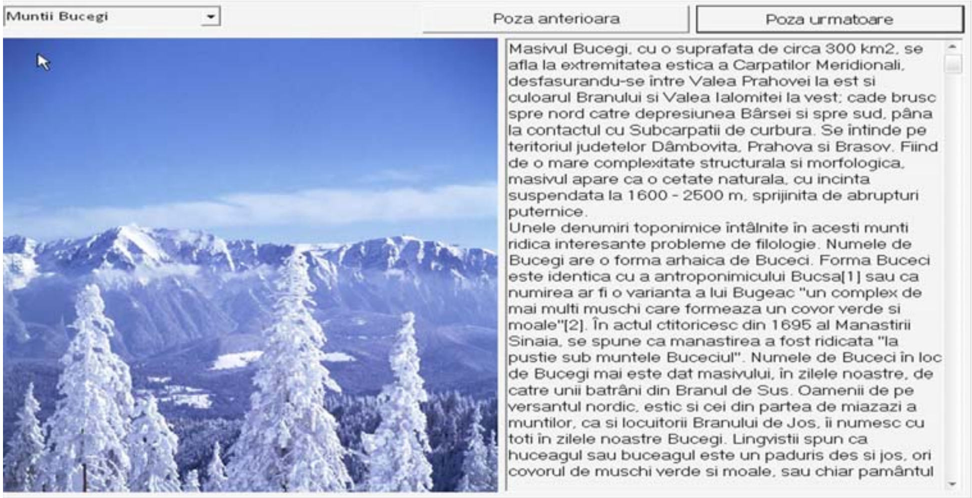
Obiective turistice
Muzee
Muzeul de Istorie Brasov, aflat în Casa Sfatului
Casa Muresenilor (site)
Muzeul primei scoli românesti, din Schei, având expusa prima carte tiparita în limba româna
Muzeul Fortificatiilor din Tara Bârsei, amenajat în Bastionul Tesatorilor

Fig. 3. *City Details.*

2.3.3 Function "Atractii Turistice" *(Tourist Attractions Details*)

While the "Informatii Orase" function provides information about the most important Romanian cities, the function "Atractii Turistice" (*Tourist Attractions Details*) is used to provide information about specific tourist attractions, both in text and images. To utilise this function, the user must click on the "Excursie" button, then click on the "Informatii" taskbar menu and select the "Atractii Turisitce" option.

```
i:=1;
if fileexists('zone turistice\'+combobox1.text+'.txt') then memo1.Lines.LoadFromFile('zone
turistice\'+combobox1.text+'.txt')
  else memo1.Lines.LoadFromFile('zone turistice\nue.txt');
  assignfile(g,'zone turistice\util.txt');
  rewrite(g);
  writeln(g,combobox1.text);
  closefile(g);
  if FileExists('zone turistice\'+combobox1.Text+inttostr(i)+'.jpg') then image1.Picture.LoadFromFile('zone
turistice\'+combobox1.Text+inttostr(i)+'.jpg');
```

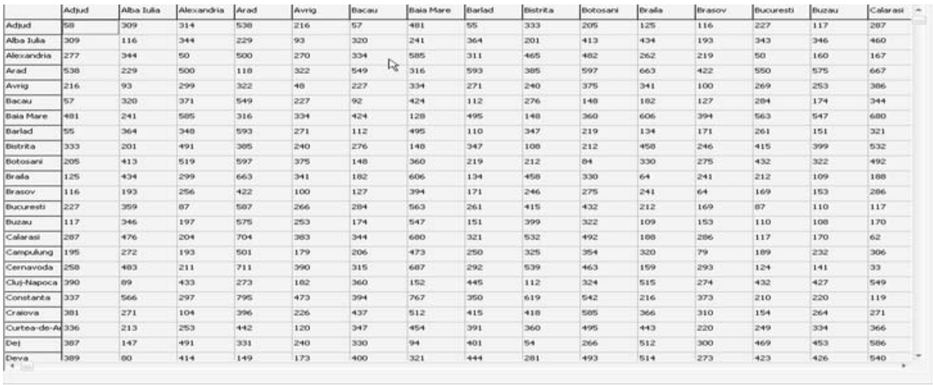


Masivul Bucegi, cu o suprafata de circa 300 km2, se afla la extremitatea estica a Carpatilor Meridionali, desfasurandu-se între Valea Prahovei la est si culoarul Branului si Valea Ialomitei la vest; cade brusc spre nord catre depresiunea Bârsei si spre sud, pâna la contactul cu Subcarpatii de curbura. Se întinde pe teritoriul judetelor Dâmbovita, Prahova si Brasov. Fiind de o mare complexitate structurala si morfologica, masivul apare ca o cetate naturala, cu incinta suspendata la 1600 - 2500 m, sprijinita de abrupturi puternice.
Unele denumiri toponimice întâlnite în acesti munti ridica interesante probleme de filologie. Numele de Bucegi are o forma arhaica de Buceci. Forma Buceci este identica cu a antroponimicului Bucsa[1] sau ca numirea ar fi o varianta a lui Bugeac "un complex de mai multi muschi care formeaza un covor verde si moale"[2]. În actul ctitoricesc din 1695 al Manastirii Sinaia, se spune ca manastirea a fost ridicata "la pustie sub muntele Buceciul". Numele de Buceci în loc de Bucegi mai este dat masivului, în zilele noastre, de catre unii batrâni din Branul de Sus. Oamenii de pe versantul nordic, estic si cei din partea de miazazi a muntilor, ca si locuitorii Branului de Jos, îi numesc cu toti în zilele noastre Bucegi. Lingvistii spun ca huceagul sau buceagul este un paduris des si jos, ori covorul de muschi verde si moale, sau chiar pamântul

Fig. 4. Tourist Attractions

2.3.4 “Distante Intre Orase” (*Distance between Cities)* Function

When the user intends to know the distance between two cities without selecting them as departure and destination points, he may apply “Distante Intre Orase” function from the “Utile” taskbar menu. The opened window will provide tabular information about the distance between all available cities. This table is generated by the software using a matrix which loads from memory and which is displayed in a .txt file. All inter-city distances shown are calculated using Dijkstra's algorithm to help determine absolute minimum distance.



	Adjud	Alba Iulia	Alexandria	Arad	Avrig	Bacau	Baia Mare	Barlad	Bistrita	Botosani	Braila	Brasov	Bucuresti	Buzau	Calarasi
Adjud	58	309	314	538	216	57	481	55	333	205	125	116	227	117	287
Alba Iulia	309	116	344	229	93	320	241	364	201	413	434	193	343	346	460
Alexandria	277	344	50	500	270	334	585	311	465	482	262	219	50	160	167
Arad	538	229	500	118	322	549	316	593	385	597	663	422	550	575	667
Avrig	216	93	299	322	48	227	334	271	240	375	341	100	269	253	386
Bacau	57	320	371	549	227	92	424	112	276	148	182	127	284	174	344
Baia Mare	481	241	585	316	334	424	128	495	148	360	606	394	563	547	680
Barlad	55	364	348	593	271	112	495	110	347	219	134	171	261	151	321
Bistrita	333	201	491	385	240	276	148	347	108	212	458	246	415	399	532
Botosani	205	413	519	597	375	148	360	219	212	84	330	275	432	322	492
Braila	125	434	299	663	341	182	606	134	458	330	64	241	212	109	188
Brasov	116	193	256	422	100	127	394	171	246	275	241	64	169	153	286
Bucuresti	227	359	87	587	266	284	563	261	415	432	212	169	87	110	117
Buzau	117	346	197	575	253	174	547	151	399	322	109	153	110	108	170
Calarasi	287	476	204	704	383	344	680	321	532	492	188	286	117	170	62
Campulung	195	272	193	501	179	206	473	250	325	354	320	79	189	232	306
Cernavoda	258	483	211	711	390	315	687	292	539	463	159	293	124	141	33
Cluj-Napoca	390	89	433	273	182	360	152	445	112	324	515	274	432	427	549
Constanta	337	566	297	795	473	394	767	350	619	542	216	373	210	220	119
Craiova	381	271	104	396	226	437	512	415	418	585	366	310	154	264	271
Curtea-de-A	336	213	253	442	120	347	454	391	360	495	443	220	249	334	366
Dej	387	147	491	331	240	330	94	401	54	266	512	300	469	453	586
Deva	389	80	414	149	173	400	321	444	281	493	514	273	423	426	540

Fig. 5. Distance between cities.

```
procedure TForm9.StringGrid1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
var f:textfile;
    i,j:integer;
    a:array[1..100,1..100] of longint;
    s:string;
begin
     assignfile(f, 'orase.txt'); reset(f);
   for i:=1 to 63 do
      begin
        readln(f,s);
        stringGrid1.Cells[0,i]:=s;
      end;
   closefile(f);
   assignfile(f, 'orase.txt'); reset(f);
   for i:=1 to 63 do
      begin
        readln(f,s);
        stringGrid1.Cells[i,0]:=s;
      end;
   closefile(f);

   assignfile(f, 'drumuri.out'); reset(f);
   for i:=1 to 63 do
    begin
      for j:=1 to 63 do
        read(f,a[i,j]);
      readln(f);
    end;
    closefile(f);

   for i:=1 to 63 do
    for j:=1 to 63 do
      stringGrid1.Cells[j,i]:=inttostr(a[i,j]);

end;
```

2.3.5 “Votare Oras” *(City Rating)* Function

The software has a scoring system which allows the user to individually attribute a mark to each city that was part of the latest route input. The software then uses these marks to calculate a different rating for each certain route. This function is accessed by clicking on the "Vot Orase" (*City rating)* button in the main menu *(however, the user must first select a route in the "Excursii" menu).* The newly opened window allows the user to individually rate each city by his/her personal preference. City marks are saved in .txt files corresponding to the city names. In the "Excursie" window, these city marks are used to calculate and display the average rating by a 1-10 star system.

2.3.6 “Harta Rutiera” (*Maps*) Section

This section, accessed by selecting the “Harta Rutiera” option from the main window, is dedicated entirely to the maps a user might make use of when he is establishing his itinerary. All maps have the “zoom level” button, which allows to decrease/increase the zoom. The type of the map which is currently on the display can be changed from the “Vizualizare” (*View*) taskbar menu.

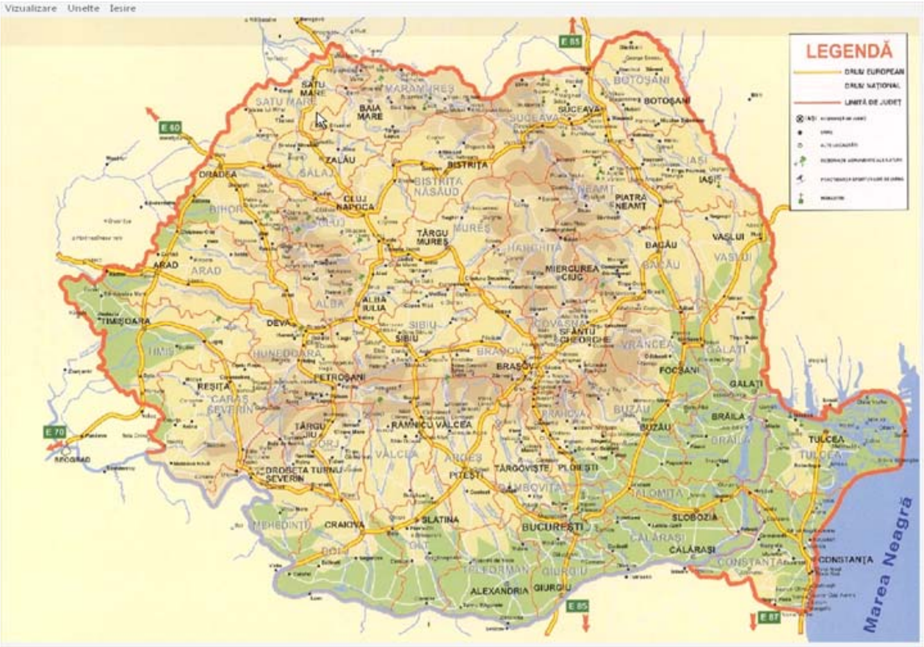


Fig. 6. Map.

2.3.7 User Help

Each of the functions listed above has a *help* option which explains its functionality and how it should be used. Help is available in both audio and written format for certain functions. To activate the audio help the user has to click on “Activare Ajutor Vocal” (*Enable Audio Assistant*) from the “Unelte” (*Tools*) taskbar menu.

```
procedure TForm2.Activareajutorvocal1Click(Sender: TObject);
begin
  MediaPlayer1.Play;
end;
```

```
procedure TForm2.Dezactivareajutorvocal1Click(Sender: TObject);
begin
  MediaPlayer1.Stop;
end;

procedure TForm2.FormCreate(Sender: TObject);
begin
   memo1.Lines.LoadFromFile('help\ajutor1.txt');
   mediaplayer1.FileName:=getcurrentdir+'\help\helpadauga.mp3'; mediaplayer1.Open;
end;

procedure TForm2.Iesire2Click(Sender: TObject);
begin
  close;
end;
```

3 Technical aspects

This application has been developed using the Delphi programming language. On a global scale, the software makes use of multiple functions, each function using a certain algorithm. The most interesting and complex of these is, by far, the one which determines the shortest distance between two cities; this algorithm was written using the graph theory. Thus, the entire software could be viewed, at its core, as a graph-based application.

Here is a breakdown of the main algorithm:

```
Start:
      n,c,x0 are used as input variables  //n-number of nodes, c-adjacency matrix
    m={x0}
    d[x]=c[x0][x]
    pre[x]=x0, for x=1,2,3...,3, x≠x0
    pre[x0]=0
Cycling n-1 times:
    A previously unselected graph node (x)  is determined:
        d[x]=min{d[y]| Y is not included in M}
    M=M U {x} //reunion between M and {x}
    The distances between node x and the other unselected nodes are updated:
        if d[y]>d[x]+c[x][y] then
            d[y]=d[x]+c[x][y]
            pre[y]=x
```

4 Conclusions and future developments

This application was based on well known and well studied algorithms, using a readily-available technology. As a consequence, it could be considered a graph-based application, yet it is also something much more original, as it uses knowledge from multiple algorithms, technologies and programming languages.

The data used in this application was obtained by carefully studying the Romanian road map and it’s respective map legend. Each city was attributed an individual ID and has had it’s distance to other cities stored in a data structure. Using these data structures we created the adjacency list which constitues the graph's nodes.

The interface, which helps the user make full use of the sofware's capbilities, was thought out to be as organised and intuitive as possible. The main menu image was created using Adobe Photoshop, while the other menu background images make use of buttons present in the programming medium/language.

The application illustrated in this paper represents the first part of what we plan to be a much larger touristic project. The software can be upgraded in the future, especially taking into consideration the volatile and ever changing nature of the tourism business. The main improvements set our goals on defining a collective database, easily accessible to any user (in hopes of a better city notes/information management) and a more efficient data access algorithm for the application itself. We would also like to add more data and up to date information regarding the existing tourist attractions, as well as adding new ones.

5 References

[1] Cerchez,Emanuela and Marinel Serban, *Programarea in limbajul C/C++ pentru liceu*, Polirom Iasi, 2006

[2] Elena Toader and Camelia Berceanu, Initiere in programarea pe obiecte, Editura Arves, Craiova, 2003

[3] Tudor Sorin, Initiere in programarea Vizuala, *L&S Info-mat*, Bucuresti, 2004

[4] http://maps.google.com/maps?q=romania&hl=ro&ie=UTF8&sll=37.0625,-95.677068&sspn=57.161276,79.013672&hnear=Rom%C3%A2nia&t=m&z=7

[5] http://en.wikipedia.org/wiki/Embarcadero_Delphi

Aldulea Cristian-Ionel
Transilvania University of Brasov
Departament of Mathematics and Informatics
Str.Iuliu Maniu Nr. 50 Brasov , 500091
ROMANIA
alduleacristi@yahoo.com

Fieraru Bogdan Mihai
Transilvania University of Brasov
Departament of Mathematics and Informatics
Str.Iuliu Maniu Nr. 50 Brasov , 500091
ROMANIA
fierarubogdan2000@yahoo.com

Vasile Vlad
Transilvania University of Brasov
Departament of Mathematics and Informatics
Str.Iuliu Maniu Nr. 50 Brasov , 500091
ROMANIA
vlad.vasile91@yahoo.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Psihological tests

Ioana Madalina Anghel
Petru Barko

Teacher Coordinator: Anca Vasilescu

Abstract

Our software application consists of some psychological tests, both for adults and for children that have to go to school. The practical aspects of our work are concerning the “zero class” – the recent Romanian problem of the children transition from kindergarten to school. The specific tests help the user to understand more about himself and they offer him a characterization of his personality. As final data, our application provides an information synthesis and the distribution of each child in school or kindergarten following the passed tests results.

The program is implemented in Microsoft Visual FoxPro 9 and it is based on a main form that allows the user to find out more details about the tests, about how to use the software and about other users opinions concerning each test. One can add comments and new tests or just fulfil the default tests.

1 Introduction

Our project consists of a software application that allows the user to go through two sets of psychological tests.

One of these sets contains six tests which help us to understand more about ourselves and offer us a characterization of our personality. We think it is important to know ourselves enough in order to try to change our weaknesses and to highlight our qualities.

The second set of tests has a relevant issue, meaning the tests for “zero grade” at school. This part can help the psychologists to test the children in a more pleasant way for them, because the program has an attractive interface that makes children to perceive and to follow the test like a game.

We chose this subject because we were always interested in psychology and in finding out more about ourselves and the people we interact with.

Our intention was to develop our own specific applications which could help the psychologists, the children and the people who want to find out more about their character and personality.

2 More details about the application

2.1 About Microsoft Visual FoxPro9

Visual FoxPro is a data-centric object-oriented and procedural programming language produced by Microsoft. It is derived from FoxPro (originally known as FoxBASE) which was developed by Fox Software starting with 1984.
VFP9 allows visual design of databases, queries and forms. Visual programming facilitates to obtain the graphics and to build a “user-friendly” interface. Any modern, commercial software is based on visual and object oriented design .

2.1.1 Operating system compatibility and information on executable files

VFP9 is compatible with the following operating systems: Windows 2000, Windows XP, Windows server 2003 and Windows Vista and Windows 7.
Someone who want to install VFP9 have to know the some information about the executable files, such as :

- EXE Size : 5,620 kb
- DLL Size : 4,600 kb

2.2 The application structure

Our software contains a main form that allows the user to complete the desired tests, to add new ones or to read information about the tests.
The application structure is the following:

- Main page, that contains the next forms:
    - How to use the tests
    - Add tests
    - Comments and opinions about tests
    - Psychological tests, where there are included the following forms:
        - Comments and opinions about tests
        - Add tests
        - Information about the test
        - Tests
    - Tests for children , with the next forms:
        - Comments and opinions about tests
        - Information about the tests
        - Tests
        - Centralization of information
        - Distribution of each child in school / kindergarten

2.3 The implementation

2.3.1 How the software works

The program is implemented in Microsoft Visual FoxPro 9 and it is based on a main form that allows the user to find out more details about the tests, about how to use the software and about other users opinions concerning each test. One can add comments and new tests or just fulfil the default tests.
Each test corresponds to a table that contains some fields where the questions, the possible answers and the score for each answer are introduced.
While the user answers the questions, the software calculates the score of each question and, according to the final result, it displays the characterization of the user’s personality. If we talk

about the tests for “zero grade”, the program will tell if the child passed the test or not and if he is prepared for school or for kindergarten.
Each test has the following options:

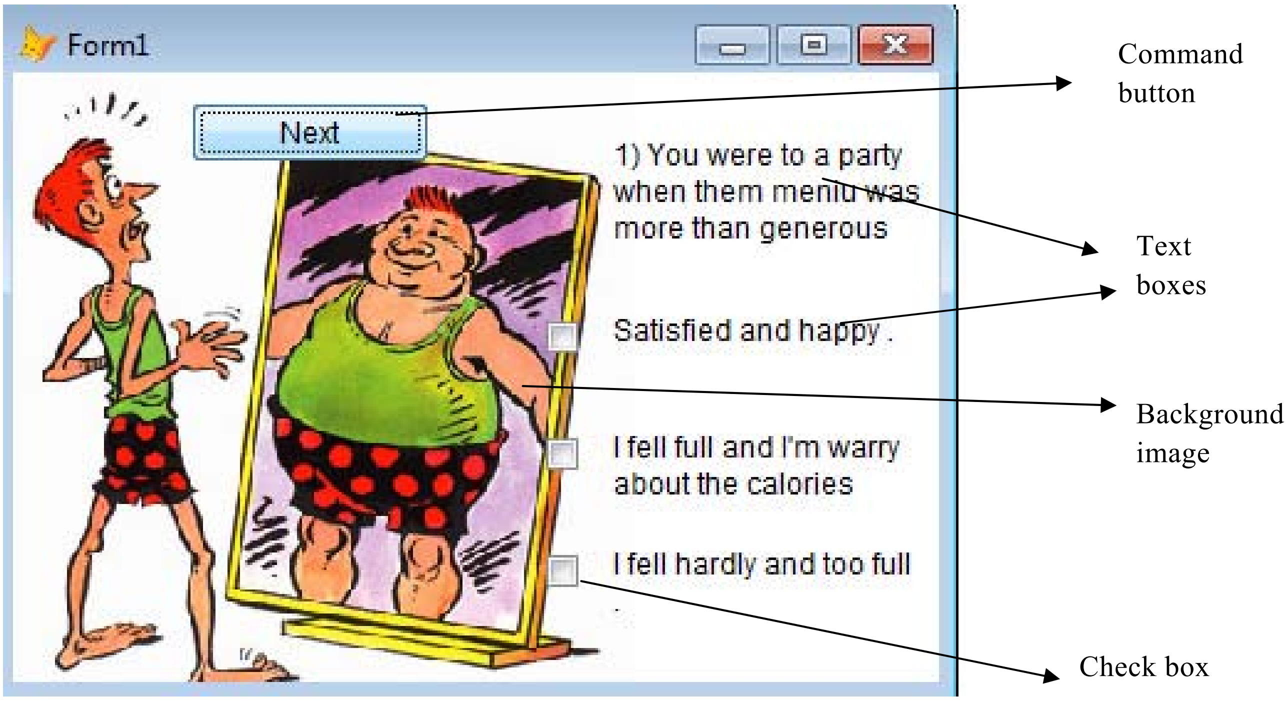
- progress to the next question
- skip or go back to any question
- choose the desired answer
- get out of the test
- solve another test

2.3.2 The graphic implementation

For the graphic interface, we used:

- labels – for the “fixed text”
- text boxes – for the texts that is changing
- command buttons – so that the user can go to the next question or get out of the test
- check boxes – so that the user can check the desired answer.

We also used background images and pictures which are changing accordingly to the final result.



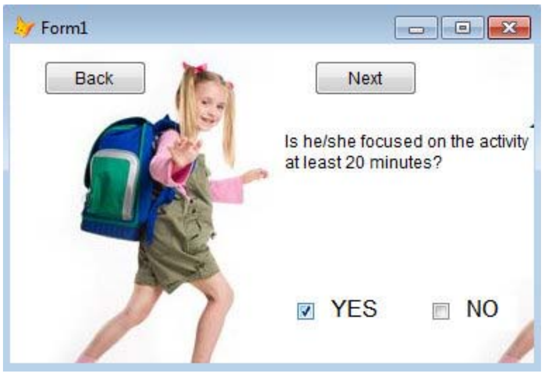
1) You were to a party when them meniu was more than generous

☐ Satisfied and happy .

☐ I fell full and I'm warry about the calories

☐ I fell hardly and too full

Fig.1. An item with multiple possibilities of choice

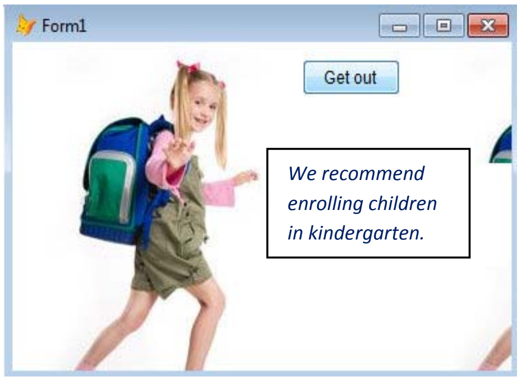


Is he/she focused on the activity at least 20 minutes?

☑ YES ☐ NO

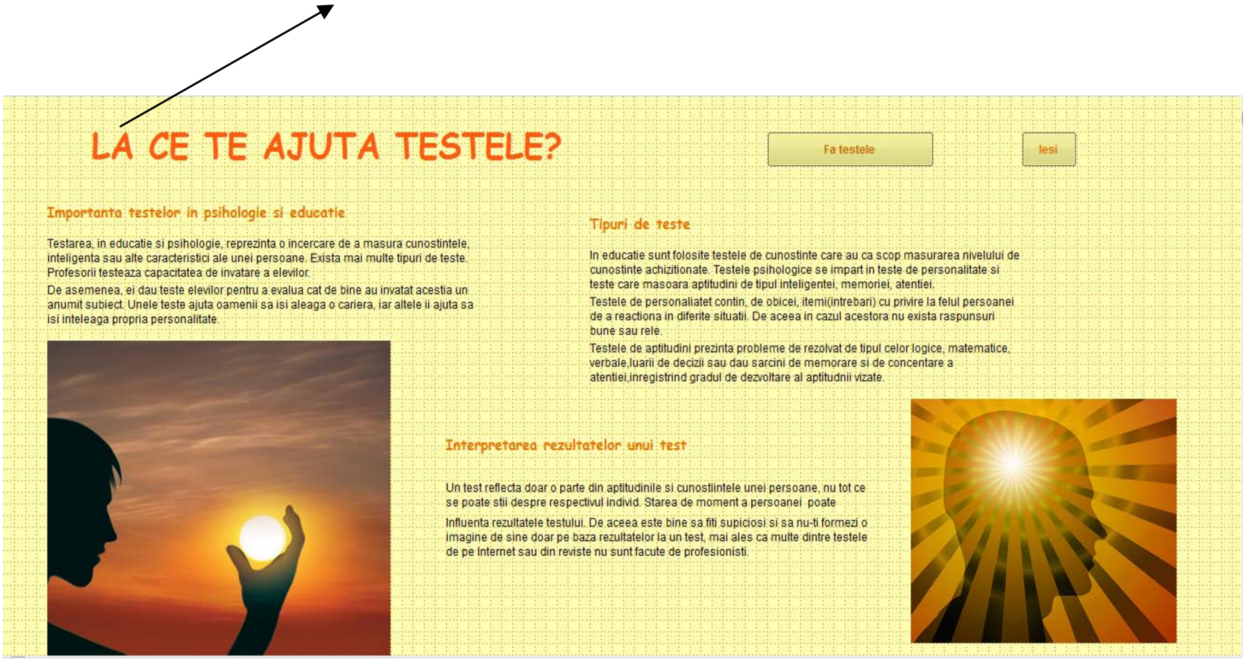
Fig.2. An item with only two possibilities of choice – yes or no

Fig.3. The final result



We recommend enrolling children in kindergarten.

Label



LA CE TE AJUTA TESTELE?

Importanta testelor in psihologie si educatie

Testarea, in educatie si psihologie, reprezinta o incercare de a masura cunostintele, inteligenta sau alte caracteristici ale unei persoane. Exista mai multe tipuri de teste. Profesorii testeaza capacitatea de invatare a elevilor.

De asemenea, ei dau teste elevilor pentru a evalua cat de bine au invatat acestia un anumit subiect. Unele teste ajuta oamenii sa isi aleaga o cariera, iar altele ii ajuta sa isi inteleaga propria personalitate.

Tipuri de teste

In educatie sunt folosite testele de cunostinte care au ca scop masurarea nivelului de cunostinte achizitionate. Testele psihologice se impart in teste de personalitate si teste care masoara aptitudini de tipul inteligentei, memoriei, atentiei.

Testele de personaliatet contin, de obicei, itemi(intrebari) cu privire la felul persoanei de a reactiona in diferite situatii. De aceea in cazul acestora nu exista raspunsuri bune sau rele.

Testele de aptitudini prezinta probleme de rezolvat de tipul celor logice, matematice, verbale,luarii de decizii sau dau sarcini de memorare si de concentare a atentiei,inregistrind gradul de dezvoltare al aptitudnii vizate.

Interpretarea rezultatelor unui test

Un test reflecta doar o parte din aptitudinile si cunostiintele unei persoane, nu tot ce se poate stii despre respectivul individ. Starea de moment a persoanei poate

Influenta rezultatele testului. De aceea este bine sa fiti supiciosi si sa nu-ti formezi o imagine de sine doar pe baza rezultatelor la un test, mai ales ca multe dintre testele de pe Internet sau din reviste nu sunt facute de profesionisti.

Fig.4. The form with information about the psychological tests

3 The code

In the form, we have to declare all the variables used in the command button and to make the contact with the table where the questions and answers are.

```
use table1 in 1
PUBLIC i,n
i=1
n=RECCOUNT()
PUBLIC s
s=0
```

If the user didn't finish answering all the questions, the command button("Next") executes the following operations:

❖ The variable i changes. It takes the value of the number of the next question and the program skips to that question in the table.

```
i = recno()
go i
The questions and answers are added in the text boxes
thisform.text1.value=intreb
thisform.text2.Value=rasp1
thisform.text3.Value=rasp2
thisform.text4.value=rasp3
```

Intreb, rasp1, rasp2 and rasp3 are the fields of the table and they contain the questions and the possible answers.

❖ The score of the chosen answer is added to amount s

```
if thisform.check1.value = 1
    s=s+r1
endif
if thisform.check2.value = 1
    s=s+r2
endif
if thisform.check3.Value = 1
    s=s+r3
endif
The page is refreshed and the check boxes are set to 0
thisform.Refresh
thisform.check1.Value=0
thisform.check2.value=0
thisform.check3.Value=0
```

Otherwise, if the user answered all the questions, text boxes ,check boxes and command button ("Next") become invisible and "Get out" command button becomes active.

```
thisform.check1.Visible=.f.
thisform.check2.Visible=.f.
thisform.check3.Visible=.f.
thisform.text1.Visible=.f.
thisform.text2.Visible=.f.
thisform.text3.Visible=.f.
thisform.text4.Visible=.f.
thisform.text5.Visible=.t.
thisform.command1.Visible=.f.
thisform.command2.Visible=.t.
```

After that, the software displays the final response, according to the score.

```
if s<=9
  thisform.text5.value='You are very critical with yourself. You often find a blemish and you're not satisfied with yourself. Try to do more often something good for you. You'll fell better with yourself.'
endif
if s>=10 AND s<=14
    thisform.text5.value='You know which are your bad points, but these don't stop you being in harmony with yourself and with your purposes. You understood you have only to win if you are satisfied with yourself.'
Endif
```

The picture can also change according to the score and the back colour too.

```
    thisform.image1.Visible=.f.
    thisform.BackColor=RGB(255,128,128)
```

The edit box allows the user to skip or go back to any question he wants. Its function is similar to the “Next” button. The difference is the value received by the variable i.

```
    i=VAL(thisform.edit1.Value)
```

The “Get out” button functions with the command :

```
    Thisform . Release
```

4. Conclusions and further developments

First of all, our software application contains two sets of psychological tests that offer the users the possibility to better understand their own character. These can help them to develop their personality, to change the bed points and to highlight the good points of their behaviour.

On the other hand, the tests offer the psychologists a better solution for testing the children for “0” grade. Because of their pleasant interface, the children perceive the tests like a game.

We have developed this software in terms of adding tests applicable in more areas (educational tests , tests for obtaining driving license and others). On the other hand, we shall also create a webpage where all our tests and their results will could be accessed by any internet user and we intend to find more potential developments for our project.

Reference

[1] Mariana Panțiru, Ionuț Panțiru and Irina Ioana Panțiru, *Informatică – Handbook for XII grade – Version Visual Fox Pro*, L&S INFOMAT Publishing, 2003

[2] Marcel Andrei Homorodean and Simona Petrescu, *Informatică – Handbook for XII grade*, Niculescu Publishing, 2001

[3] Horst H. Siewert, Teste de personalitate, Gemma Print Publishing, 2001

[4] ***, www. testepsihologice.org

[5] http://en.wikipedia.org/wiki/Visual_FoxPro

Ioana Madalina Anghel
Transilvania Universitry of Brasov
Depatment of Mathematics and Informatics
Iuliu Maniu nr. 50
BRASOV
E-mail: madalina.i_anghel@yahoo.com

Petru Barko
Transilvania Universitry of Brasov
Department of Mathematics and Informatics
Iuliu Maniu nr. 50
BRASOV
E-mail: petrubarko@yahoo.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

RevontuLED

Ana Barar
Teacher Coordinator: Alexandru Radovici

Abstract

This project is an attempt to exploit the microcontroller technology in order to create a complex light show. RevontuLED is a dynamic picture simulating the northern lights. It consists of a wooden box with a background and a series of RGB LEDs on its lower base. We will program an Arduino microcontroller board and wire up the LEDs to it. The code we will upload on the microcontroller will change the colours of the RGBs, modify their light intensity and even impose the order and the speed with which the LEDs wil turn on and off. The code will be written in the Arduino soft, which syntax is similar to C++. All these aspects, as well as some future improvements will be presented in greater detail in the sections to come.

1 Introduction

Microntroller technology has been developped since 1971 at an amazing speed. It has enabled engineers to design and create automatically controlled devices, such as implantable medical devices, remote controls, power tools, even toys. It contributed with an enormous boost in the progress of technology.

All this may sound nice, but what actually is a microcontroller? It is an integrated circuit which has whatever a mini computer would need: processor core, memory, i/o peripherals. This device is designed to reduce the costs of the creation of such products.They are a very efficient and economical form of digital control over the devices they are embedded in.

As far as this project is concerned, microcontrollers will play an essential part. Basically, the project consists of a picture animated by a series of RGB LEDs. These LEDs will simulate the northern lights. In order for the LEDs to be turned on with the right intensity and colour, our microcontroller will be programmed accordingly. In the further sections of this chapter, we shall present the project in greater detail, explaining what it consists of and what it does, which were the materials and softwares we have used to build it. We will also bullet-point some of the improvements we are planning to bring up and we will close this paper by summing everything up in a set of conclusions.[1]

2 What is the project about?

Basically, the project is a picture animated by RGB LEDs, which will simulate the northern lights. We will use a wooden box with a glass window. On the lower base of the box we will assemble the LED rows in such manner that they will light the background sky. By programming the microcontroller properly, we will make the LEDs simulate the movements and the colours of the aurora. Eventually, we will install some RGB LEDs in the background to imitate the stars. The window glass will have a northern forest landscape painted on it. Thus, the result will be a lovely and realistic representation of a night lit by auroras of various colours. The effect of such a picture is appeasing, yet impressive and fascinating.

The unpredictible passing from a lively colour to another keeps the viewer attentive without stressing him/her. All in all, it will be a mini northern lights show, which will be both relaxing and surprising at the same time.

The hardware part of the project is fairly simple, as it has been previously explained. The wooden box will have 30x40 dimensions, with 10 cm in depth. The LED rows will have a certain number of RGBs soldered on a board, which will be placed on the lower base of the box. We preffer placing the LEDs thus instead of making an entire LED wall backround, because that positioning of RGBs will create the effect with greater likeness to the real phenomena. Having a dull LED matrix placed on the background of the box would only make a plain and non fluent image of the phenomena. It would be non fluent because the LEDs are not small enough to reproduce the image with a good resolution. These LEDs will be all wired up to an Arduino microcontroller board. What Arduino is, how does it work and the reason why we have chosen it are topics which we will discuss in the "Materials" section, since it has a longer explanation to it and it is linked to the software part.

We have sketched the hardware aspect of the project, so now we shall focus on the software part, which is more complex. We will be using the Arduino software to elaborate our microcontroller's code. The LEDs will be set on the PWM(power modulation) pins (see section 3), which means we will be able to write both analog and digital values on them. We will enable the microcontroller to change the RGBs' colours periodically and also to change the speed to which the "colour-bands" move. Eventually, we will add an interactive aspect to the project, allowing the user to change colour and speed to his liking.[4] [2]

3 Materials

The main piece of this project is, obviously, the microcontroller. Our microcontroller board is, by choice, Arduino UNO. It has an ATmega328 microcontroller integrated on a board which also has 14 digital i/o pins, out of which 6 can be used as analog too (PWM), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header and a reset button. It also has 32KB memory (out of which 0,5 KB are used for the bootloader), 2KB SRAM and 1KB EEPROM. The reasons why we have chosen Arduino UNO: it is quite simple to use and it suits the project perfectly because of that.

Given the number of LEDs we will be using and the number of PWM pins we have at our disposal on our Arduino, it is only natural for this question to occur: how are we going to get all those LEDs connected? The answer is: shift register. These will not only allow us to get all those LEDs wired up, but they'll also simplify the programming task.

The code will be written in the Arduino soft. The programming language is a simplified version of C++, so the syntax will not cause any problems. When we will make the project interactive, we will use the serial port communication, which will enable us to send and receive data from our Arduino. Thus, we will be able to communicate the right voltages to the RGBs in order to get the desired colour, or to modify the speed to which the colours will change. Not to worry! the user will not be choosing voltages for the red, green and blue LEDs. There will be some colours already defined by us and given at his disposal to choose from. [3] [5]

4 Future work

Because a good piece of work is never really finished, we consider adding more aspects to this project. Besides making it interactive by enabling serial port communication through the Arduino port and adding a series of buttons for the viewer to use, weare also intending to add music to the picture. Basically, we will make colours sing. Everytime a colour will show up, an associated sound will come along. This part of the project will be possible with another board, called Music Shield. It is a professional audio codec, compatible with our Arduino. It decodes WMA, MP3, FLAC, WAV, MIDI, Ogg Vorbis formats, it has two control-push buttons, one knob swith, it is able to play music from SD cards. It also has I2S interface for external DAC, a line in for OGG format recording and a great sound quality (90 dB dynamic range for playback). Music Shield will be added as an extension to our Arduino.

We would also like to go further with the light shows. This means our picture will also be able to

simulate sunrises, sunsets, rainbows, and other sky phenomena. This means we will have to change our black background to a cameleon one, which will simulate day, night and different weather conditions. This also means that additional LEDs might be required, as well as some modifications brought to the landscape, just to make it adjustable to all those weather changes. Our code will be modified, too. We'll have to find a way to "encode" all these phenomena.
In the end, we wish to have a picture which can simulate almost any weather condition and which can give the image of a northern sky and the way it looks throughout the year.

5 Conclusion

In a nutshell, this project is bound to create a realistic, yet impressive effect, thanks to which the sky will become a piece of reality closed in a box. The sky has been a fascination and an endless source of inspiration for artists such as Turner or Monnet, inspiration which gave birth to breathtaking paintings. Why shouldn't we try to digitally paint the sky? Not only will it immortalize the moment, but it will allow us to relive it, but never with the same impression.
The beauty of such views can be expressed on more than one level. This project will give us the freedom to recreate the moment both visually and musically at the same time. It will keep it real, yet it will make you feel that the magic of the northern lights exists.

References

[1] Brian Evans. *Beginning Arduino Programming.* Apress, 2011.

[2] Hugh Blemings Jonathan Oxer. *Practical Arduino- Cool Projects for Open Source Hardware.* Apress, 2009.

[3] Michael McRoberts. *Beginning Arduino.* Apress, 2010.

[4] Mike McRoberts. *Arduino Starter Kit Manual- A Complete Guide to the Arduino.* Earthshine Design, 2009.

[5] Harold Timmis. *Practical Arduino Engineering.* Apress, 2011.

Ana Barar
"Politehnica" University of Bucharest
FILS
Splaiul Independentei 313, Bucuresti, 060042, Corp JE 105
ROMANIA
c.luthien@yahoo.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

A language for teaching kids programming

Ioana Alexandra Bărbătei
Teacher Coordinator: Alexandru Radovici

Abstract

Since HTML5 brought us so many possibilities to build complex applications, offering an alternative to Flash, I chose to use it in order to make my ideas become reality. Together with Canvas and JavaScript, HTML5 helped me create a project wich has the purpose to help children learn programming, because my application has similar principles with the Logo computer programming language. The children can draw or play but for that they must use commands in order to move their character, so if they want to succed in their plans they have to build a strategy and think about the way they can get there, using the available commands.

1 Introduction

HTML5 is a very useful platform both for the Web developers and for the developers who use for example Microsoft tools and technologies. This fifth revision of the HTML standard introduces markup and application programming interfaces for complex web applications. That's why HTML5 represents a potential candidate for cross-platform mobile applications.

A very interesting thing is that HTML5 adds many new syntactical features such as <video>, <audio> and <canvas> elements. All these features were created to facilitate the action of including and using multimedia and graphical content on the web without needing to resort to proprietary plugins and application programming interfaces. In the same time, HTML5 provides rich animations and interactivity, due to the capability of using Java Script together with the new HTML canvas tag for drawing of 2D and 3D objects, CSS3 animation and transformation capabilities, and also the browser support for Scalable Vector Graphics.

Another important thing is that HTML5 became a candidat to replace Flash: games, apps and other things Flash is famous for can be done as well in HTML5. There are many advantages in using HTML5 : it's an open standard, it can be accesed by everywhere, it runs in the browser, it's better supported and faster.

My project consists of a software that helps the children understand and learn what programming means. They have acces to a friendly interface, they can choose their favorite character that will be later manipulated through specific commands to draw different forms or to find the right way to get out of a labyrinth. The goal of my project is not only to entertain, but also to make the children build a strategy and think about the stages through which they need to pass to succed. That's why, in fact, this application is a challenge for them to use their creativity and to find a solution using the available commands.

2 My program

The purpose of my project is the implementation of a language that should be similar with the Logo one but has some advantages: there are more characters, the children will choose between them and it is developed on the browser, so it's more accessible.

The character has a pen that can be lifted up (and there will be no trace left behind), or not (in this case, the character will leave a trace through all the points he will pass by).

The program will wait for the instructions given by the user. The instructions must contain a command (for example "forward") and parameters (the distance in pixels or the mesure in degrees of the angle of rotation). If the child wants to draw, then he must introduce the instruction "pen down". Otherwise, the instruction should be "pen up".

The kids can also change the color while drawing, by giving the command "pen color" followed by the name of the color they chose.

Another interesting command is "go" (or "goto","jump","jmp") followed by two parameters, representing the coordinates of the point where the user wants to move his character. This instruction will change the current position of the character to the specified new position, without changing the current value of the angle of rotation.

The children can also choose to play with their character and find a way to get out of a labyrinth, trying to make the correct steps, indicated by giving the correct instructions.

3 Related Work

As a related work, I used the Logo multi-paradigm computer programming language used in education. It represents an adaptation of the Lisp language and was conceived and written as a functional programming language for educational use (for constructivist teaching) because it can be used to teach computer science concepts. It also has facilities for handling lists, files, I/O, and recursion. Its intellectual roots are in artificial intelligence, mathematical logic, developmental psychology. The goal of the first implementation of Logo, called Ghost, was to create a math land where kids could play with words and sentences.

Logo's best-known feature is the turtle, which is an on-screen cursor receiving movement and drawing instructions, and is used to programmatically produce line graphics. So Logo is remembered today mainly for its turtle graphics: Turtle graphics were added to the Logo language by Seymour Papert to support a simple robot controlled from the user's workstation, designed to carry out the drawing functions assigned to it, using a small retractable pen.

The use of turtle geometry instead of a more traditional model mimics the actual movement logic of the turtle robot. Turtle geometry is operating in a Euclidean space so that the turtle moves with commands that are relative to its own position

Turtle geometry is also used as an alternative to a strictly coordinate-addressed graphics system (for example, the Lindenmayer system for generating fractals).

4 Technologies

Unlike the Logo Programming Language, my application is developed on the browser, so it can be easily accesed by anyone, anytime. In order to create my project, I used the HTML5 platform and the JavaScript language.

JavaScript is a prototype-based scripting language, dynamic and weakly typed. It has first-class functions and it supports object-oriented, imperative, and functional programming styles. JavaScript is primarily used in the form of client-side JavaScript, implemented as part of a Web browser in order to provide enhanced user interfaces and dynamic websites. JavaScript copies many names and naming conventions from Java, although the two languages are unrelated and have very different semantics. The most common use of JavaScript is to write functions that are embedded in or included from HTML pages and that interact with the Document Object Model (DOM) of the page.

I used the JavaScript language in my program in order to implement all the functions which make the character move or draw. But the animation would not be possible without the <canvas> HTML5 element, that allows for dynamic, scriptable rendering of 2D shapes and bitmap images. Canvas consists of a drawable region defined in HTML code with height and width attributes. JavaScript code have access to the area through a full set of drawing functions Some anticipated uses of canvas include building graphs, animations, games, and image composition.

5 Conclusions

Using the HTML5 platform, along with the JavaScript language and the <canvas> element, we can create a lot of animations or games without being forced to choose Flash. This way, many useful applications like this game that is a correspondent to the Logo's Turtle Graphics can be accesed very easy by everyone, anytime.

That's one of the reasons why, although nowadays Flash still the most used platform, in the future, his place will be taken by the HTML5 platform.

6 Future Work

As a future development, I want to upgrade my project and offer the children the possibility to play in an interactive way, to draw together with their friends, in order to challenge them to be both creative and clever and to have fun in the same time.

References

[1] Rob Hawkes, *Foundation-HTML Canvas For Games and Entertainment*

[2] Michael Morrison, *Head First JavaScript,* O'Reilly Media, United States of America, 2008

[3] Richard Wagner and Richard Mansfield, *Creating Web Pages All-in-One Desk Reference For Dummies 3rd Edition,* Wiley Publishing, Canada, 2007

[4] https://developer.mozilla.org/en/Canvas_tutorial

[5] http://www.w3schools.com/html5/default.asp

[6] http://www.w3schools.com/js/

[7] http://mckoss.com/logo/

Ioana Alexandra BĂRBĂTEI
Polytechnic University of Bucharest
Department of Engineering in foreign languages
Splaiul Independentei, 313, sector 6, Bucharest
ROMANIA
ioanalxndra@yahoo.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Machine learning in gear changing

Alexandru Petruț Becheru
Teacher Coordinator: Cătălin Stoean

Abstract

In the past few years applying computational intelligence to games has gained a lot of interest. Car racing simulation games are a great test bed, as researchers can directly see the practicability of their work. This paper is an initial approach to optimizing the basic controller for the game TORCS. We did this by using an artificial intelligence algorithm: Hill Climbing. It was used in finding proper values for gear changing that will make the car faster. A gear changing value represents the rpm value of the motor where the shift is done, i.e. like shifting from 3rd to 4th gear at 2500 rpm. After several laps of a circuit the algorithm was able to determine several sets of values, which in some cases made the car faster by 30%. Just to make things more real we used 2 tracks that resemble a motorway and a city road, and we also introduced several competitors within the same race.

1 Introduction

Computers have made our life better in such many ways that they became part of our daily routine. In the past when someone had to design something he or she was obliged to make a lot of prototypes and test them, which is very time consuming and expensive. Nowadays programs like CAD are able to simulate tests, and aid the users to design the product faster. The accuracy of some programs makes them a necessity to have for every researcher.

Computational intelligence games simulate the real world, or at least some aspect of it, as good as they can. This is done with the scope of easily testing new ideas, we can perceive it as the first feedback necessary to determine the potential of one's idea. Another part that makes these games receive such attention is that one can have fun while working. This translates in better motivated and productive researchers.

Today, cars are a big part of our life, whether we like or not. Every boy and more and more girls dream to have their own car. Some of them dream to race one another, others don't just want to drive, they want to enhance their cars performance. Games made those dreams easier, cheaper and healthier. But what about those people who really want to make a change in the car design, the real car, not just those that powerful graphic cards generate on our monitors? They also have their games, not as fancy looking as the others, but which simulate races to their best. Race simulation is not just about how a car turns brakes and accelerates, it is about why, when, and in which circumstances it does that and many other. Driving on two different roads is not the same, one might need to brake harder and accelerate more, change gear faster and use different techniques of driving.

TORCS [11] (The Open Race Car Simulator) is a multi-platform racing simulator used in artificial intelligence racing as a research platform. If offers a universe of 50 cars, 20 circuits and the possibility to create your own to suit your needs. It has several modules implemented such like damage control, aerodynamics, wheel control; where one can even specify things like the rigidity of the springs. In 2007 at its 10th birthday *Linux Journal* proclaimed it as the most capable open source simulator [12].

The following section will present the previous work done in this research field and the motivation. The software and the approach to the problem are also put into light. Finally the experimental results together with the concluding regards will close this paper.

2 Previous work and motivation

On top of the TORCS platform there are competitions held, one of which is the IEEE *Simulated Car Racing Championship* [13]. In 2011 the competition took place during 3 conferences: EVO held at Torino, ACM GECCCO held at Dublin, IEEE CIG held at Seul; each consisting of 3 races. Its purpose is to encourage the development of intelligent agents capable of driving cars on unknown circuits. It improves the TORCS platform by offering a package of software to aid the development of intelligent agents. A car can be driven by a set of sensors and actuators.

One approach to creating an intelligent agent is learning from another agent. At first Jorge Muňoz et al. [3] tried to imitate the behavior of a human agent. But as he concludes, it is very complicated to learn the human behavior, because of several causes: neither the human makes the same action in the same circumstances, nor performs all actions in a proper way. Creating a pattern for learning is difficult in this situation. Next, they tried to imitate the behavior of a non-human agent, the 2008 winner of the Simulated Car Racing Championship. Although they demonstrated that it was possible to imitate that agent with good results, they had a problem in gear changing learning.

Evolving a rule based agent with the aid of genetic algorithms was tried by Diego Perez et al. [5]. They succeeded in creating about 120 rules, of which they reduced to a subset of 10 to 20 rules depending on the circuit. Their algorithm showed that after a few generations the lap time was reduced and the damage to the car was almost inexistent. In this study gear changing did not take part in the learning process.

Other many approaches have been used to create intelligent drivers, like using Ant Colony optimization algorithm by Luis delaOssa et al. [1], or developing fuzzy rules by Diego Perez et al. [6] just to name a few.

In studying all of the above, gear changing either was not a issue of interest or it became a problem. Gear control module is one of the five basic modules in the architecture of the TORCS Racing Engine according to E. Onieva et al. [4], so the aim of the current research was to find out if proper gear changing can reduce the lap time.

The *Simulated Car Racing Championship* offers a basic agent that can drive a car on a circuit; though it is a pretty slow one it keeps the car from going off track. The idea was to use this agent to run several times on the circuit with different sets of gear changing values given by the AI algorithm, trying to get better lap times. After each lap the agent will send to the AI algorithm the feedback consisting in time of the lap, the AI algorithm based on this result will generate a new set of gear changing values based on which the agent will race again.

At first we had to develop a program which had to automate the running and the collecting of feedback from TORCS, also this program had to ensure the data transmission between the AI algorithm and TORCS, finally it had to manage all the data obtained by writing it to a Microsoft Excel document so that it can be viewed and analyzed with ease.

Next step was to employ the AI algorithm: Hill Climbing (HC). Its parameters had to be adjusted in order to better fit with our scenario. Having the software developed we tested the agent on 2 separate tracks with and without other cars. Unfortunately due to the hardware problems, we could just add 2 other cars, which thankfully was enough to get some conclusions.

3 Tools

In this section you'll find the most important piece of software that we used. Apart from those mentioned in the following subsections we used Microsoft Excel, and Dev-C++ [14]. The obtained software incorporates the AI algorithms and the interface with the TORCS server [15].

3.1 TORCS

TORCS [10][11] is one of the most popular car racing simulator for academic purposes, developed by the *University of Würzburg* and *Politecnicio di Milano*, licensed under GPL (General Public License), so it is free to use. It presents the following advantages according to E. Onieva et. al. [4]:

- Besides being a good car racing game (Fig. 1), it offers a fully customisable environment, like the ones typically used by computational intelligence researchers for benchmarking purposes.
- It features a sophisticated engine (aerodynamics, fuel consumption, traction, etc.) as well as 3d graphics engine for the visualisation of the races.
- It was not conceived as a free alternative to commercial racing games, but it was specially devised to make it as easy as possible to develop your own controller.

It is a fully customisable environment offering the possibility to develop you own cars and circuits [2].

Fig. 1 Torcs game screenshot

3.2 Simulated Car Racing Championship

As previously mentioned, the Simulated Car Racing Championship [12] offers a platform on which racing competition are held. This software package is multi-platform, working on both Microsoft Windows and major Linux distributions. Controllers can be developed both in C++ and JAVA.

The environment and the car's status are perceived by a number of sensors [2]. There are sensors which only give feedback, current car ranking, distance to the opponents etc. Other sensors which are called actuators don't only give feedback but by modifying their value we can control the car, i.e. by modifying acceleration, one can drive faster or slower.

The architecture of the race is client-server based, this way the competition is fair and data can be compared, and if one controller is stuck, the others can run without a problem:

- The client is the game of TORCS, utilized as an interface for the user so he can receive the data and see the race if he wishes. On top of this, there is a communication module that sends data back and forward to server.

- On the server side we can find the engine of TORCS, but there is no visualization here, only the physics engine. During a race, the controller on the client part will receive information concerning the environment and the status of the car from the server at every 10ms. If the controller wishes to act in some way, i.e. change gear, he modifies the specific actuator and sends the data to the server.

3.3 AutoIt

AutoIt [16] is a BASIC (Beginner's All-purpose Symbolic Instruction Code) like programming language, conceived to automate the GUI (Graphical User Interface) of Microsoft Windows, but in can be used for general programming. It simulates the presence of a user by controlling the mouse and the keyboard, thus making possible the manipulation of program windows and the automatisation of user programmed tasks.

It was initially developed to automate the configurations necessary for millions of computers before being shipped to clients. Over time other modules were added for better interoperability with other programs which run on the Microsoft Windows platform, i.e. Microsoft Excel. It also has the possibility to create GUI's by using the following components: edit boxes, check boxes, list boxes, buttons, status bars and combo boxes. This feature makes it more user friendly.

4. The approach

The approach is based on the 3 main objectives. First is to build a stable and reliable platform which will facilitate the integration of the AI algorithms with TORCS and Simulated Car Racing Championship software. The second is to adjust the algorithms to our scenario. Last but not least is to get enough data so we could establish if gear changing does count in the lap time

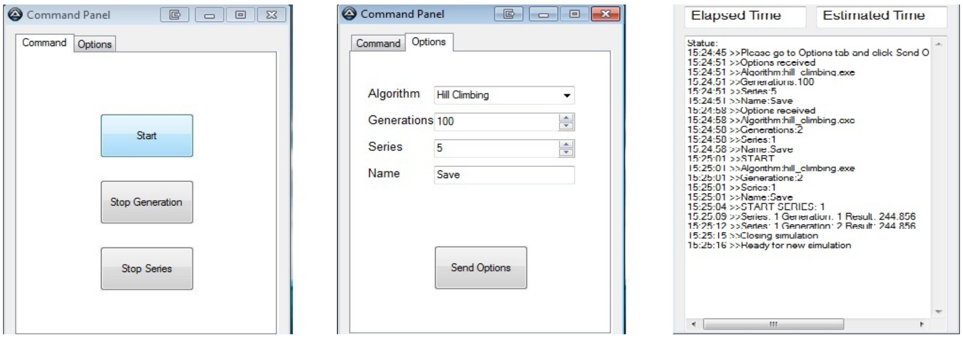
4.1 TORCS and AI algorithm integration

The bridge between the algorithm and TORCS was coded in AutoIt. To facilitate the testing we introduced the concept of *series* and *generations*. A *generation* is one race with a fix number of laps, in our case a generation equals a circuit lap. A *series* is a number of *generations*, after several tests we concluded that a series should have 200 "generations". If this number was smaller then we may not always reach the vicinity of good results, but if it was bigger, it would make no sense hence we already reached it. It offers the following facilities:

- A graphical interface, where we can program the testing and get information about the state of the test shown in Fig. 2. It is composed of a window with 2 functionalities:

  a. Running options and management, in the top half. The *Command* pane contains buttons for the management of the tests: *Start* for starting the tests, *Stop Generation* stops the tests after the current generation end, *Stop Series* stop the tests after the current series/lap. The *Options* pane is where we set the algorithm, the number of generations and series, and the name for the excel file where the data will of the tests will be saved. Also here there is a *Send Options* button, which we must press if we want to save the changes we made.

b. Running state, in the bottom half. Here we receive data about the elapsed time from the beginning of the test, the estimated time until the end of the test, the current number of generations and series, results for each series, push button messages.



Algorithm Hill Climbing
Generations 100
Series 5
Name Save

```
Statue:
15:24:45 >>Please go to Options tab and click Send O
15:24:51 >>Options received
15:24:51 >>Algorithm:hill_climbing.exe
15:24:51 >>Generations:100
15:24:51 >>Series:5
15:24:51 >>Name:Save
15:24:58 >>Options received
15:24:58 >>Algorithm:hill_climbing.exe
15:24:58 >>Generations:2
15:24:58 >>Series:1
15:24:58 >>Name:Save
15:25:01 >>START
15:25:01 >>Algorithm:hill_climbing.exe
15:25:01 >>Generations:2
15:25:01 >>Series:1
15:25:01 >>Name:Save
15:25:04 >>START SERIES: 1
15:25:09 >>Series: 1 Generation: 1 Result: 244.856
15:25:12 >>Series: 1 Generation: 2 Result: 244.856
15:25:15 >>Closing simulation
15:25:16 >>Ready for new simulation
```

Fig. 2 Command Panel, Options Panel, Status Panel (from left to right)

- ➢ The automatisation of the TORCS platform and the Simulated Car Racing Championship software that comes on top, meaning that the bridge is able to start the game and race, close the game without any human intervention
- ➢ Data management which consist in the following:

a. Retrieve the data obtained after running on a track, in our case the elapsed time. This is done by reading an XML file.

b. Transfer data from the controller to the algorithm and back. This was done by reading and writing data to a file, and by starting and pausing the algorithm and the game in an alternative way. When a race is running the algorithm waits for it to finish, next the game is paused and the time is retrieved from the race and passed to the algorithm. Then the algorithm starts, reads the new data and after doing the necessary calculations passes the new set of values to the controller, and pauses.

c. Saving the data of the test into an Excel file. There we can find the time and the set of values for each series. At the end the best time, the average time, and the worst time will be retrieved and written here.

- ➢ The automatisation of the algorithm which means that they can be started, paused and closed.

4.2 Developing the AI algorithm

The algorithm [7][9] needed to be adjusted for the current scenario, this was done during 10000 test laps. We wanted to make it as good as they can be in reaching the best results in a faster way, but at the same time having them consistent. The tests were run on a computer with 2.8 GHz dual core computer, having 4 GB of ram and a graphics card with 256 MB of shared memory. The software simulated a lap of the circuits in about 30 seconds.

4.2.1 The mutation

Within HC one set of gear values is modified to another set of gear values by mutation. We started with uniform distribution (Fig. 3) but it did not conduct to the expected results, the consistency of getting in the vicinity of the best result was low. We then tested new distributions, and discovered one that has a bigger space of search and reaches a consistency where almost every time the algorithms come in a close vicinity of the best result that we found (Fig. 4).

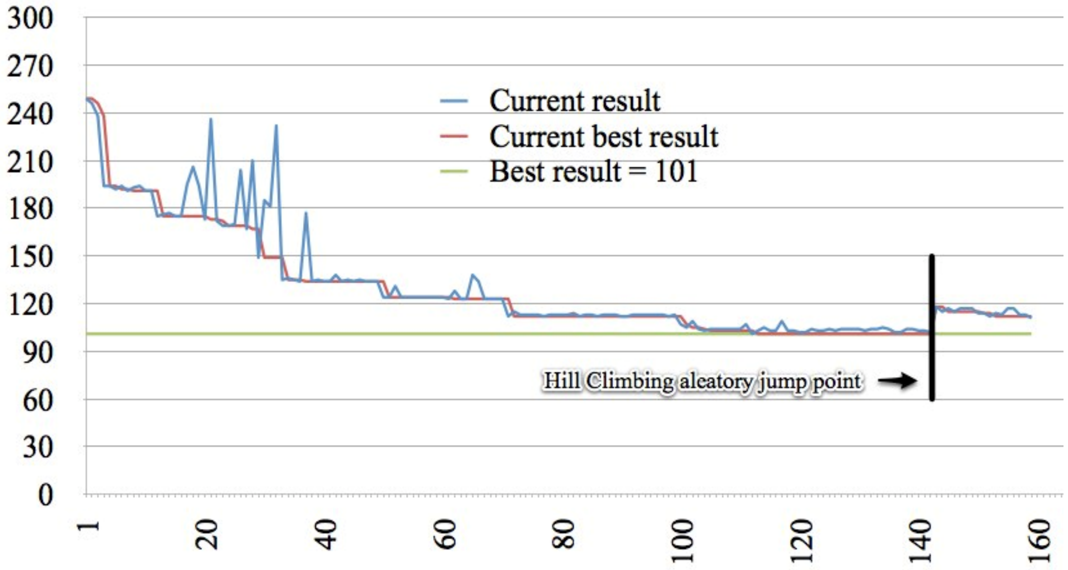


Fig. 3 Results obtained using HC with a uniform distribution

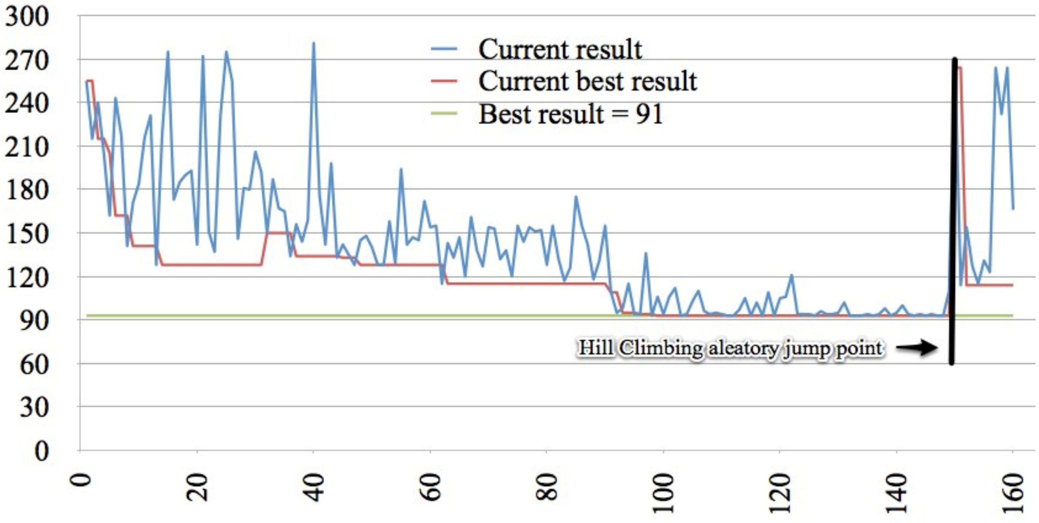


Fig. 4 Results obtained using HC with the proposed distribution

In Fig. 3 and 4, the horizontal axis represents number of generations, and the vertical axis represents the time for each lap. Restart point appears because of the random jumping of the Simulated Annealing algorithm that was used to generate these charts. One can see that the proposed distribution gives results which are lower than 100, while the uniform reaches its best at 100. Also by following the blue line it can be seen that the space of search is significantly bigger. The X parameter for the distribution is calculated the following way:

Algorithm 1 Proposed mutation

```
while $s \geq 1$ do
    $u=rand(1,1000)$
    $v=2*(u/1000)-1$
    $s=2*v^2$
end while
$X= sqrt( (-2* log(s) )/2 )*v$
```

4.2.2 Hill Climbing

For a better understanding of the algorithms the following notions are necessary:

- vectorRpmOptim : is the vector of rpm values with the best result. The #Age represents the number of generations since it has not been modified, and the #Result represents the lap time associated with those values.
- vectorRpm: is the vector that is constructed out of the vectorRpmOptim and is due to be used on the next race. Also #Result attached represents the result associated with it.

This algorithm had to be adjusted to our needs and improved by introducing an random jump (line 6 – 10 of Algorithm 2). If the best result is not improved within 20 generations the algorithm randomly generates a new set of values for vectorRpmOptim. This avoids getting stuck on a local optimal, thus making our search area bigger.

The mutation strength has a value of 500, but if the best result remains unmodified for 10 generations, it is compressed to 250 (line 11 to 15 of Algorithm 2). At first we chose the value of 500 explore the search space better. If that fails we concentrate in slightly improving the current result by changing the value to 250.

The probability for a gear value to change depends on the last result (line 18 of Algorithm 2). A random number is generated between 1 and 1000; if this is smaller that the last result, then the value will be changed. So if the result is higher, then the chance of change increases. The results range from 90 to 300, so in the worst case scenario with the result being 300, there should be an average of 3 gear values changed out of 10. This way we make sure that a consistent change is made, but this is not large enough to make the search chaotic.

Algorithm 2 Hill Climbing

```
1: vectorRpm#Result ← 0
2: vectorRpmOptim ← vector_of_randomly_generated_values[1,8000]
3: vectorRpmOptim#Restult ← 1000
4: while currentExecution ≤ maximum_number_of_executions do
5:         currentExecution ← currentExecution +1
6:         if vectorRpmOptim#age > 20 then
7:                 vectorRpmOptim#Age ← 0
8:                 vectorRpmOptim ← vector_of_randomly_generated_values[1,8000]
```

```
        vectorRpmOptim#Result ← 1000
        vectorRpm#Result ← 0
    else if ( vectorRpmOptim#Age > 10 ) and ( vectorRpmOptim#age ≤ 20 ) then
        mutationStrength ← 500
    else
        mutationStrength ← 250
    end if
    vectorRpmOptim#Age ← vectorRpmOptim#Age+1
    for i ← 1, i ≤ vectorRpm#Length do
        if randomly_generated_number[1,1000] ≤ vectorRpm#Result then
            deviation=mutationStrength*X
            vectorRpm[i] ← vectorRpmOptim[i]+deviation
        end if
    end for
    Algorithm is paused and a new race is run with the values from vectorRpm
    vectorRpm#Result ← Lap Time
    if vectorRpm#Result ≤ vectorRpmOptim#Result then
        vectorRpmOptim ← vectorRpm
        vectorRpmOptim#Result ← vectorRpm#Result
    end if
end while
```

4.3 Circuit choosing

We chose two tracks for our tests. The first one (Fig. 5 left side) resembles a major city artery with a width of 15m and medium to fast corners, the second one (Fig. 5 right side) looks a lot like a highway with a width of 30m and fast corners. The tracks length is not important as we will not compare the two of them directly. In choosing the tracks we also consulted [8].

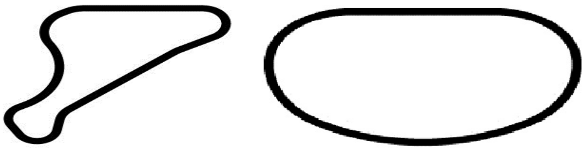


Fig. 5 On the left you can see the city track, and on the right the highway track

5. Experimental results

In the following charts there are some results that spike up to the value 300, this could be explained either by the random jump in Hill Climbing, or by impossibility to run the race. The horizontal axis represents the number of laps / generations, and the vertical axis the lap time. In the charts where will present specific values of the gears, the horizontal axis represents the gears and the vertical axis represents their value. In the latter case there will be 10 gear values, 5 are for changing gear upwards, i.e. from 3rd to 4th, the other 5 are for changing gear downwards, i.e. from 2nd to 1st.

We needed to track the improvement to the lap time. At first we calculated the average of the results obtained from the 200 laps. Then by comparing the best result obtained during those 200 laps to the average we calculated the improvement.

On the city track Fig. 6 (read it form right to left), the results get better as the generations pass. Actually the improvement is 34% compared to the average of results. The best lap was 91.338 and the average 122.60538.

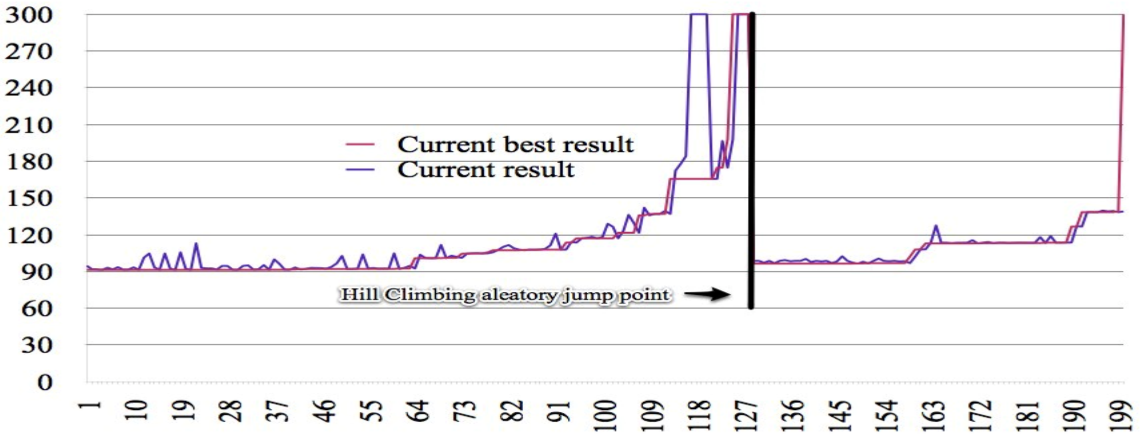


Fig. 6 HC results on the city track with no other cars

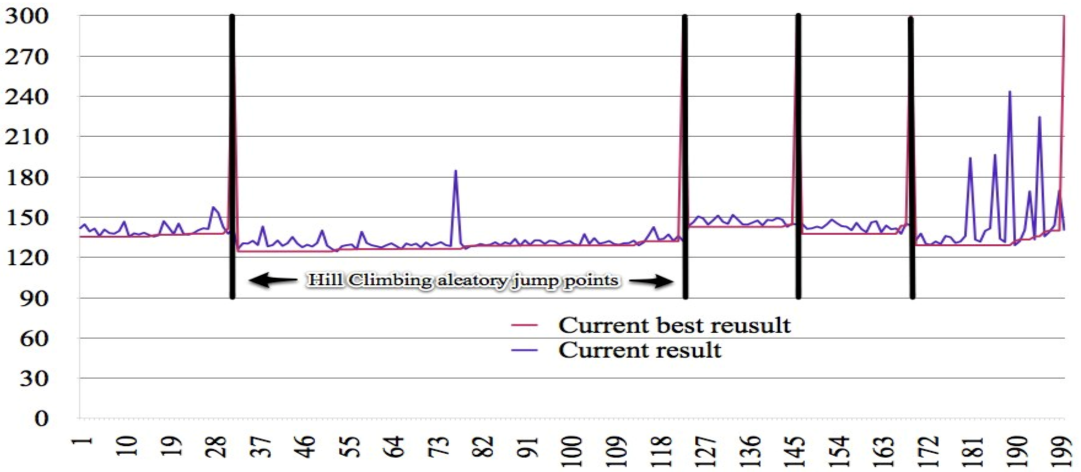


Fig. 7 HC results on the highway track with no other cars

On highway track in Fig. 7 (read from right to left) the improvement is not so obvious, 11% percent is not as good as the one before but we believe it's significant. One explanation could be that on the highway track, the controller doesn't manipulate the car so often as in the city track.

Next we introduced two other cars on both circuits .We realised that on the city circuit these had influenced the results in a bad manner, but the algorithm managed to improve the results. With our car alone on the circuit the best results came at about 90, with two other cars they were about 110. But considering that the average results even with one car was 120, this is still significant. On the highway circuit the results showed the oponents had no effect on the algorithm performance, the results were all in the range of 120 to 125. One might say that introducing other cars made the controller a little faster, but the difference is to small to say that for sure.

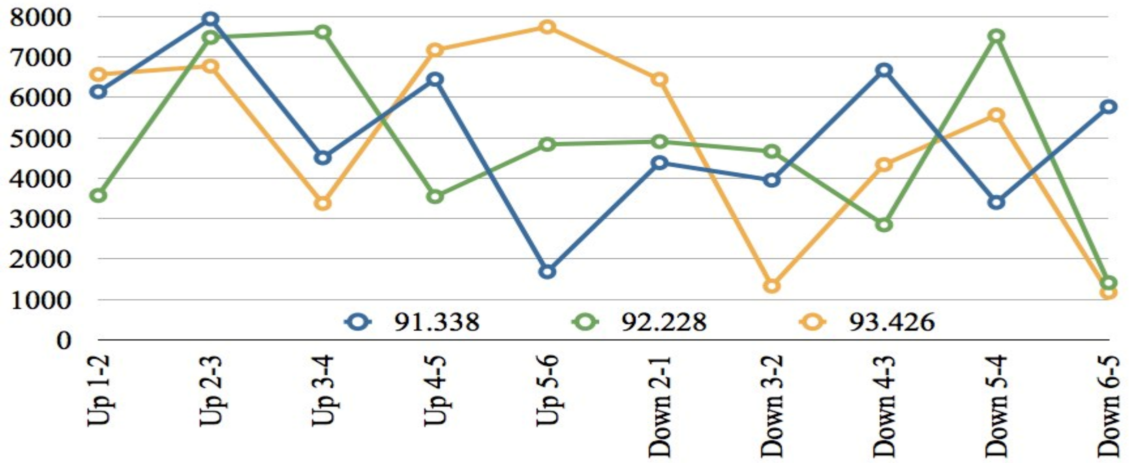


Fig. 8 Gear values for city track, no other cars

Up 1-2	Up 2-3	Up 3-4	Up 4-5	Up 5-6	Down 2-1	Down 3-2	Down 4-3	Down 5-4	Down 6-5	Result
6139	7942	4504	6449	1683	4382	3951	6677	3399	5768	91.338
3571	7484	7619	3543	4837	4905	4663	2843	7520	1412	92.228
6566	6773	3374	7171	7745	6446	1328	4335	5566	1173	93.426

Table 1 Gear values for city track, no other cars

Up 1-2	Up 2-3	Up 3-4	Up 4-5	Up 5-6	Down 2-1	Down 3-2	Down 4-3	Down 5-4	Down 6-5	Result
4214	2887	5946	7001	4233	1932	7144	4532	2564	2904	124.41
6252	1962	6561	4373	6591	2214	4559	2409	2421	6276	124.072
4214	2887	5946	7001	4233	1932	7144	4532	2564	2904	124.41

Table 2 Gear values for the highway track with 2 other cars

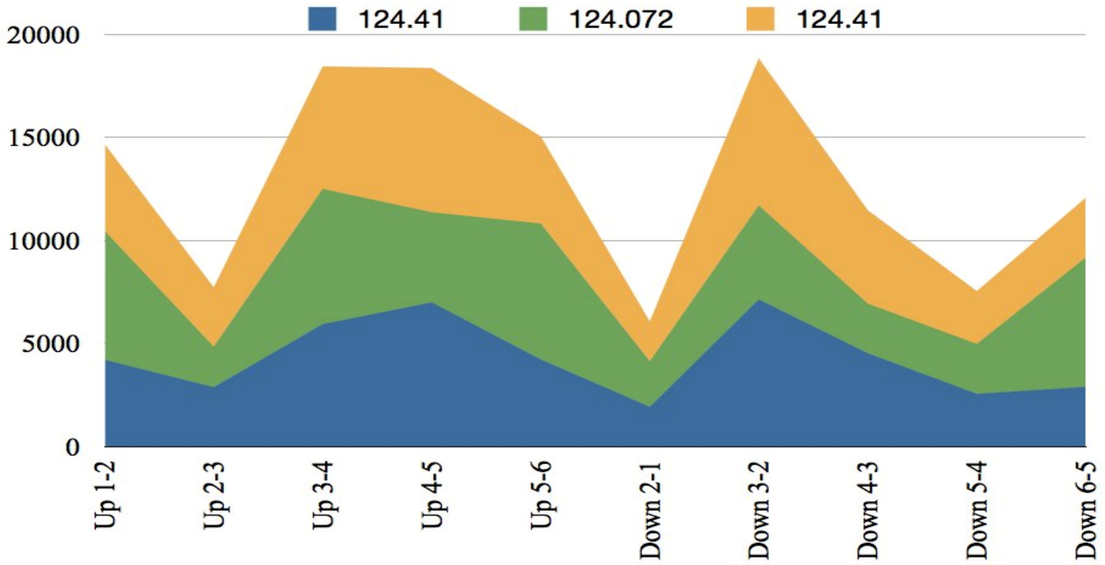


Fig. 9 Gear values for the highway track with 2 other cars

After this, we looked upon the set values which gave the best results. On the city track with no other cars, the values appeared to be absolutely random, as shown in Fig. 8 and Table 1. On the other hand on the highway track, the values of the gears look to be correlated, as shown in Fig. 9 and Table 2.

6. Concluding regards

We started work on this paper with the scope to find out if gear changing learning could improve the lap time. As the results show there is no doubt that lap time is influenced by gear changing values. The data obtained depicts big improvements 10% to 30 %, but considering that the used controller is a slow one, we can't tell the margin of improvement that may be achieved in a good controller. One thing is certain, from now one every controller designer should take in consideration gear changing learning.

Regarding the algorithm, we recommend testing and improving it. But we believe that other algorithms could give good results, so in a further research we shall try adapting other algorithms to this scenario. This poses another question for future development: Is there a way that we can create an algorithm that can return the best sets of values for gear changing, by serving it with the tracks proprieties?

Another thing that we learned during these tests is that there is no universal set of values that will give very good results no matter the track. But we can imagine that in the future there will be intelligent gear boxes that will adapt to the track proprieties and probably even to the driving mode.

References

[1] Luis delaOssa, José A. Gámez and Verónica López. *Improvement of a car racing controller by means of Ant Colony Optimization algorithm. IEEE Symposium on Computational Intelligence and Games, 2008.*

[2] Daniele Loiacono, Luigi Cardamone and Pier Luca Lanzi. *Simulated Car Racing Championship 2010 Competition Software Manual. Politecnico di Milano, Dipartimento di Elettronica e Informazione, Italia, may 2010. IEEE Symposium on Computational Intelligence and Games, 2011.*

[3] Jorge Muňoz, German Gutierrez and Araceli Sanchis. *Controller for TORCS created by imitation. IEEE Symposium on Computational Intelligence and Games, 2009.*

[4] E. Onieva, D. A. Pelta, J. Alonso, V. Milanés and J. Pérez. *A Modular Parametric Architecture for the TORCS Racing Engine. IEEE Symposium on Computational Intelligence and Games, 2009.*

[5] Diego Perez, Yago Saez, Gustavo Recio and Pedro Isasi. *Evolving a rule system controller for automatic driving in a car racing competition. IEEE Symposium on Computational Intelligence and Games, 2008.*

[6] Diego Perez, Gustavo Recio, Yago Saez and Pedro Isasi. *Evolving a Fuzzy Controller for a Car Racing Competition. IEEE Symposium on Computational Intelligence and Games, 2009.*

[7] David L. Poole and Alan K. Mackworth. *Articial Intelligence Foundations of Computational Agents. Cambridge University Press, 2010. http://artint.info/html/ArtInt.html*

[8] Mike Preuss, Jan Quadflieg, Günter Rudolph. *TORCS Sensor Noise Removal and Multi-objective Track Selection for Driving Style Adaptation.*

[9] Cătălin Stoean and Ruxandra Stoean. *Evolutie si inteligenta articiala. Paradigme moderne si aplicatii. Editura Albastra, 2010.*

[10] Bernhard Wymann, T.O.R.C.S. *Manual installation and Robot tutorial. http://berniw.org/*

[11] The Open Race Car Simulator. *http://torcs.sourceforge.net/*

[12] Linux Journal regards toward TORCS. *http://www.linuxjournal.com/node/1000435*

[13] Simulated Car Racing Championship. *http://cig.ws.dei.polimi.it/*

[14] Dev-C++. *http://www.bloodshed.net/devcpp.html*

[15] Link for downloading the software developed by us. *http://inf.ucv.ro/~cstoean/diploma.html*

[16] AutoIt. *http://www.autoitscript.com/site/autoit/*

Alexandru Petruț Becheru
University of Craiova
Faculty of Exact Sciences
Department of Computer Science
Str. A.I. Cuza, Nr. 13, Craiova
ROMANIA
E-mail: alex.becheru@inf.ucv.ro

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Game Based Learning
Project Sotirios

Author: Florentin Bota
Teacher Coordinators: Prof. PhD. Dana Simian, Lect. Ralf Fabian

Abstract

The purpose of this project is to design and develop a unique state-of-the-art video game where the regular player is given the opportunity to learn by play. Unreal Development Kit is used as the game engine, but other editing programs were also required.

The finished video game will be called Sotirios (Saviour in Greek) and will follow the story of a future robot who dared to question the axioms that ruled his world. During play, the user will be faced with different puzzles and lessons about programming and IT, which will be hidden inside the game flow and logic.

This project intends to create an effective e-learning (Game Based Learning) application, combining cutting-edge graphics with learning puzzles.

1 Introduction and description

Game Based Learning is one of the fields that are not being developed as a serious alternative to standard teaching systems, but the current technology offers a solid base for most programmers to develop in this direction. In addition, in this paper we will be referring to **Digital** Game Based Learning (DGBL) as opposed to board, card or class games.

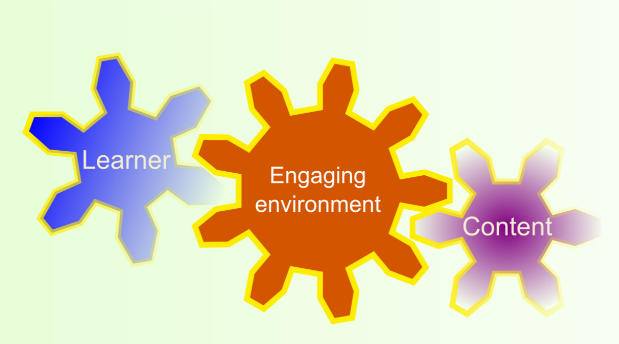


Fig. 1: Linear learning in video games

GBL can be considered a branch of serious games that follow defined learning outcomes. Actually, if you take out the fun out of a good game, what is left and makes it enjoyable is nothing else but the built-in learning process. When we progress in a game, we are actively learning, our minds are experiencing the pleasure of understanding and using a new system, the same way as a child explores the world with curiosity. This is especially true when the game is considered "entertainment" (e.g., Defence of the Ancients) or "serious" (e.g., an FAA-approved flight simulator).

The real question *is how a regular student can learn to use over 53.000 items from a game like World of Warcraft but has problems in writing a simple sorting algorithm in Java?* We started this project to help regular students or pupils solve difficult tasks with the use of simple steps hidden in an enjoyable game. Of course, Edutainment is applicable to any field and in various degrees according to the age of the subjects. This way, a life-simulator, a run-and-jump game is suitable for children, whereas a FPS can be played by adult students without the danger of them loosing interest. **Edutainment** (also infotainment or entertainment-education) is a form of entertainment designed to educate as well as to amuse.

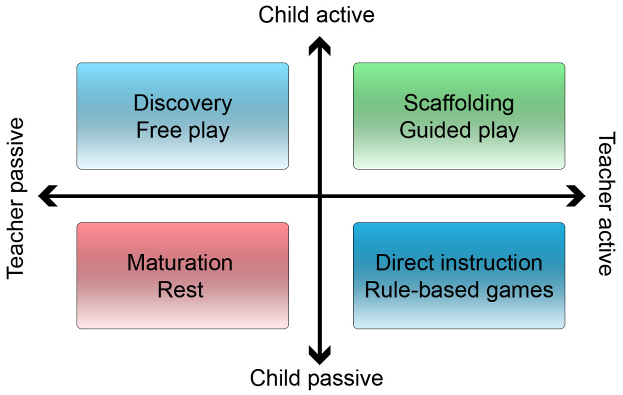


Fig. 2: Graph for direct instruction and play
Source: http://www.naeyc.org/content/research-news-you-can- use-play-vs-learning

To answer the question above, we have to understand the way the human brain works. When we learn something, we do not only memorize the content, we create complex links and connections between different entities and concepts, so when the training process becomes dull and uninteresting, the performance drops suddenly and is not efficient. This is a problem we see everyday especially at younger colleagues from high school, where the importance of social networks and social games is considered greater than the actual learning by an increasing number of people. The play vs. learning dichotomy from Fig.2 can be applied at any age and we consider the video game as a teacher.

Digital GBL can be considered a form of e-learning, as E-learning comprises all forms of electronically supported learning and teaching. The information and communication systems, whether networked learning or not, serve as specific media to implement the learning process, creating out-of-classroom and in-classroom educational experiences via technology, even as advances continue in regard to devices and curriculum.

The *motivation of this project* came after I saw a report on a *15 year old student* who declared that whenever he starts learning for school he gets bored after half an hour and then he plays *HALO* for several hours with his friends.

The principle is simple, all we have to do is hide the information we want the user to learn inside the game, but do it in a way that does not deconstruct the fun or the normal flow of the game, as can be seen in Fig.1 and Fig.3. This is not an easy task, simply because designing a successful game is hard, but designing a great educational game is even harder. Mostly because when the player notices that he is using an educational application, he looses interest. This is why the curriculum should be blended inside the theme of the game and the reason we chose a robot- sci-fi story for our informatics lessons.

Fig. 3: Digital Game-Based Learning Cartoon
Author: Son Le
Source:http://www.flickr.com/photos/donjonson/5351362611/

1.1 About Digital Game Based Learning

Games usually have a fantasy element with narrative or storyline that engage players. Educational video games can be motivating to children and allow them to develop an awareness to consequentiality. Children are allowed to express themselves as individuals while learning and engaging in social issues, without facing real-life problems. Today's games are more social, with most teens playing games with others at least some of the time and can incorporate many aspects of civic and political life. Students that participate in educational video games can offer deeper, more meaningful insights in all academic areas. The success of game-based learning strategies owes to active participation and interaction being at the centre of the experience, and signals that current educational methods are not engaging students enough.

The cartoon from **Fig.3** is a perfect example of what we are promoting with this project: a future where learning is fun and easy. Experience with and affinity for games as learning tools is an increasingly universal characteristic among those entering higher education and the workforce, but it should be considered even at lower educational levels, simply because of their efficiency. As a side note, most of the people who played Rome Total War or even Age of Empires will remember the Historical Battles (with accurate data) because they probably spent days playing them in an enjoyable way, but won't remember the same battles if they studied them for a History exam after a few weeks. This is also because memorizing data is not enough to learn. We have to create connections and develop skills in order to remember the information for long periods.

Game-based learning is an expansive category, ranging from simple paper-and-pencil games like word searches all the way up to complex, massively multiplayer online (MMO) and role-playing games (RPG). The use of collaborative game-based role-play for learning provides an opportunity for learners to apply acquired knowledge and to experiment and get feedback in the form of consequences or rewards, thus getting the experiences in the 'safe virtual world'.

1.2 Project motivation and objectives

The reason game design was chosen was the fact that it is a passion that started from the first time I had a PC and played Volfied, as a child. I started to question the way it was created and worked, so it was only a matter of time until I learned IT and tackled areas like Image Processing, E-learning and so on. This project is yet another challenge, combining the entertainment of a modern FPS with the puzzles of a C++ e-learning application.

Regarding the practical application, two perspectives were addressed:

1) Creating a prototype for a First Person Shooter (FPS) using UDK in order to take advantage of the modern game engine capabilities, completed with an interesting storyline.
2) Finding a way to embed the learning puzzles in the storyline and creating a different GUI for them, also making the game easy to play even for non-student players.
3) Creating a Turned Based Game (**TBS**) with a History theme for mobile phones and tablets.

Secondary objectives in the above example:

a. A better understanding of UDK technology and DirectX/OpenGL functions.
b. Learn how to design a simple map in UDK.
c. Learn how to use 3D modelling and rendering software and import the results in the game scene.
d. Create a unique story and gameplay.
e. Create animations and trailers to make the game more vivid.
f. Test the prototype on different OS's to check for compatibility issues.
g. Create an outside puzzle database that can be easily manipulated.
h. Promote the finished product.

2 Materials and Methods

2.1 Unreal Development Kit

Fig. 4: UDK Logo

UDK is Unreal Engine 3 – a complete professional development framework game engine developed by Epic Games, first version illustrated in the 1998 first-person shooter game Unreal. Although primarily developed for first-person shooters, it has been successfully used in a variety of other genres, including stealth, MMORPGs and RPGs. We can use it in this project because of it's formidable build-in functions for complex game-developing actions. With its core written in C++, the Unreal Engine features a high degree of portability and is a tool used by many game developers, researchers, television studios, machinima directors, artists and students.

Moreover, it is completely free for non-commercial and educational projects and a great deal of the gameplay code can be written in UnrealScript, a proprietary scripting language, and as such, large parts of the gameplay can be modified without delving deep into the engine internals. Additionally, as with other middleware packages, the Unreal Engine also provides various tools to assist with content creation, both for designers and artists.

The *originality* of this project comes from the idea of combining e-learning with cutting edge graphics as the ones offered by UDK.

2.2 Autodesk 3ds Max

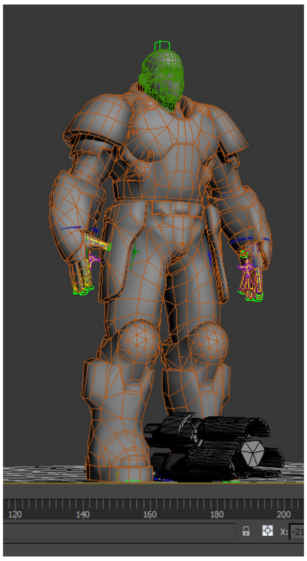


Fig. 5: Setting a biped rig for UDK in 3ds Max

Autodesk 3ds Max, formerly 3D Studio Max, is 3D computer graphics software for making 3D animations, models, and images. It was developed and produced by Autodesk Media and Entertainment. It has modeling capabilities, a flexible plugin architecture and can be used on the Microsoft Windows platform. It is frequently used by video game developers, TV commercial studios and architectural visualization studios. It is also used for movie effects and movie pre-visualization.

In addition to its modelling and animation tools, the latest version of 3ds Max also features shaders (such as ambient occlusion and subsurface scattering), dynamic simulation, particle systems, radiosity, normal map creation and rendering, global illumination, a customizable user interface, and its own scripting language, all delivering a polished tool for 3d modelling. A completely free but powerful alternative is Blender.

Autodesk studio can also be used free by students for non-commercial projects and is an incredible asset for creating custom objects in the virtual world, as seen in the left image.

2.3 Adobe Flash

Although Flash is mainly used for web development and simple animations, it's compatibility with UDK makes it great for designing User Interfaces (UI) that can be accessed from the engine using fscommand() function. We used it to create the puzzles, validate the answer and push the action forward.

3 Sotirios

Sotirios is a robot who dared to question the axioms that ruled his world. Project Sotirios follows the main character as he escapes from the brainwashing camp where he was being kept and explores the world outside the walls. In order to escape, he has to force his way out, using more and more difficult algorithms to solve simple and complex problems. A door won't open for example if the cracking code is not semantically correct, and a wrong choice can alert the nearby guards. The player has to stay with his eyes open and check for both fight and programming solutions to the puzzles from the game.

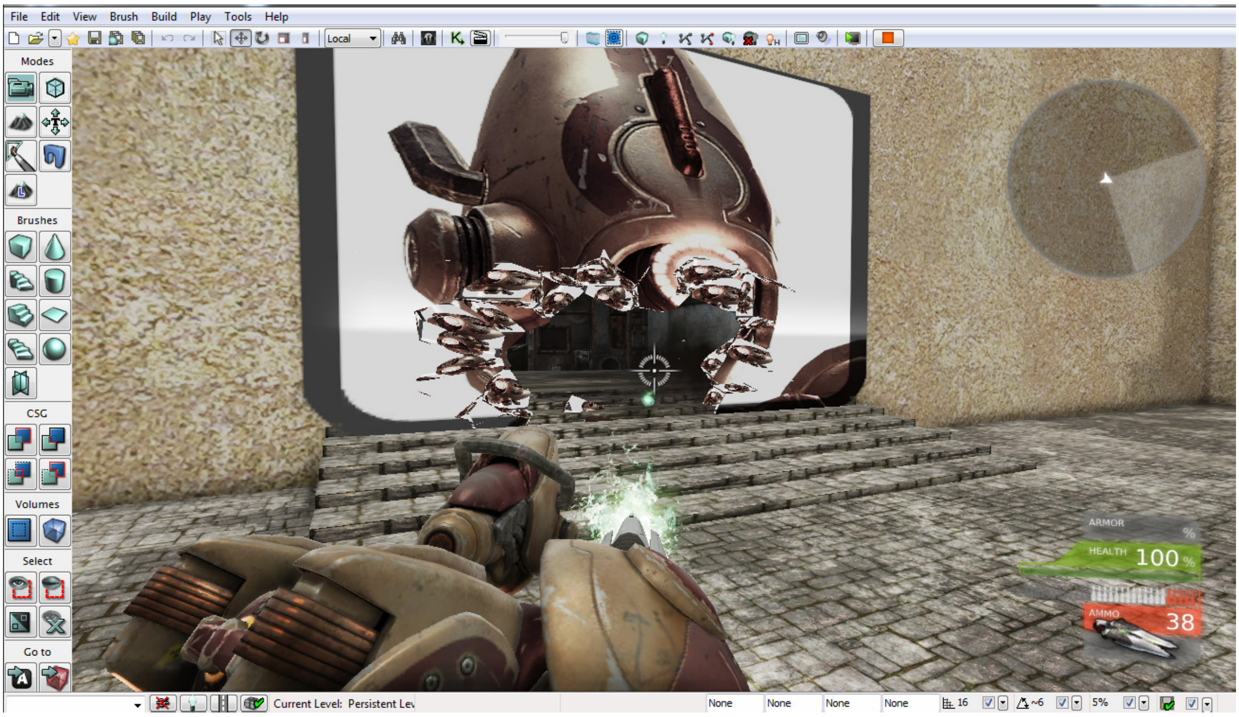


Fig. 6: Sotirios breaking the monitor to escape the first room , an editor view

In order to destroy and pass through the monitor as in Fig.6, the objects have to be fractured and pre-broken. In Fig.8 we can see the fracture tool in action, pre-slicing object so they can be destroyed in-game. Kismet (Fig.7) is an invaluable tool, a modern programming environment, where objects are represented visually.

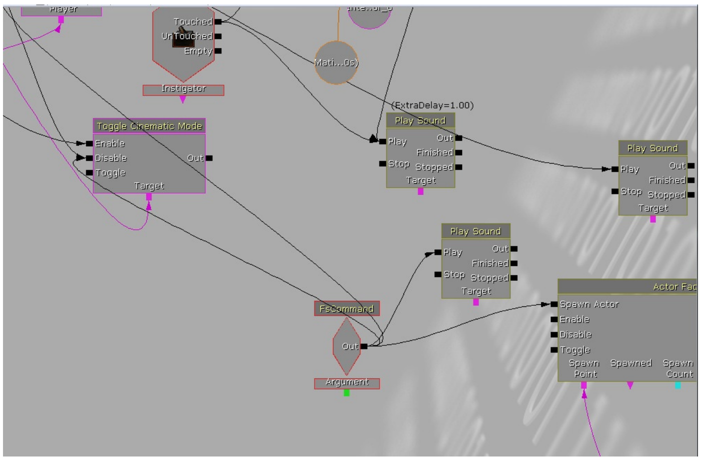


Fig. 7: Kismet, the visual editor

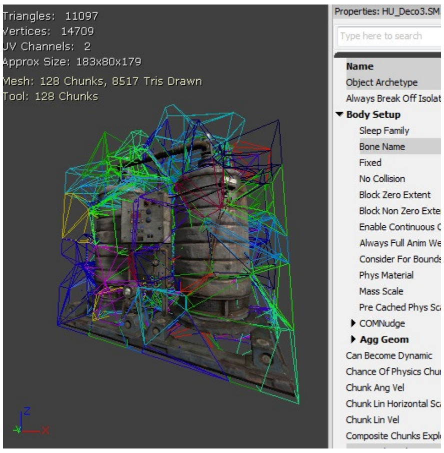

```
Triangles:  11097
Vertices:  14709
UV Channels:  2
Approx Size: 183x80x179
Mesh: 128 Chunks, 8517 Tris Drawn
Tool: 128 Chunks
```

Fig. 8: The Fracture Tool in UDK

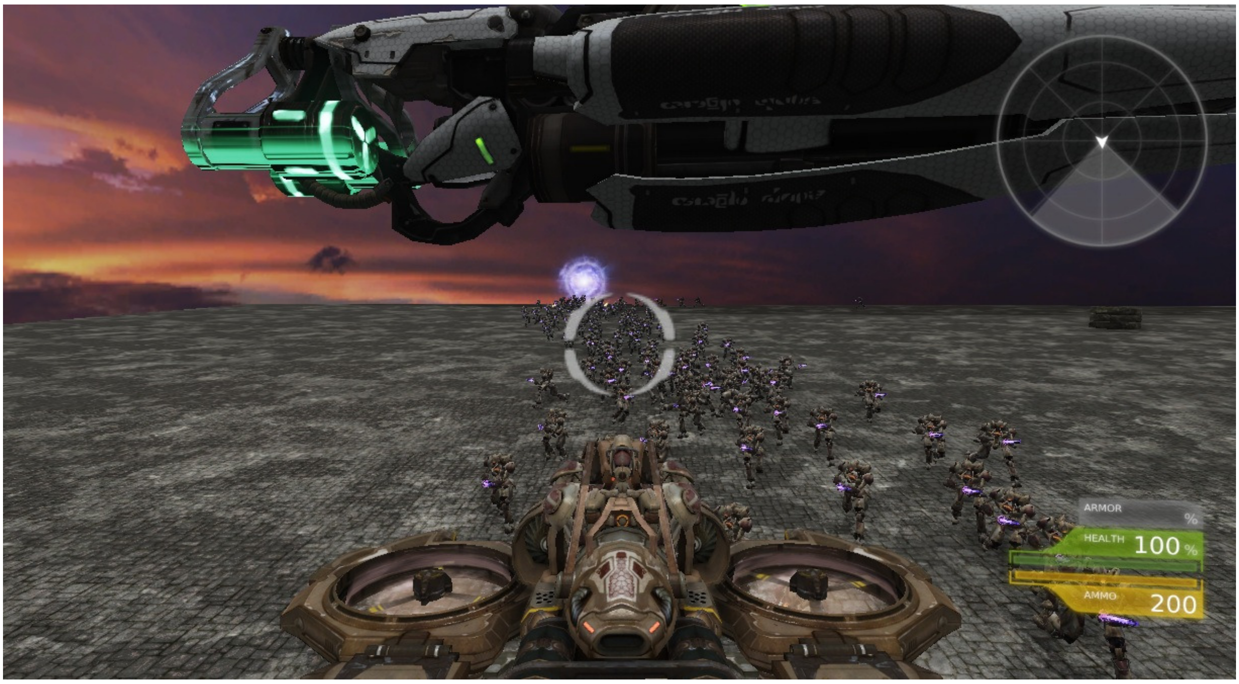
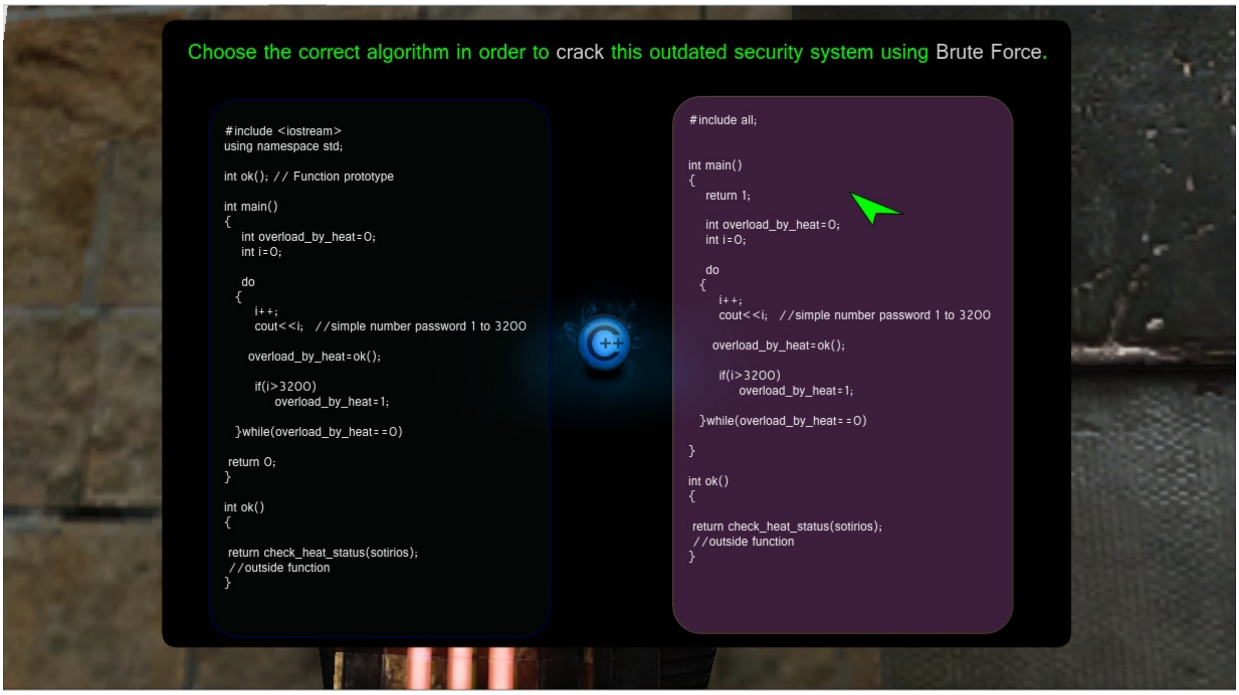


Fig. 9: Using a crowd control agent and a vehicle in UDK

But there would be nothing interesting about this game without the learning puzzles. Here we will present a short demo to prove that it can be done. As written earlier, Adobe Flash was used to create the quiz UI. The player selects the answer he thinks is correct.



Choose the correct algorithm in order to crack this outdated security system using Brute Force.

```
#include <iostream>
using namespace std;

int ok(); // Function prototype

int main()
{
   int overload_by_heat=0;
   int i=0;

   do
  {
      i++;
      cout<<i;   //simple number password 1 to 3200

     overload_by_heat=ok();

      if(i>3200)
          overload_by_heat=1;

  }while(overload_by_heat==0)

 return 0;
}

int ok()
{

 return check_heat_status(sotirios);
 //outside function
}
```

```
#include all;


int main()
{
   return 1;

   int overload_by_heat=0;
   int i=0;

   do
  {
      i++;
      cout<<i;   //simple number password 1 to 3200

     overload_by_heat=ok();

      if(i>3200)
          overload_by_heat=1;

  }while(overload_by_heat==0)

}

int ok()
{

 return check_heat_status(sotirios);
 //outside function
}
```

Fig. 10: Using a trigger to open the puzzle UI, with immediat effect on the game flow, based on player selection

If right, the door will open and he is free to venture outside. If he chooses the semantically incorrect algorithm, a bot is spawned and the player is forced to fight with it. The different outcomes can be seen in Fig.11.

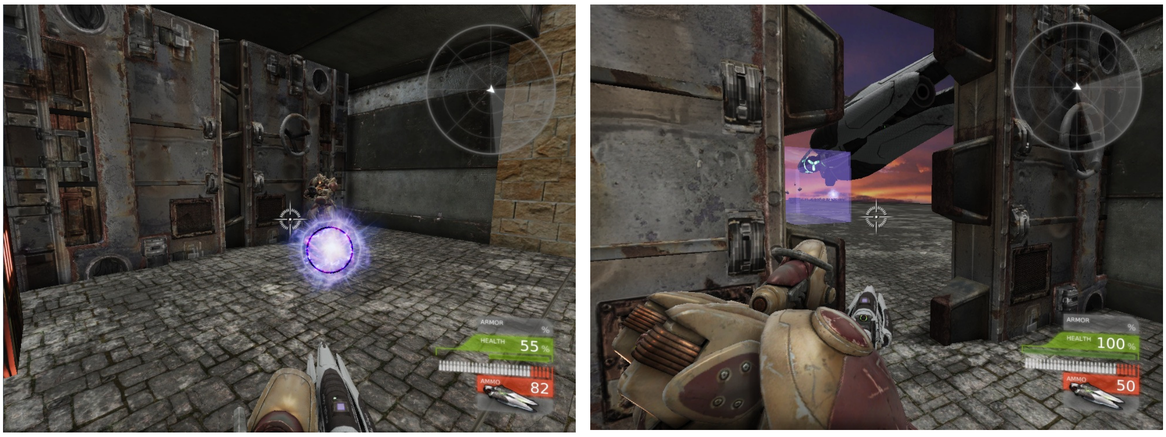


Fig. 11: Different outcomes, based on the player's answer

4 Conclusions and future development

In conclusion, most of the objectives have been met and the pre-alpha prototype received extremely positive feedback, 100% of the subjects asked for a copy of the final game, when it will be released. The *'Sotirios'* project presented in this paper and conducted as part of practice during my *Master-Thesis* is the most significant and original part of the work characterized by design, research and programming.

Future developments follow:

- Multi-level design
- More custom-made meshes
- A multiplayer plugin
- Experience system
- A help system

References

[1] Kelly L. Murdock,*3ds Max 2012 Bible*, John Wiley & Sons, Canada, 2011.
[2] Alan Thorn, *Udk Game Development,* Course Technology, 2012, ISBN 1435460189
[3] Wikipedia, Online Encyclopedia, E-learning, accessed January 2012, http://en.wikipedia.org/wiki/E-learning .
[4] Wikipedia, Online Encyclopedia, Article about GBL, accessed January 2012, http://en.wikipedia.org/wiki/Game_based_learning .
[5] Website of Rome Total War, a Historical TBS, http://www.totalwar.com/rome.
[6] Naeyc.com, Association for young children education, Learn vs play dichotomy, accessed February 2012, http://www.naeyc.org/content/research-news-you-can-use-play-vs-learning.
[7] Articles about UDK on Wikipedia: http://en.wikipedia.org/wiki/Unreal_Engine, accessed October 2011.
[8] Video tutorials by kcin228, vizited on November 2011, http://www.youtube.com/watch?v=Y80Eg_Jj2J4.

Bota Florentin
"Lucian Blaga" University
Faculty of Sciences, Informatics
Sibiu, 5-7, Ioan Ratiu str.
Romania
botaflorentin@yahoo.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Self growing ASP.NET site with custom CMS

Francesco Capuzzo

Teacher coordinator: Prof. univ. Dr. Dana Simian

Abstract

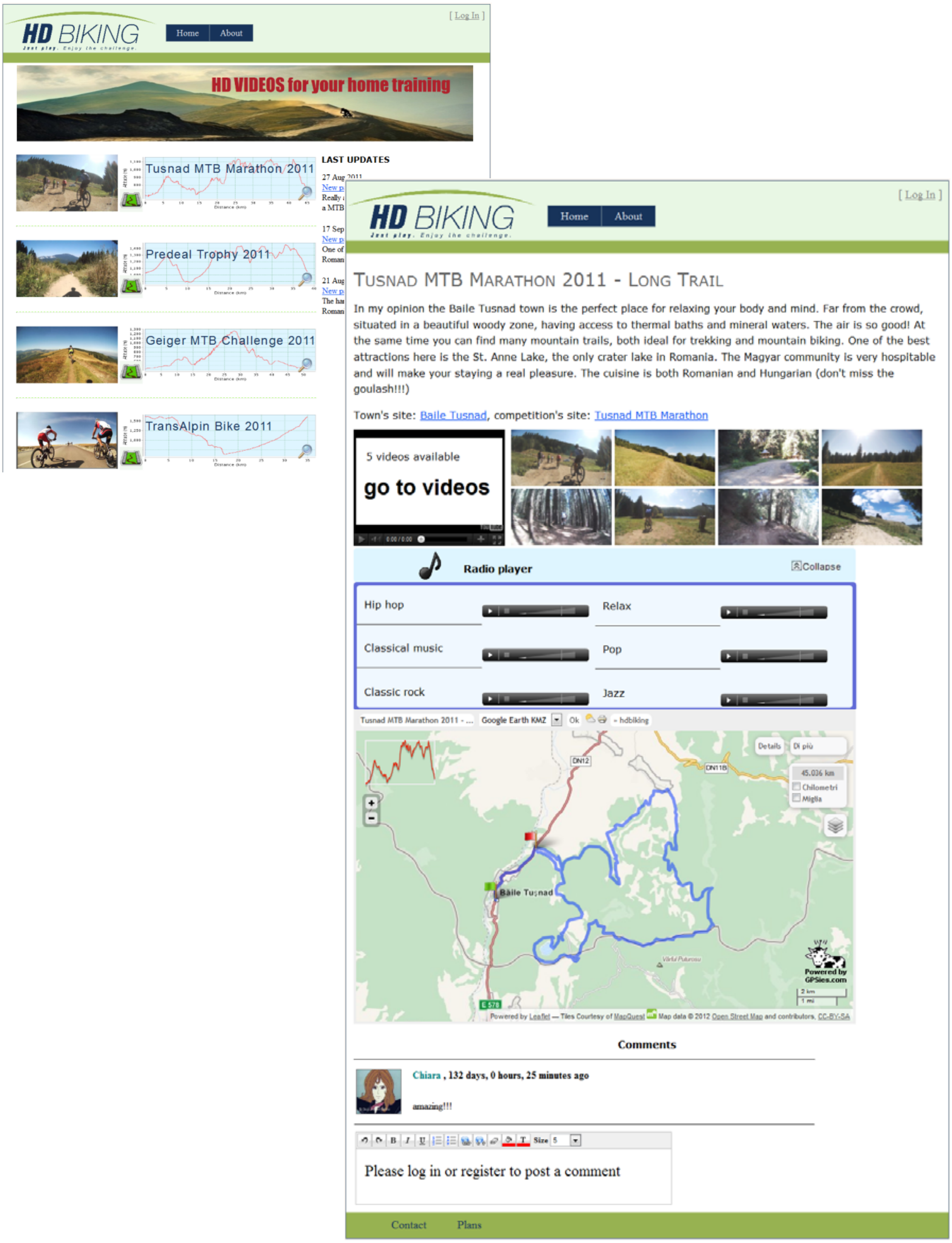
This paper presents an easy way to let an Internet site grow in size just with the contribution of the active users, like YouTube and Facebook do. It's simple and this technique can be applied to many different domains. The key point is to use a custom CMS functionality based on generating the html code programmatically.

1 Introduction

The most successful sites are those that constantly have fresh contents or offer useful services. In this article we are going to explore the first group. Apart from blogs, which are normally populated by a single person, the other sites that offer information have to be populated by a team of people or by a community. Building a custom CMS (Content Management System) it is possible to see the site growing with new pages every day. These pages are built after a specific template in order to keep the site's layout clean and consistent.

2 Application context and development environment

In the example shown in this paper we are using a SQL Server 2008 database and a Visual Studio 2010 development environment. The real-life example is implemented in the Internet site http://hdbiking.net, which is a collection of videos shot from bicycles, and much more. The site's aim is to offer video and gps documentation, together with a description, pictures and comments, on mountain tracks that can be rode on a mountain bike, or asphalted roads. At the same time it is useful as footage for training while doing indoor cycling.



TUSNAD MTB MARATHON 2011 - LONG TRAIL

In my opinion the Baile Tusnad town is the perfect place for relaxing your body and mind. Far from the crowd, situated in a beautiful woody zone, having access to thermal baths and mineral waters. The air is so good! At the same time you can find many mountain trails, both ideal for trekking and mountain biking. One of the best attractions here is the St. Anne Lake, the only crater lake in Romania. The Magyar community is very hospitable and will make your staying a real pleasure. The cuisine is both Romanian and Hungarian (don't miss the goulash!!!)

Town's site: Baile Tusnad, competition's site: Tusnad MTB Marathon

Comments

Chiara , 132 days, 0 hours, 25 minutes ago

amazing!!!

Please log in or register to post a comment

Fig. 1: Default page top part and a detail page of hdbiking.net Internet site.

3 Main components of the architecture

The default page is a collection of links to other pages, as the default page of YouTube is. After clicking a link to a track, the user is redirected to the detail page. Every user can, through a special section of the site, add content to the site. This content is represented by a new entry in the default page and an entire new page that can be accessed through it. Actually, the detail pages are not different pages; instead they are the same page whom is given a specific parameter in the url, so that it accesses a specific content each time. As an example, here follow the links to two different tracks specific pages:

http://hdbiking.net/TrackPage.aspx?Name=TransAlpin_Bike_2011

http://hdbiking.net/TrackPage.aspx?Name=Geiger_MTB_Challenge_2011

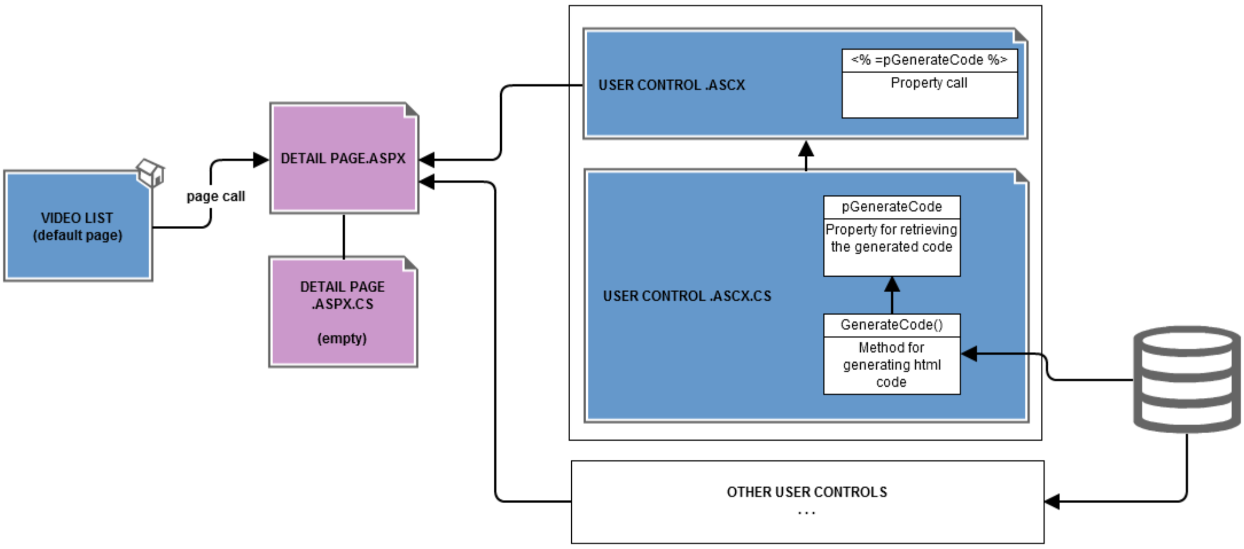


Fig. 2: Populating the detail page of each track selected from the list in the default page.

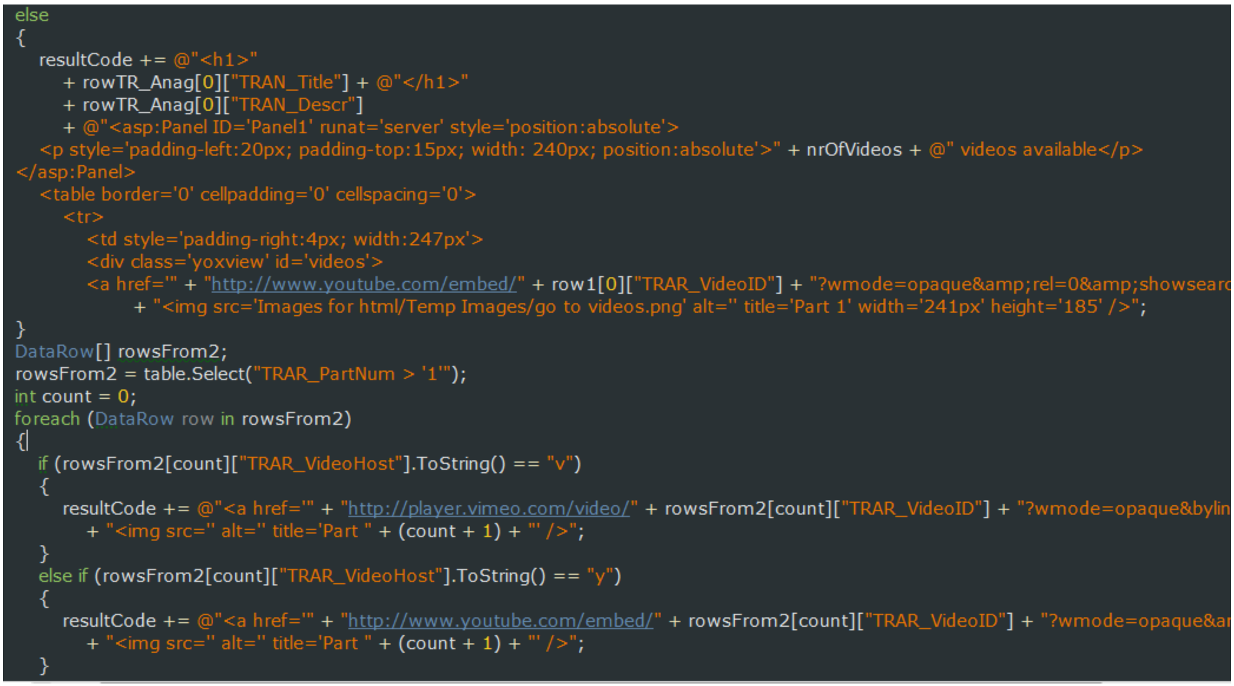
In the above picture, you can see that the html content shown in the detail page is built within one or multiple user controls (ascx pages), whose content comes from the database. It is a simple concatenation of html code and the content retrieved from the database. The GenerateCode method of the user control code behind page returns a string. That string is the complete html code needed by the detail page in order to show a specific content. The pGenerateCode property, in the user control code behind page, is needed by the user control page to show the content of the GenerateCode method, using the call <% =pGeneratecode %>. The user controls are suddenly referenced as usual in the detail page [1].

3.1 Database

The database can contain raw data or data comprehensive of html tags. It is up to the programmer to chose the best combination depending on the situation. Having html tags in the database seems to be more practical, as the GenerateCode method results clearer, but in the case the programmer wants to change the layout of the application, the change has to be made both on the code behind as well as in the records in the database. The second choice seems to break the consistency principle and is more exposed to error rising. For generating the detail pages, the GenerateCode method asks the database for the data filtered by the track selected in the default page. The data obtained is treated as a group of parameters needed by the GenerateCode method itself. As you can see from fig. 1, the detail page of a track contains links to pictures and videos (in this case 5 videos). Everything is managed through modal windows and jQuery effects. The way the users populate the database remains to the programmer choice and is not discussed in this paper.

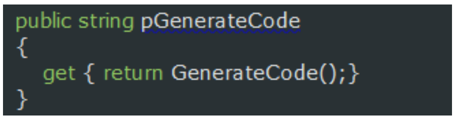
3.2 User controls

Using user controls is not mandatory. But in some cases different html components cannot be nested in the same generated code; in order to show properly, they have to be imported separately in the aspx page using user controls. This choice, at the same time, makes the code cleaner and more manageable. In the user controls, the code is generated in the code behind file, in one or more nested methods, but only one method is responsible with the string output. That method is called by a property. The property is finally called by the ascx file, as showed in fig. 2. The ascx file only contains the typical heading, may contain some javascript reference useful for the expected output, and the above mentioned property call. Below you can see a fragment of code that generates html content. It is not easily readable and for this reason can be split in other methods, but is highly maintainable as it appears only once in the whole application.



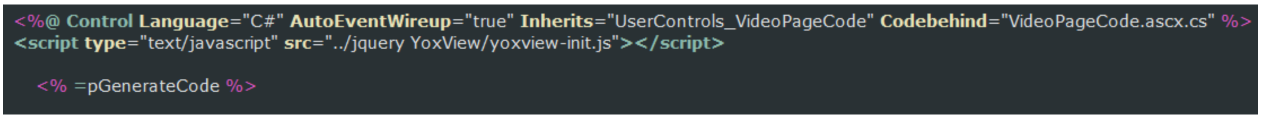
```
else
{
   resultCode += @"<h1>"
      + rowTR_Anag[0]["TRAN_Title"] + @"</h1>"
      + rowTR_Anag[0]["TRAN_Descr"]
      + @"<asp:Panel ID='Panel1' runat='server' style='position:absolute'>
   <p style='padding-left:20px; padding-top:15px; width: 240px; position:absolute'>" + nrOfVideos + @" videos available</p>
</asp:Panel>
   <table border='0' cellpadding='0' cellspacing='0'>
      <tr>
         <td style='padding-right:4px; width:247px'>
         <div class='yoxview' id='videos'>
         <a href='" + "http://www.youtube.com/embed/" + row1[0]["TRAR_VideoID"] + "?wmode=opaque&amp;rel=0&amp;showsearc
               + "<img src='Images for html/Temp Images/go to videos.png' alt='' title='Part 1' width='241px' height='185' />";
}
DataRow[] rowsFrom2;
rowsFrom2 = table.Select("TRAR_PartNum > '1'");
int count = 0;
foreach (DataRow row in rowsFrom2)
{
   if (rowsFrom2[count]["TRAR_VideoHost"].ToString() == "v")
   {
      resultCode += @"<a href='" + "http://player.vimeo.com/video/" + rowsFrom2[count]["TRAR_VideoID"] + "?wmode=opaque&bylin
         + "<img src='' alt='' title='Part " + (count + 1) + "' />";
   }
   else if (rowsFrom2[count]["TRAR_VideoHost"].ToString() == "y")
   {
      resultCode += @"<a href='" + "http://www.youtube.com/embed/" + rowsFrom2[count]["TRAR_VideoID"] + "?wmode=opaque&a
         + "<img src='' alt='' title='Part " + (count + 1) + "' />";
   }
```

Fig. 3: A fragment of the code behind (the GenerateCode method) that generates the “go to videos” link to videos (see fig. 1).



```
public string pGenerateCode
{
   get { return GenerateCode();}
}
```

Fig. 4: The property that returns the string containing the html code generated in GenerateCode the method.



```
<%@ Control Language="C#" AutoEventWireup="true" Inherits="UserControls_VideoPageCode" Codebehind="VideoPageCode.ascx.cs" %>
<script type="text/javascript" src="../jquery YoxView/yoxview-init.js"></script>

  <% =pGenerateCode %>
```

Fig. 5: The property call in the user control ascx file. This is the only code needed here.

3.3 Detail page

The detail page (aspx file) is just going to contain the references to the user controls and eventually some content that is common to all the detail pages.

4 Conclusions

This paper shows a real-world example for letting a website grow in size with the contributions and efforts of its users. This methodology goes behind the forum style of building content. It is more similar to a CMS like WordPress, but open to a community of users. The layout has some restrictions that, if needed, can be overcome with some more programming work in order to give the users more freedom in building custom content.

The range of applications of the presented methodology can be very wide, going from information content to video, from pictures to sound, from social networks to science. Anyone can give her/his contribution to a site's content, virtually with unlimited possibilities, using this custom CMS functionality.

References

[1] Adding User Controls to a Web Forms Page: http://msdn.microsoft.com/en-us/library/5d0t5fak%28v=vs.71%29.aspx

Francesco Capuzzo
University Lucian Blaga of Sibiu
Faculty of Sciences
5-7 I. Ratiu str., Sibiu, 550021
ROMANIA
Email: francesco.capuzzo@gmail.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Multitouch Vector Graphics Shared Mobile Whiteboard

Ioana Culic
Teacher Coordinator: Alexandru Radovici

Abstract

Tablets are becoming important in education. This paper present a software designed to replace the old blackboards and also improving them. It is a multitouch whiteboard, which can help teachers and students collaborate. It allows them to create presentations in an interactive way. The software runs on multiple devices, which can be connected, thus the result will be a board on which every user can draw or add text and the result will appear on every screen. The elements can be easily manipulated, what makes the application suitable for use in schools. Each figure can be resized, moved, or it can have its color changed without affecting the other elements on the board. The education field can benefit a lot from this application.

1 Introduction

The world of today is all about speed and mobility. As people spend most of their time away from home, the needs for a mobile device, which can help them store information and keep in touch with the everydays changes, have increased.

In the early 1970s the idea of a personal computer aroused. A ”personal, portable information manipulator” was imagined by Alan Kay at Xerox PARC in 1968. The purpose of such a device was to provide the population a computer that could be carried anywhere so that people wouldnt be dependent on their personal computers in their homes. After the laptops, the netbooks followed. These are more lightweight, economical, energy-efficient and are especially suited for wireless communication and Internet access.

In April 2010, Apple released the iPad, which innovated the market with its reduced dimensions and the multitude of tasks it can accomplish. A tablet computer, or a tablet, is a mobile computer, larger than a mobile phone or personal digital assistant, integrated into a flat touch screen and primarily operated by touching the screen rather than using a physical keyboard. Such a device offers many advantages, like:

- Usage in environments where a keyboard is difficult to access;
- Lighter weight;
- Image manipulation is easier, more precise and intuitive due to the touch environment than using a mouse;
- It is easier for some users to use a stylus, pen or finger to point and tap on objects, than to use a mouse;
- Current tablets typically have longer battery life than laptops or netbooks.

On the other hand, the tablets have weaker capabilities:

- Screen risk, as tablet computers are handled more than conventional laptops; in addition, their screens are also used as input devices what makes them indispensable;

- Limited battery lifespan;

- Lower processing power than a personal computer.

Thus, it is of no surprise that the iPad surpassed even the most optimistic of projections to define a brand new product category and become the best-selling gadget of all time. As tablets have caught on with consumers, the education market is also starting to consider using them.

This project presents an iPad application, which aims at helping both students and teachers in the teaching process. It can replace the classic blackboard. In order to use this application, each student and teacher should have a tablet, which can communicate, consequently, enabling the communication between them. In addition to this, tablets current shortcoming (limited multitasking) could be their greatest asset in education, as it forces students to focus on one task at a time.

2 Computer Tablets in Education

Tablets represent a means through which people can access enhanced e-books featuring images, video and audio, elements which are impossible to include in print or in a standard e-book. These make the process of reading and studying more interesting and interactive. People get their attention caught by the colored elements, the buttons, the photos, or even the videos a document stored on a tablet can feature. This is why these devices are extremely suitable for educational uses. Due to their increased interactivity, the tablets can engage students more effectively in the educational process.

In the last few years, Apple has shown a lot of interest in introducing their iPads to the education field. Firstly, the App Store offers a great variety of applications designed specifically for students and teachers. Students can use apps in order to track their homework, take notes and study for exams. In addition, teachers can give lessons, monitor progress and stay organized simply by owning a tablet which has all this applications. What is more, the free iTunes U [App12b]app gives students access to all the materials for their courses, all stored in a single place. Right in the app, they can play video or audio lectures, read books and view presentations, see a list of all the assignments for the course and check them off as they are completed. Also, when a message is sent or a new assignment is created, students receive a push notification with the new information.

In order to create a course, teachers can use the iTunes U Course Manager. This offers them the possibility to include in their courses a syllabus, handouts, quizzes, and other items. All of the uploaded course materials are hosted by Apple and are available to anyone taking the course. Teachers can pull content and links from the Internet, iBookstore, App Store, and iTunes Store, or they can gather material from among the 500,000 plus resources at iTunes U, including audio and video content from museums, universities, cultural institutions, and more. Once the course is ready, it is extremely easy to be distributed to anyone who is interested in the topic.

Moreover, by allowing students to highlight text, take notes in the margin and access a dictionary directly within the book itself, the tablets can easily replace the classic blackboard and textbooks. As a result, a number of whiteboard softwares have already appeared (Easiteach[Eas12], NotateIt[Not12], Dabbleboard[Dab12]). All these allow people to share information, especially in classes, where it is easy for the teacher to draw something and share it with his students. These softwares act just like a whiteboard, but they offer the person using them the possibility of choosing between a various range of tools, thus making it easier for them to express an idea. On the other hand, besides their advantages, these types of softwares are quite limited regarding the connectivity.

3 Multitouch Vector Graphics and Sharing

The application presented in this paper aims at solving this problem as its main characteristic is the fact that everything is done on multiple devices which are connected. What is more, this software is not restrained in connecting with only a certain platform. Consequently, it would be of a great use in classrooms as it would increase the interactivity.

The main improvement in comparison to other similar softwares of this application is the fact that it uses vector graphics[Web12]. The result is that the elements can be manipulated. This is important in a classroom, as the examples can be shown in a real time. The teacher will be able to insert text and different shapes and figures, some of them are predefined and others can be designed on the spot. Moreover, the shape, size, position, background and foreground colors of the elements can be changed. This will happen without the need of erasing the previous elements that were drawn on the whiteboard. In addition to this, the user also has the possibility of erasing only certain figures or text elements, leaving the others unchanged.

As everything happens on a tablet, the software also presents the advantage of supporting multitouch gestures. In contrast to other whiteboards, here, shapes can be drawn using two, or even more fingers. This makes it much easier and faster to draw. This is not the only advantage of the fact that this application runs on a tablet. Portability is another important factor. Tablets are light and it would not be difficult for each student to carry one on them during their classes.

In addition, by using this software, both parts, the teachers and the students, would have the possibility to contribute actively to the teaching process. Unlike the usual situation, in which the students are simply watching what the teacher is drawing, by using this software, the students can also write or draw on their own devices. This is possible as all tablets can be connected and each user is able to see and interact with the board. The result would be a more interactive and engaging course.

This is done by using the eXtensible Messaging and Presence Protocol (XMPP)[SAST09]. By using this protocol, the users are not stuck with a certain server. It also offers the users the possibility to store in their application a list of contacts with whom they will share information. This list can be manipulated; contacts can be added and removed. In addition, a server is not required in order to connect. One of the users devices will act as a server. Users will also be able to subtract some privileges from other users they are connected with, for example, the teacher has the ability to choose whether what a student draws on the board can be shared with the others or not.

4 Technology

The application runs in iOS 4[MNL11] and it has been written in Objective C[App12a]. It uses the XMPP protocol in order to communicate between devices.

Objective C is an object oriented programming language. It is used to write programs for Apple's Mac OS X and iOS. Objective C is an extension of the C programming language, thus it includes all its libraries. As its name suggests, Objective C is built around objects and it is based on message passing to object instances. In other words, a message telling the object to apply a method is sent to each[ML09] [PP11].

Another characteristic of Objective C is that it has no garbage collector. The programmer has to take into consideration memory management. This can be done by using the mechanism of reference counting. The advantage is that there is less memory used as each object is released when it is of no more use. iOS 4 operating system was released on the 21st of June 2010. The new features it brought were the multi-tasking technology[NWM10] which allows multiple applications to run in background. The XMPP is an open-standard communications protocol, which uses XML messages for communicating. The advantage is the fact that the messages can be understood both by people and the computers, enabling their debugging.

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Graphical simulation of chemical reactions

Farcaş Nicolae
Teacher Coordonitor: Lect. PhD. Camelia-M. Pintea

Abstract

The paper illustrates an interesting approach to chemical experiments for beginners. The software used is Scratch. The scripts are detailed and exemplified with images. There are used different objects as sprites and costumes. Several variables and scripts for passwords and movement issues are also included.

1 Introduction

One of the most practical platform for learning programming structures is Scratch[1]. The software provides drag-and-drop technique which is one of the high quality of software. Beginners can learn quickly programming, in an interactive way, constantly viewing their graphical application.
In the current paper is described an application simulating several chemical reactions.

2 Scratch environment

The working space in Scratch has four distinct areas, each one with well defined properties. In the left side of the screen there are the controls and scripting modules; in the upper part of the central area are included the scripts, costumes and sounds used in the program and in the lower side are sprite sequences and backgrounds used in the program. In Fig. 1 is illustrated the menu and some actions easily accessible using the buttons from the toolbar.

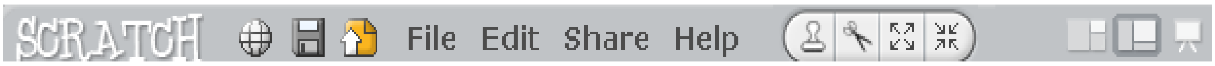


Fig. 1 The bar menu and tools

Fig. 2 and Fig. 3 illustrate the work areas. In Fig. 2 are shown the right side with the tools and the central area where the scripts are created using the tools and where are changed costumes and sounds.

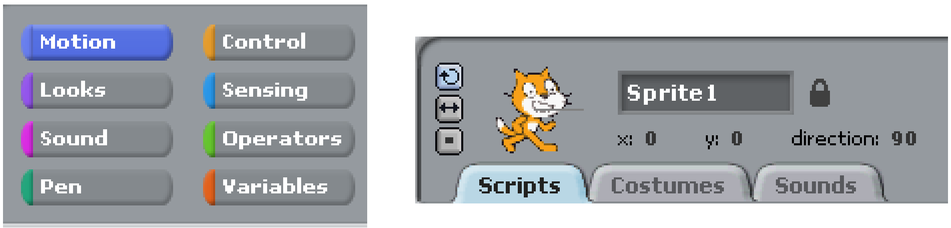


Fig. 2 The left and central area of the work space in Scratch

In Fig. 3 is illustrating the working area with the actors, the backgrounds and buttons and on the right side there are the green flag as the starting button and the red button as the stopping button.

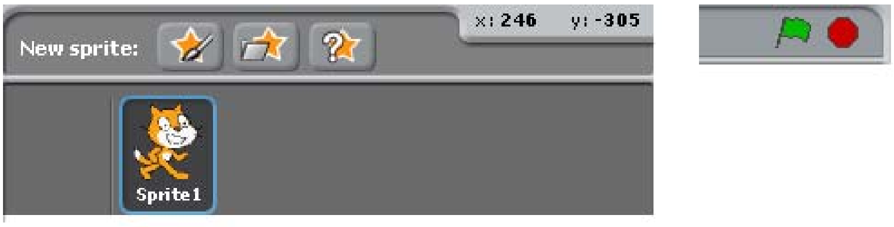


Fig. 3 The right side of the workspace in Scratch

Scratch is based on a series of scripts that involve actors, costumes, various backgrounds and objects. In Fig. 1 are shown several sets of commands and objects that can be used in scripts.

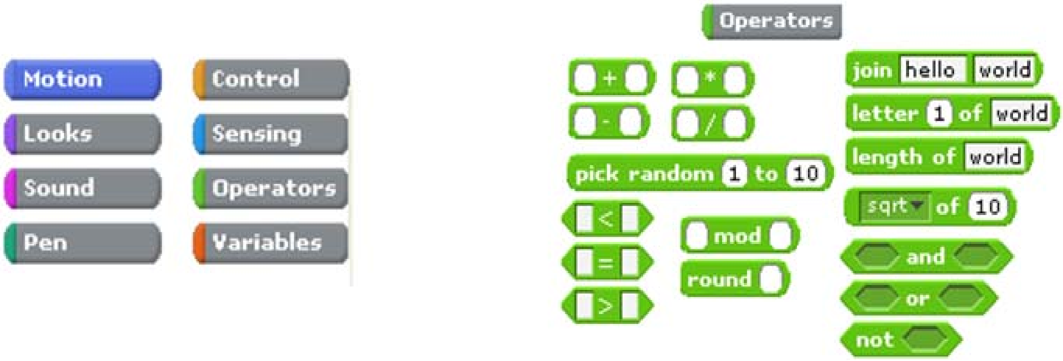


Figure 4 Controls and operators in Scratch

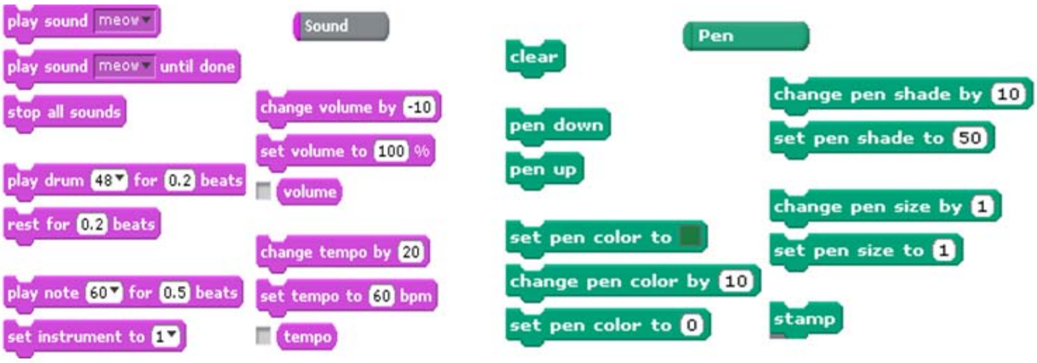


Figure 5 Ilustration of sound commands and drawing instruments

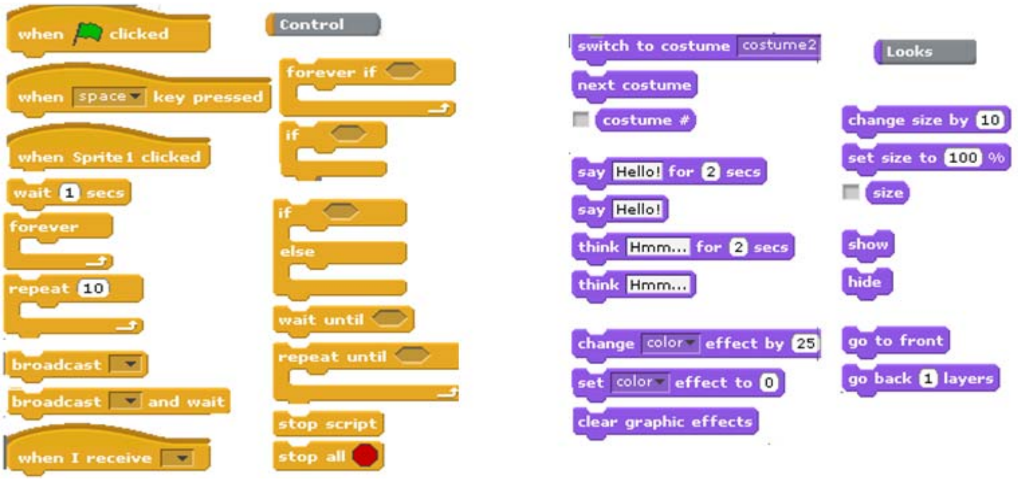


Figure 6 Ilustration of control commands and image commands

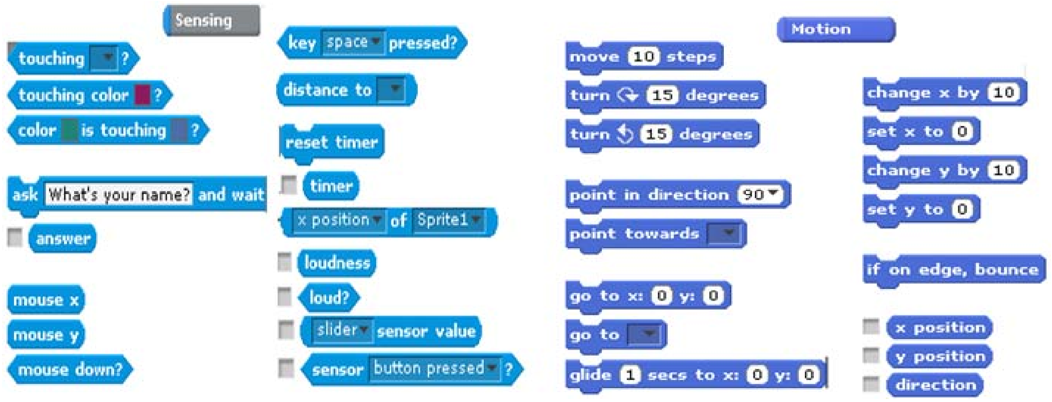


Figure 7 Illustration of controls for sensing and motion

3 The chemical simulator

The application simulates several chemical experiments. The chemical substances used are symbolic. The application emphasis how the chemicals reacts and the way the chemical substances reacts when are heated.

a.”Boiling” a chemical substance

The way a chemical substance is „boiled” is detailed in the following script. When you press the tube and the tube heats, after 5 seconds the substance will boil; just about 10 seconds after, it will return to its previous state.

```
When Sprite1 clicked
If touching Flaskheater
   Wait 5 secs
   Switch to costume pink stuff1
Wait 10 secs
Switch to costume pink stuff
Stop script
```

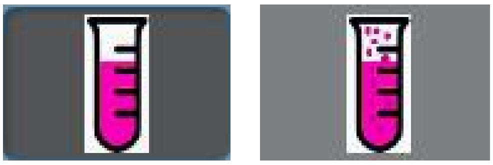


Figure 8 A substance before and during boiling

b.A chemical reaction

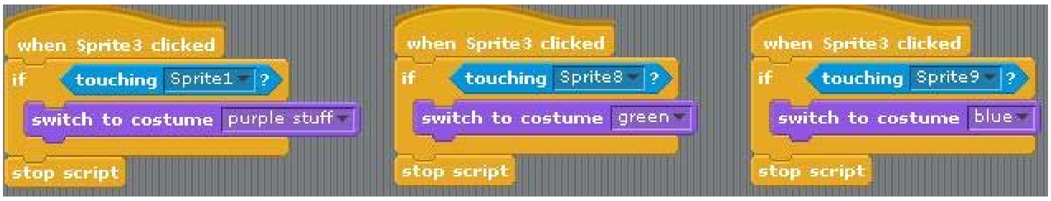
Using the following script we will simulate a chemical reaction. As long as the tube is toucing the heater , the tube will change its color as an effect of a chemical reaction. Furthermore, when we take away the tube from the heater, after one second all the graphical effects will be canceled so the reaction stops.

```
When Sprite1 clicked
If touching Flaskheater
  forever if touching Flask heater
   change color effect by 25
else
   wait 1 secs
   clear graphic effects
 Stop script
```

```
When H key pressed
Switch to costume Flaskheater
Repeat until key S pressed
    Change color effect by 25
Switch to costume costume1
Stop script
```

The script used for heating is simple but effective. Pressing H-key will change the image of the heater to heater-on and its color will be change also with some Scratch effects; pressing S-key simulates the fire burning. Pressing S-key again the costume will be changed with heater-off image.

c. Chemical combinations



```
when Sprite3 clicked
if touching Sprite1 ?
  switch to costume purple stuff
stop script
```

```
when Sprite3 clicked
if touching Sprite8 ?
  switch to costume green
stop script
```

```
when Sprite3 clicked
if touching Sprite9 ?
  switch to costume blue
stop script
```

Figure 9 Chemical reactions

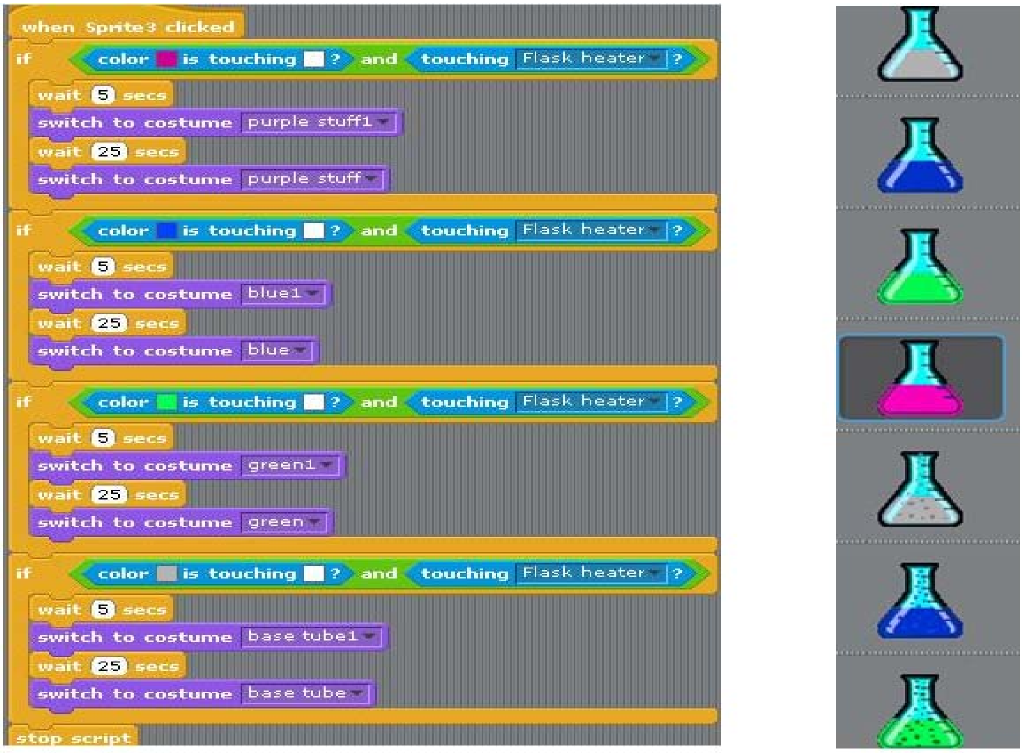
The scripts from Fig. 9 and Fig. 10 controls the combinations of substances based on sensing: whatever a sprite is touched by another the costume of the first one is changed, simulating in this way a chemical combination.

d. "Magic" chemistry

We try to recreate in a fun way some cartoons heros. Thus, when you press T-key, will appear some surprising elements; when they will reach the boiler, the secret ingredients will lead finally to our cartoon heros. The script from Fig. 11 shows the way it is used the password and how the surprise element is revealed.

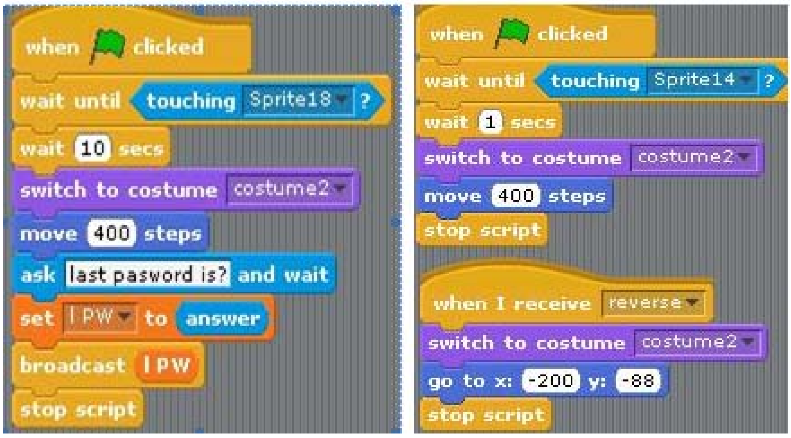
4 Conclusion

In recent years visual techniques have an important role in the computer work of young people around the world. Scratch environment is a user-friendly environment for the beginners in computer graphics. The application from current paper is a model for interactive chemical experiments.



```
when Sprite3 clicked
if < <color 🟪 is touching ⬜ ?> and <touching Flask heater ?> >
    wait 5 secs
    switch to costume purple stuff1
    wait 25 secs
    switch to costume purple stuff
if < <color 🟦 is touching ⬜ ?> and <touching Flask heater ?> >
    wait 5 secs
    switch to costume blue1
    wait 25 secs
    switch to costume blue
if < <color 🟩 is touching ⬜ ?> and <touching Flask heater ?> >
    wait 5 secs
    switch to costume green1
    wait 25 secs
    switch to costume green
if < <color ▨ is touching ⬜ ?> and <touching Flask heater ?> >
    wait 5 secs
    switch to costume base tube1
    wait 25 secs
    switch to costume base tube
stop script
```

Figure 10 Chemical Combinations



```
when ⚑ clicked
wait until <touching Sprite18 ?>
wait 10 secs
switch to costume costume2
move 400 steps
ask last pasword is? and wait
set I PW to answer
broadcast I PW
stop script
```

```
when ⚑ clicked
wait until <touching Sprite14 ?>
wait 1 secs
switch to costume costume2
move 400 steps
stop script

when I receive reverse
switch to costume costume2
go to x: -200 y: -88
stop script
```

Figure 11 Using passwords to access and show the surprise element

Bibliography

[1] http://wiki.scratch.mit.edu/wiki/Main_Page
[2] http://scratch.mit.edu

Farcaş Nicolae
Tech Univ Cluj-Napoca.
North Univ Center Baia-Mare
Str. Victoriei, 76
ROMANIA
E-mail: nfarcas@mail.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Object recognition in OpenNERO

Filip F. Filipov
Teacher Coordinator: Milena Lazarova

Abstract

The paper presents a machine learning classifier who is based on supervised learning and naive Bayers. Primitive characteristics of objects and agents in a game platform are used for feature vectors. The characteristics are extracted from snapshots in the game through image filters. All agents and objects are spawned randomly, both position and look direction, at each game start. Several classifier based on different object's features are presented in the paper. Each of the feature vectors is selected in view of its discriminative abilities for the targeted objects. Experimental results for evaluation of the classification accuracy points out the direction of further improvement of the feature space in regard to higher recognition rate of the selected objects.

1 Introduction

Recognition is a classical computer vision problem. In the tutorial of Francis Bach & Jean-Yves Audibert[1] many different examples are given on how supervised learning unsuitable for universal recognition is. The paper aim and objective is to construct a machine learning classifier for a fixed environment with objects in it. The environment will be the game platform OpenNERO and the objects – the agents in the game and the environment obstacles such as trees. The paper structure is as the following: Section 2 describes the algorithm used for processing, extracting features and object classification. Section 3 describes the experimental results. Section 4 is a conclusion from the results.

2 Algorithm

The input data as shown in Fig. 1 comes from a snapshot in OpenNERO. Then primitive features from this snapshot are exacted and fed to the machine learning classifier.

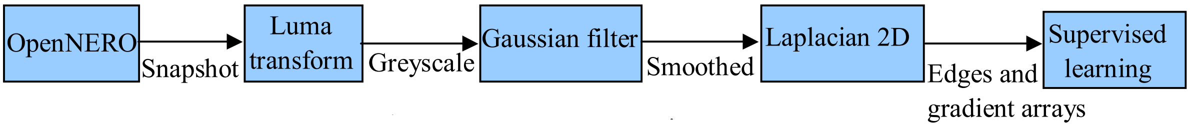


Fig. 1 This are the implemented steps in the algorithm. An object label is produced as a result

2.1 OpenNERO

It is a open source Python written platform based on the work of the Neural Networks Research Group and Department of Computer Science at the University of Texas at Austin and the project Neuro-Evolving Robotic Operatives (NERO) game[2] .OpenNERO also includes examples and demos of Stuart Russell and Peter Norvig's textbook "Artificial Intelligence: A Modern Approach" [3]. A Chapter of this book is about perception and the early vision image-processing algorithms who extracted only primitive features such as edges and regions from images. The cs343vision2 branch of OpenNERO allows a Nero agent to walk in the Nero world and take snapshots of objects which are spawned randomly. These objects are a tree (Fig. 2), a cube (Fig. 3) and two agents Sydney (Fig. 4) and Steve (Fig. 5)

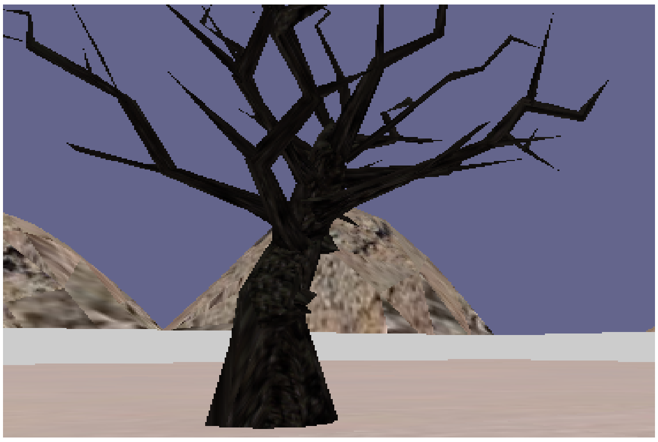


Fig. 2 This is a snapshot of the tree in the game

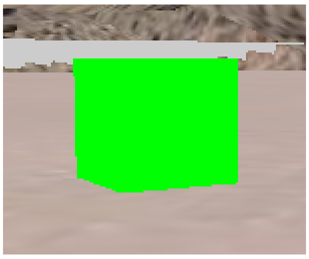


Fig. 3 This is another game object – a cube

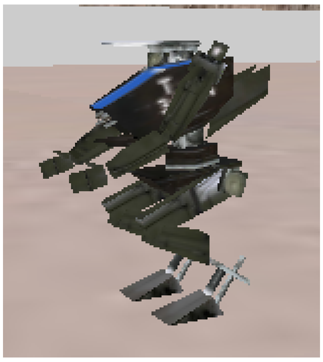


Fig. 5 This is Steve the robot like agent

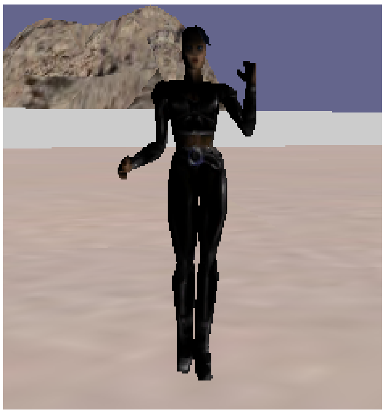


Fig. 4 This is Sydney

2.2 Stages

2.2.1 Luma transform

As first step the RGB color image is translated to a greyscale image with the Python build in ITU-R 601-2 luma transform.

$$L = R * 299/1000 + G * 587/1000 + B * 114/1000 \qquad (1)$$

2.2.2 Gaussian filter

Then in order to reduce the edge sharpness and get better edge results the image is smoothed with a two dimensional Gaussian filter with sigma = 1.

$$G(x, y) - \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad (2)$$

2.2.3 Laplacian

As third step the gradient for each pixel is calculated and then thresholded producing an array with edge pixels and the gradient for each pixel. This is done with the build in edge filter in the numpy and scipy packages for Python. They use a two dimensional Laplacian filter :

$$\begin{matrix} -1, & -1, & -1, \\ -1, & 8, & -1, \\ -1, & -1, & -1, \end{matrix} \qquad (3)$$

2.2.3 Machine learning classifier

The fourth step is the machine learning classifier whose purpose is to identify known objects based on edge pixels and their gradient when walking in the Nero environment. This step is implemented with supervised learning with naive Bayer's, since this method makes the naive assumption that each probability is independent.

In supervised learning [3] you have feature vector/s -'x' and a target label - Y:

$$\begin{matrix} x11 \;\; x12 \;\; x13 \rightarrow Y1 & \\ x21 \;\; x22 \;\; x23 \rightarrow Y2 & \\ : & F(xn) = Yn \qquad (3) \\ : & \\ xn1 \;\; xn2 \;\; xn3 \rightarrow Yn & \end{matrix}$$

The aim of the machine learning would be to study all features and the resulting Y on a known set of data in such way that the resulting function can help predict a target label, when there is a new 'x' which wasn't part of the training.

It is common practice in supervised learning to have a training set, test set and final validation one. The test set is used to measure the accuracy of the classifier for the chosen feature vectors and prevent overfitting, while training, because the trainer has to find an equilibrium between data fitting and function complexity.

3 Experimental results and analysis

Several test cases were made with different feature vectors and different probability smoothing for each set of vectors. In all cases the feature extraction area (Fig. 6) wasn't the whole snapshot but a part of it, since the objects are always on the ground and in-front of the viewer.

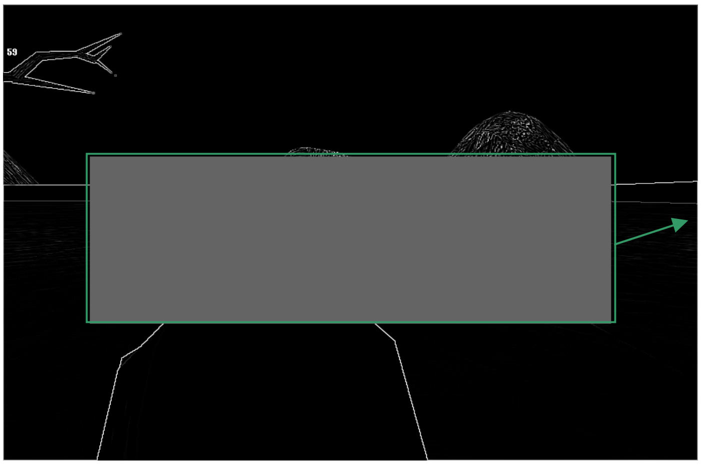


Fig.6 Shows the extraction area

In the first classifier build the following feature vectors were chosen: sum of all pixels that have color value different than 0, sum of all pixels from the first vector who had horizontal gradient and sum of all pixels from the first vector who had diagonal gradient. For the probability the average value of all vectors for all objects was calculated and depending on how near to this value the value the particular vector in the snapshot was a probability value was assigned to it. Since the cube had always the smallest average value, the tree the highest and Sydney and Steve almost the same this classifier always preferred the cube or the tree.

In the second attempt the probability estimation was changed. It was calculated for each snapshot vector how many values from the training set the particular object vector had above the snapshot value. If they were more than the half then the sum of them would be subtracted from the total number. Then the new number would be divided by the total number.

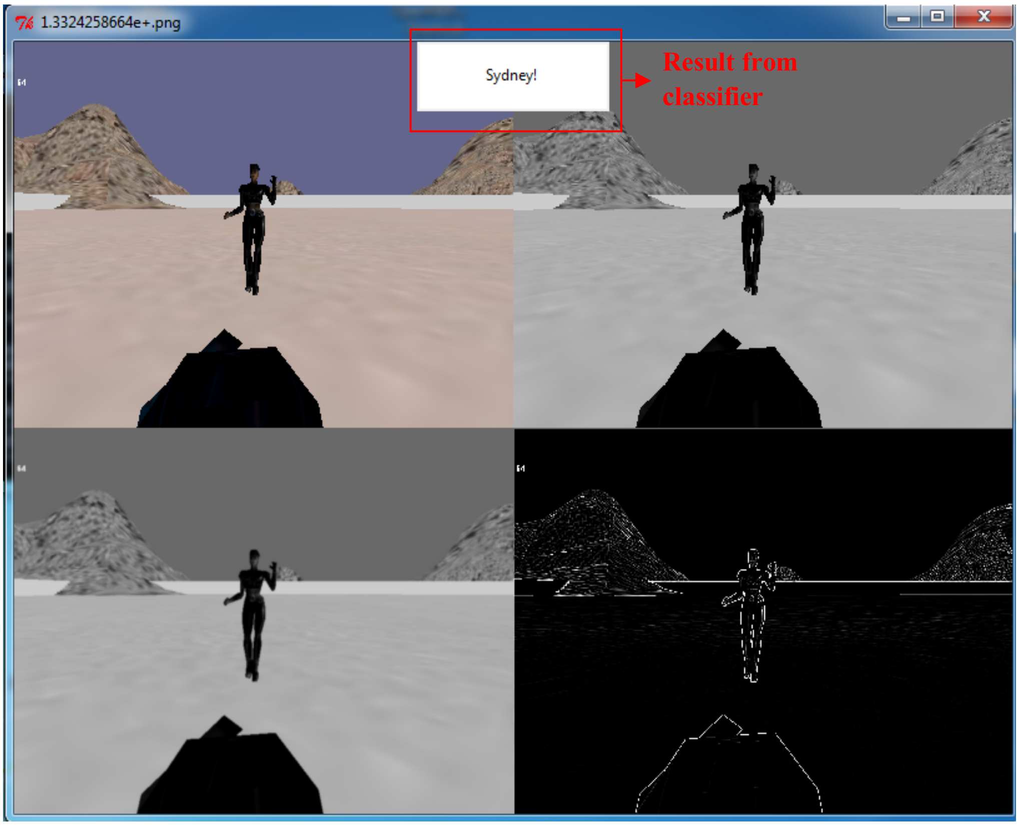


Fig. 7 Shows the results form the snapshot and how it was processed

As last step which for all attempts is valid, all vectors for each object would be multiplied and the maximum from this object values would be chosen as the object in the snapshot. This classifier had good results but depended heavily on the snapshot environment since he learned characteristics about the specific instance of the environment rather than the objects.

In the third attempt the sum of all pixels with color value equal to 255 as first feature vector was gathered. The second and third gathered from the first vector the sum of the pixels that had horizontal or vertical gradient. The probability estimation remained the same as in the second attempt.

From the third version of the classifier the precision table 1 was for formed. Ten snapshots

(Fig 7.) of each object were taken in total forty.

	Sydney	Cube	Steve	Tree	Accuracy
Sydney	10	0	0	0	100%
Cube	0	10	0	0	100%
Steve	2	0	7	1	70%
Tree	0	0	0	10	100%
Error	0%	0%	30%	0%	

Table 1 Shows how many times a object was labeled right and how many wrong

4 Conclusion

The build classifier has a good precision but still produces wrong classification. Since the aim of the paper is the classifier future work will include experimenting with different image-processing algorithms for different feature vectors in which the features of Steve and Sydney will be bigger.

References

[1] Francis Bach & Jean-Yves Audibert . *Supervised learning for computer vision: Theory and algorithms* , http://www.di.ens.fr/~fbach/eccv08_fbach.pdf

[2] Neuro-Evolving Robotic Operatives (NERO) game, http://www.nerogame.org/.

[3] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach* , Prentice Hall Dec 1, 2009

Filip F. Filipov
Technical University of Sofia
Faculty of Computer Systems
1000 Sofia, blvd. Kl. Ohdiski 8
Bulgaria
E-mail: pilif.pi.lif@gmail.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Optimization of Parallel Implementation of Quicksort

Dimitar Gadzhev
Teacher Coordinator: Associate Professor, PhD Milena Lazarova

Abstract

The paper discusses parallel computations of Quicksort algorithm. Algorithm Quicksort is one of the fastest sorting algorithms, but for very large data sets the speedup of sorting on high-performance computing platform might be relevant. The parallel implementation of the algorithm increases its speed, but fastest sorting is not always guaranteed in cases of poor load balancing of the concurrent tasks. Several optimization techniques are suggested in the paper in order to increase the efficiency of the parallel implementation of the Quicksort algorithm.

1 Introduction

Sorting algorithms are used to arrange elements of different sets according to certain condition. Sorting may require a lot of computations and time especially when large amount of data are processed. The performance of the sorting algorithms depends on several factors among them the number of elements, the initial sequence of the elements and even the values of the elements.

Quicksort is the fastest known comparison-based sorting algorithm [1]. It is a recursive algorithm and has a worst-case running time of O (N^2) but is often the used in practice for sorting because it is average efficiency [2].

The current trends of hardware development and innovations are oriented towards extensive usage of high-performance computations based on multicomputer and multiprocessor computer systems. Grid and cloud computing also pose the requirement for distributed data processing [3].

Parallel sorting algorithms and parallel implementations of sorting are required in order to Speed Up the data processing.

The parallel implementation of the Quicksort algorithm based on divide and conquer approach increases its speed, but fastest sorting is not always guaranteed in cases of poor load balancing of the concurrent tasks. Several optimization techniques are suggested in the paper in order to increase the efficiency of the parallel implementation of the Quicksort algorithm based on ideas used for parallel implementation of a variety of sorting algorithms [4][5].

2 Quicksort algorithm

The sorting algorithm complexity and the time required for sorting are determined by the number of iterations to be executed so that to order a certain set of values. The algorithm's running time depends on the number of the elements to be sorted. The worst case running time of in place sorting algorithms is O (N^2). Some sorting algorithms have complexity of O ($N.\log_2 N$) which is considered as better asymptotic time [2]. The Quicksort algorithms average case expected running time is O ($N.\log_2 N$) even if its worst case is as slower as O (N^2).

2.1 The divide-and-conquer strategy

One of the key features of Quicksort algorithm is the model of sorting which it uses that is based on the strategy "divide and conquer". According to this model the set of data to be sorted is divided into sub-sets and the same sorting algorithm is then applied on each sub-set. The sub-sets comprises data with different values and the process of allocation of data to the sub-sets and the further division of the sub-sets leads to the sorting of the initial set.

The utilization of the "divide-and-conquer" paradigm in the operation of the Quicksort algorithm is an important feature that can be used in order to Speed Up the algorithm's processing time. Each of the two sub-sets that are created in the process of division of a set in two parts can be processed independently from the other. Thus the new sub-sets are simultaneously sorted without affecting the results of the algorithms.

2.2 Prerequisites for parallel execution

Nowadays not only the supercomputers' architecture features many core processors but most of the desktop and laptop computers, some smart phones and even other embedded computing devices are equipped with multi core CPUs. This is a prerequisite for the successful utilization of more than one processing thread in the software applications. High performance computer platforms with multi processor and multi core architecture enable simultaneous execution of several tasks. The parallel implementation of the Quicksort algorithm is facilitated by the existing possibility for independent processing of sub-sets of data as well as the option to run different tasks simultaneously.

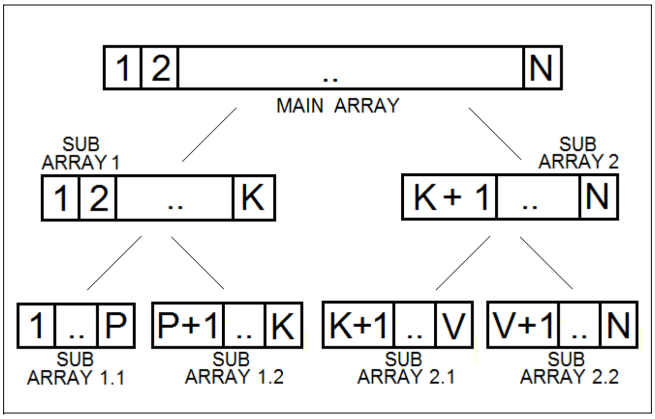


Fig. 1. Division of the set and the sub-sets into smaller sub-sets of data

On Fig. 1 an example of the division of a set and the sub-sets is given together with the construction of smaller sub-sets is presented. The basic data set "Main array" is divided into two sub-arrays, “Sub array 1” and “Sub array 2” that can be processed independently of one another. From these two sub-arrays for new data sub-sets can be obtained: “Sub array 1.1”, “Sub array 1.2”, “Sub array 2.1” and “Sub array 2.2”, which again can be processed independently of one another. Independent processing of sub-arrays can be done in parallel provided sufficient number of processor units.

3 Parallel implementation of the Quicksort algorithm

Due to the divide-and-conquer paradigm used in the operation of the Quicksort algorithm it allows independent processing of the generated sub-sets thus making the parallel algorithm implementation straight forward.

The parallel implementation of the Quicksort algorithms has two stages. In the first stage the sub-sets are created, which will later be processed in parallel. The construction of the sub-sets based on general sequential operation of the algorithm. The set division continues until the number of created sub-sets becomes greater than or equal to the number of the available processor cores that will be used for the concurrent processing of the sub-sets. Furthermore for the division of a sub-set parallel processing can also be used. At the second stage of the parallel Quicksort algorithm the generated sub-sets are processed utilizing the available hardware CPU cores. Each sub-sets is processed following the sequential algorithm.

In order to implement the described parallel processing for the Quicksort aimed at multithreading programming model OpenMP API is used [6]. Using the parallel loop execution the following program code for multithreading model of Quicksort can be suggested:

```
omp_set_num_threads(Number_Of_Threads);
for (i = 0; i < Number_Of_Threads; i++)
{
  QuickSort(PositionsArray[i][0], PositionsArray[i][1]);
}
```

4 Experimental results of the parallel Quicksort

The OpenMP multithreaded implementation of the Quicksort algorithm is used for experimental evaluation of the performance parameters of the parallel sorting. In order to be able to estimate the scalability of the algorithm in view of the problem size and parallel workload sets comprised of different number of elements are used – 50 to 500 mln. elements. The experimental computer platform is a server with CPU 8 core Intel Xeon @ 2.2 GHz, 4 GB RAM @ 800 MHz, 2x160GB Hitachi SATA HDD and OS Scientific Linux 5.4. Thus the scalability of the parallel algorithm is evaluated as depending on the number of CPUs used for the computations – 1 (sequential processing), 2 and 8 processing cores. The experimental results for the execution time are given in Table 1 and the profile of the execution time is presented in Fig. 2.

The ratio of sequential processing time and the execution time required for the parallel sorting of given set determines the Speed Up of the parallel execution:

$$Speed\ Up = Sequential\ Execution\ Time\ /\ Parallel\ Execution\ Time \qquad (1)$$

The experimental results for the Speed Up are presented in Table 2 and the Speed Up profile depending on the parallel workload and the multiprocessor platforms are given in Fig. 3.

From the results in Table 1 and Fig. 1 obtained for the time required for parallel execution of the Quicksort algorithm some observation about the algorithm performance can be made. In some cases the algorithm sorts quickly bigger set of elements – this situation is clearly observed in the processing time for 350 and 400 million values. The reason for these cases is due to the uneven distribution of the elements in the creation of the sub-sets. This implies the requirement for further optimization of the parallel Quicksort algorithm with the purpose of better load balance of the processing workload.

Table 1. Results for the execution time of the sequential and parallel Quicksort algorithm

Number of elements (millions)	Execution time (sec.)		
	Sequential (one processor core)	Parallel (two processor cores)	Parallel (eight processor cores)
50	18.18549	16.61226	7.545379
100	38.25491	50.27498	30.80737
150	57.97577	47.08829	21.7399
200	78.82135	75.52103	40.76056
250	100.2268	96.89994	61.71374
300	119.4986	115.9806	62.55289
350	141.8761	133.4652	62.04608
400	162.0338	125.8847	43.45787
450	182.8315	172.2939	92.5854
500	204.4282	171.9106	69.0126

Table 2. Speed Up of the parallel Quicksort

Number of elements (millions)	Speed Up	
	Parallel (two processor cores)	Parallel (eight processor cores)
50	1.094703	2.410149
100	0.760913	1.241745
150	1.231214	2.66679
200	1.043701	1.933765
250	1.034333	1.62406
300	1.030333	1.910361
350	1.063019	2.286624
400	1.287161	3.728527
450	1.06116	1.974733
500	1.189154	2.962187

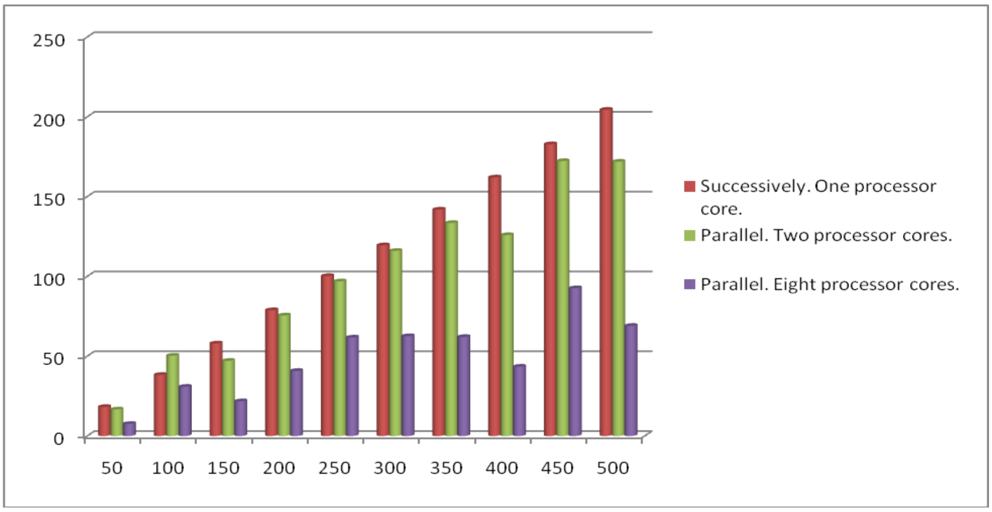


Fig. 2. Profile of the execution time of the parallel Quicksort algorithm

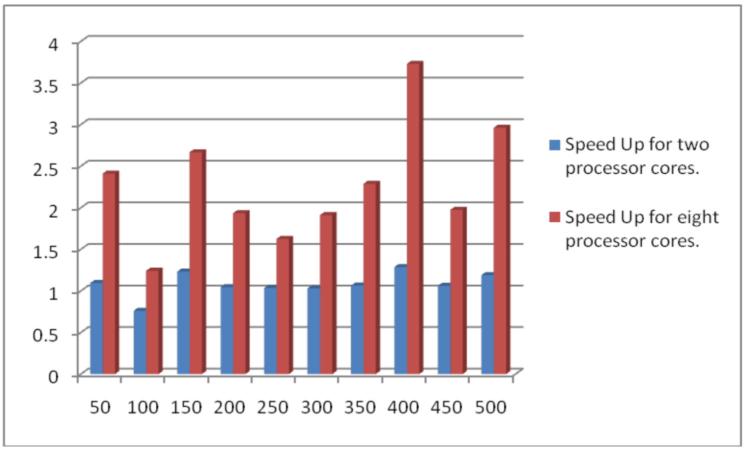


Fig. 3. Profile of the Speed Up of the parallel Quicksort algorithm

5 Optimization of the parallel Quicksort algorithm

As described in the previous section based on the experimental results gathered using parallel OpenMP multithreaded Quicksort algorithm in some cases the sorting is faster for larger number of elements and slower for less values in the set. These results can be explained by the fact that in dividing the set into sub-sets, each of the generated sub-sets may comprise different number of elements leading to unbalances workload for the processing units of the hardware platform. When the number of the elements in one of sub-sets is big enough that its size is comparable to that of the initial set, then the execution time for parallel sorting of multiple sub-sets is close to the sequential processing time. That’s way the load balancing of the parallel workload is of crucial importance for the performance of the parallel Quicksort algorithm.

An example of not optimized and not balanced load of the sorting with much bigger processing time of one of the sub-sets is given on Fig. 4. Parallel processing time T_p is close to the sequential processing time T_s, which reduces the Speed Up of the sorting.

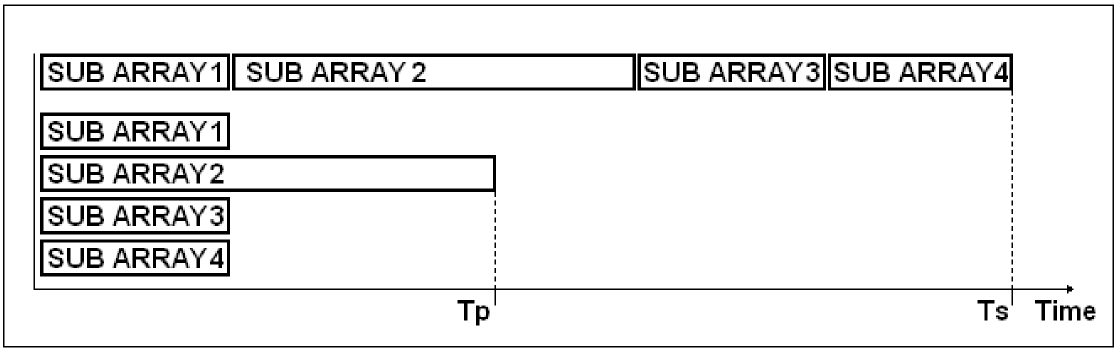


Fig. 4. Not optimized parallel processing

In order to improve the performance the slower processing time of a sub-set has to be decreased that can easily be achieved by reducing the amount of the sub-array. For this purpose the sub-arrays are divided into smaller sub-arrays that will optimize the load on the processor cores and thus reduce the execution time. Similar example is presented in Fig. 5, where the largest sub-array is divided into two sub-arrays and each is processed by different processor core. As a consequence the parallel processing time is reduced from T_p to T_{p1}, which accelerates the parallel algorithm execution.

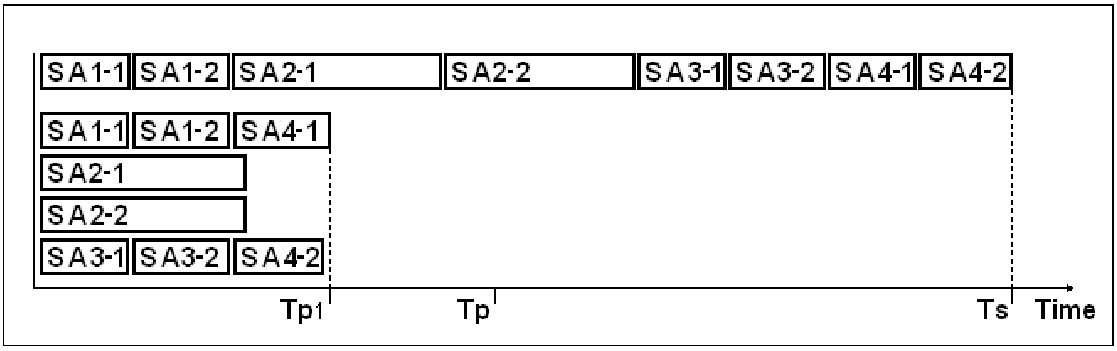


Fig. 5. Optimized parallel processing

Although the additional division of the sub-sets into smaller sub-arrays, it is possible that one or more of them will still have large number of elements. To ensure that the parallel processing time does not converge to the sequential processing time the constructed sub-arrays should be small enough – the number of the elements in each sub-set should be approximately equal to or less than the quotient of the total number of elements and the number of processor cores:

$$\textit{Required number of elements} \leq \textit{Total number of elements / Number of processor cores} \qquad (2)$$

Besides the division of the sub-arrays until each of them is small enough, they need to be handled in such a way as to balance the load of the processor cores. For this purpose it is necessary that the sub-arrays with large number of elements are processed before the others. In case the sub-arrays with more elements are retained for processing last of the sub-sets, the algorithm performance will degrade due to the bigger execution time of one or more processors responsible for the largest sub-sets, which would slow the overall performance, as shown in Fig. 6a. In cases where the sub-arrays with more elements are sorted in the beginning, the parallel workload of the processor cores will be better balanced and will result in shorter time delay in the Quicksort algorithm, as shown in Fig. 6b.

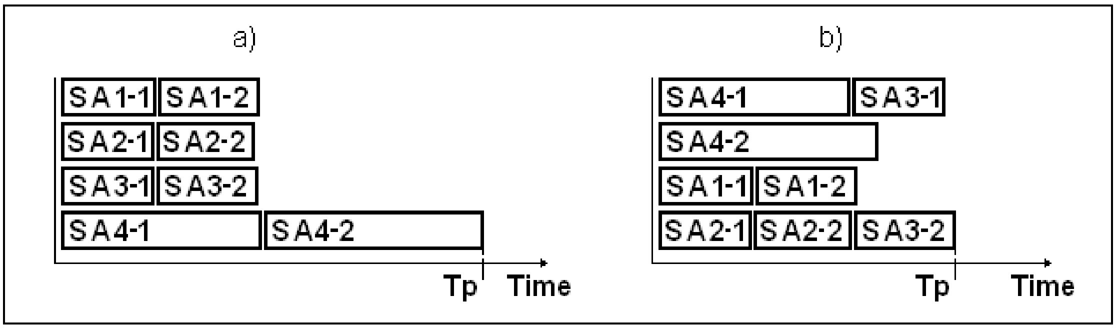


Fig. 6. Additional optimization of the parallel processing

6 Conclusion

The utilization of a parallel algorithm for "Quicksort" can increase its performance that is particularly required in the sorting of very large data sets. Due to the nature of the Quicksort the use of divide-and-conquer paradigm is the most appropriate choice for the parallel computation of the sorting. The experimental results presented in the paper are based on parallel multithreaded OpenMP implementation of QuickSort and are aimed at evaluation of the scalability of the parallel Quicksort in respect of the number of the elements in the data set and the parallel computational platform. As the result shown the divide-and-conquer parallel Quicksort has good speed up and scalability, but the performance parameters are influenced by the load balancing of the concurrent tasks. Thus in case of unbalanced partitioning of the data set the linear speed up is not guaranteed. To increase the efficiency of the parallel computations of the algorithm an optimization in the division of the initial set into sub-sets is suggested in the paper. The size of the sub-sets and the order in which they are handled are important factors influencing the performance of the parallel algorithm that can further decrease the time required for sorting of extremely large data sets.

References

[1] Knuth D., Art of Computer Programming, Volume 3: Sorting and Searching, Addison Wesley, 1998.
[2] Cormen Th., Ch. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, MIT Press, 2009.
[3] Buyyaa R., C. Yeoa, S. Venugopala, J. Broberg, I. Brandic, Cloudcomputing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, Journal Future Generation Computer Systems, Elsevier, Vol. 25, No. 6, June 2009, pp.599–616.
[4] Borovska P., Synthesis and Analysis of Parallel Algorithms, Technical University of Sofia, 2008, ISBN: 978-954-438-764-4.
[5] Grama A., A. Gupta, G. Karypis, V. Kumar, Introduction to Parallel Computing, Addison Wesley, 2003.
[6] The OpenMP API specification for parallel programming, www.openmp.org

Dimitar Gadzhev
Technical University of Sofia
Dept. Computer Systems
8 Kliment Ochridsky blv.
BULGARIA
E-mail: dimityr.gadjev@gmail.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Intelligent reasoning agents used in marketing strategies

Valentina LAZĂR
Teacher Coordinator: Lecturer PhD Florin Stoica

Abstract

The project aims the marketing process from the moment the customer interacts with marketing environment until the construction of new marketing strategies to enable sales and gain the customer’s loyalty, qualities absolutely necessary for the existence and profitability of business. That for, in developing process it was concluded that the use of applications based on threads is more expensive in terms of query time, effort and updating user data, than if it would use a system based on intelligent agents.

All this led to the development of an application based on intelligent agents that is easy to use and excludes the detailed reports and cumbersome.

1 Introduction

Given that we live in a world in constant motion, a world where time is the main problem, the present application is built precisely as an alternative to this problem in the banking field. It is known that in banks, as large firms, the size of the database is a reasonable size, therefore, browsing and extracting data representing their interests is a problem [1][2]. Thus, the present application comes with the following improvements: not only that the data is strictly filtered to obtain what it seeks, in this case that speaks about marketing on the Internet (E-Marketing), data is taken on the market where the company operates, generating a double advantage: market knowledge and improvement of the marketing plan. [8][9]

The first approach to bank marketing evolution is performed by Philip Kotler [7]. It involves five steps, which were identified according to the diversity and effectiveness of marketing techniques implemented.

- The first stage is designed as a bank marketing technique of advertising and sales promotion, image banks (providing small gifts, but useful: umbrellas, flashlights, pens.). The emphasis is on friendly atmosphere within the bank.[5][6]
- The second phase of banking marketing is defined as relaxed smile - officers are trained how to have a more open attitude towards the customers. It is rearranged so that the interior of banks have a more pleasant atmosphere and the outside is "humanized". Benevolence has ceased to be the deciding factor in choosing a bank.
- In the third stage of development: marketing segmentation and upgrading banking means. Banks have discovered a new competitive tool when they began to segment their markets and develop new products for each target market segment.[2]

What keeps the bank's ability to develop new products continuously determines the position of market leader. Marketing has entered the insurance and investment services. Innovations have resulted in widening globalization of financial markets and banking business through the provision of less traditional services and the use of information technologies and computer networks.[10][11]

At the fourth stage, marketing means bank positioning. Banks realize that no commercial bank is able to provide all products and be the best for all customers. It concludes that a bank must "occupy a specific position, taking account of its possibilities. [3][21]

2 Client user interface design

The project begins with the realization of user interface, and more specifically the Web Site[13] [14] [17].The first point is to be sought for interface design. To motivate customers to view the site presented is necessary, from choosing colors for presentation to the selection information to be presented.

Such colors were chosen that did not look tired (shades of blue, gray and white) and the information was compressed to avoid losing the essence when making customer interest to the attention of those present.

Another way to attract customers is interactivity and data is presented that not much effort is required on the part of customers to reach the desired information.[8]

One can easily see how the environment and efficient way scroll to the desired benchmark. This presentation is especially necessary if speaking of a world in constant motion in time is most precious.

2.1 User Interface as marketing component

In terms of marketing, creating and developing a web site should have to support the existence of specific needs, specific, related in most of the way marketing communication takes place to the bank.[22]

It is important to note that in general, the project does not create products, does not set prices, does not ship products to the recipient and does not make the bank's reputation double and triple the sales. The website can contribute significantly to the manner in which the organization makes the exchange of information or marketing environment, mainly with customers (and potential) and its competitors.

In the first case it is about providing a volume of information attractive enough to cause a favorable reaction to clients: marketing buying or ordering products (loans) promoted or at least requesting additional information about them.[24]

In the second case it is about differentiating the bank in relation to its competitors in general and especially in terms of marketing communications.

Interface [16] [30] [32] provides access from the first page of this information by following the user to view what you are looking for.

Also is easy to understand what they are made for, from the first page of this information, namely that, especially for banking institutions, the legislation is very thorough and contains clauses which are omitted, possibly resulting in the existence of disputes between customer and bank, even the loss of credibility to customers.

2.1.1 Marketing strategies that result from database [30]

The application itself is built to develop marketing strategies to differentiate between firms in the market, i.e. between banks.

This starts from the idea of knowledge by understanding the market segment of its characteristics: the size of this segment, placing the physical market, etc.

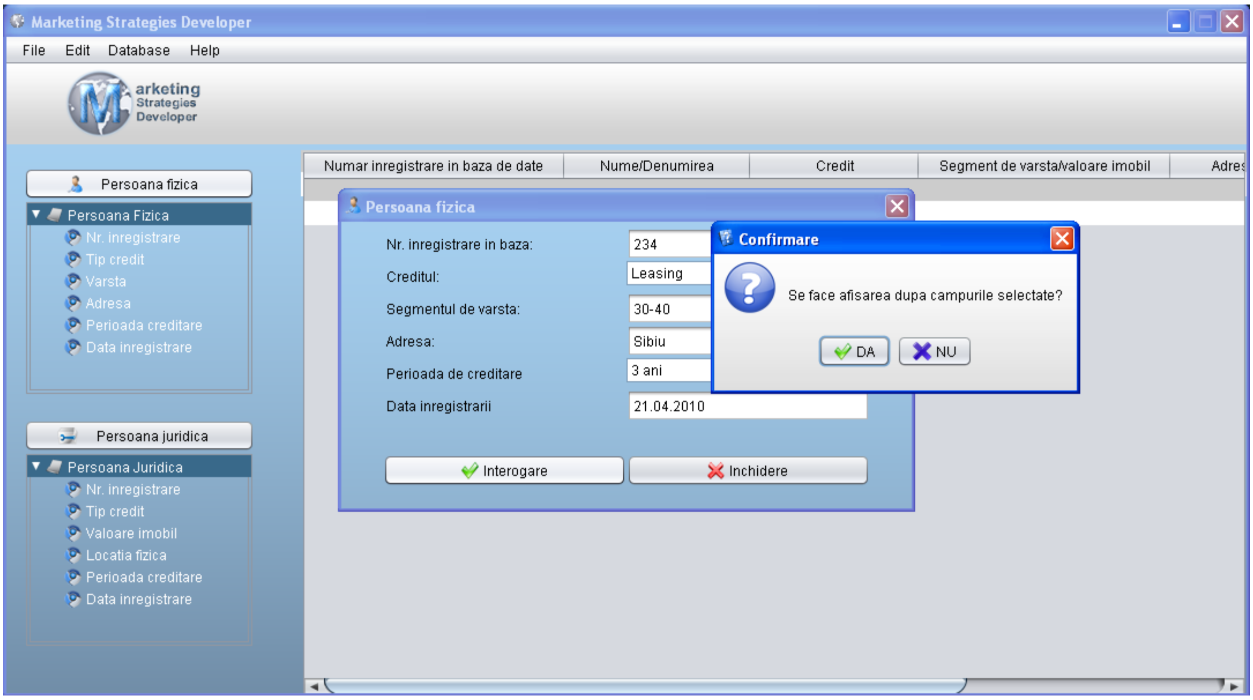
To know this requires a market study and that study is better than their own resources.

Therefore intended to provide some queries in the database is recorded data from customers. This process is the result of a program in Java programming language that allows data to obtain a result of queries.

Thus was born the MDC, that means Developer Marketing Strategies application.

If you wish to obtain a marketing report, you must follow several steps:

Step 1: After the opening of the application, the type is determined by the person who addresses the query.



Numar inregistrare in baza de date	Nume/Denumirea	Credit	Segment de varsta/valoare imobil	Adres

Nr. inregistrare in baza: 234

Creditul: Leasing

Segmentul de varsta: 30-40

Adresa: Sibiu

Perioada de creditare: 3 ani

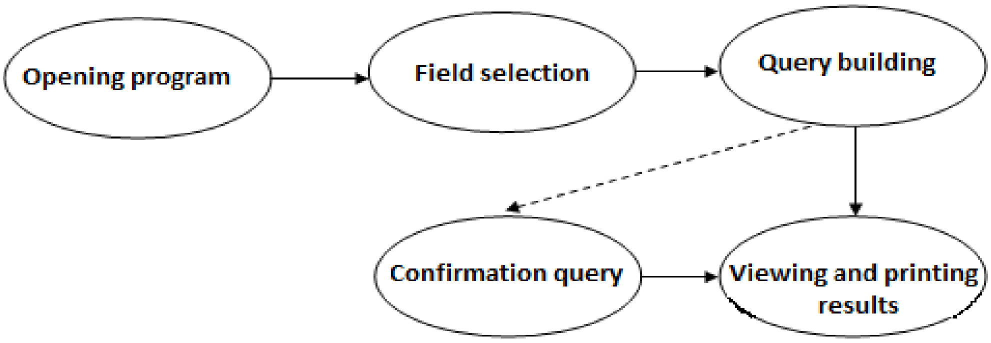
Data inregistrarii: 21.04.2010

Se face afisarea dupa campurile selectate?

Fig. 1. Confirmation of the current query..

Step 2: When the fields were generated in the table, you can choose Save As from the File menu option to save the results into a text document.

These steps can be schematically represented as follows:



It may be noted the ease of use of the application as well as that the user is assisted at every step to avoid mistakes from queries.

2.1.2 Marketing reports

The core of any marketing information system is the repository for the data center. They contain solid information on all customers and potential customers. For the sake of brevity, this paper uses the generic term for a specific site database where information is stored. In addition, to gain maximum advantage by exploiting this information it is necessary for each system to provide reliable and documented data on targeted markets and customers.

Applications for marketing uses information from the database of clients, but there are some features of this system functions:

- Making simultaneous queries in multiple tables;
- Queries made by a serial number placed in the database;
- Displaying tabular results of queries;
- Enabling the rescue of the results;

The main advantage of the customers is that it provides a comprehensive picture of business relations with a client or group of customers. This overview allows a deeper understanding of consumer behavior and market in general, a point of departure, among others, marketing research, especially in studying the market.

The main characteristic of the system lies in the possibility of handling large volumes of information from various sources, internal and external, in order to develop models of market segments or commercial behavior. Analysis reports can be quickly generated, modeled and saved for future use, and business plans and scenarios improved and refined.

3 A new approach - intelligent agents

When architecture development concluded that the use of applications based on threads is more expensive in terms of query time, effort and updating user data, unless they would use a system based intelligent agents.[34][35][36]

So, it was concluded that it is better to develop five intelligent agents:

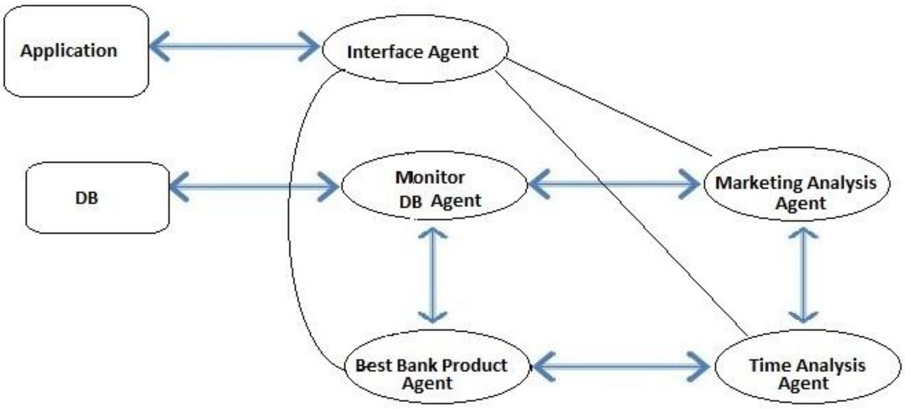


Fig. 2. Agencies working with the database

As can be seen from the figure above, the application has three-tier architecture with agents embedded on each tier:

- **Data tier** – the Monitor DB Agent.
- **Application tier** (business logic) – Best Bank Product Agent, Time Analysis Agent, Marketing Analysis Agent.
- **Presentation tier** – the Interface Agent.

In the following we present a short brief of agents functionalities:

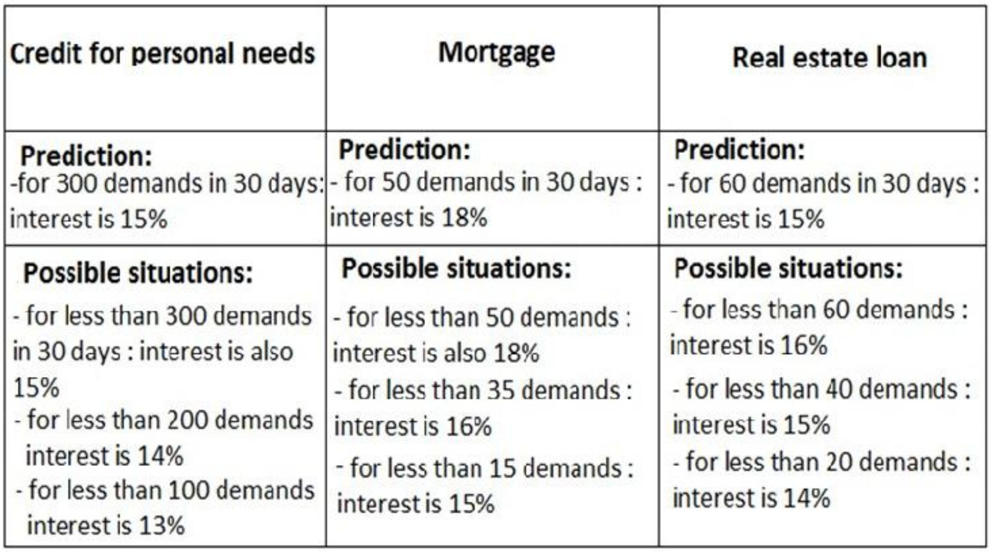
- ➢ The Interface Agent collects information from other agents and brings it into the application interface;
- ➢ DB Monitor Agent monitors inputs, outputs and changes in the database;
- ➢ Product Best Agent Bank determines which product is most used;
- ➢ Analysis Marketing Agent determines increased or decreased interest as recorded in DB for a period of time;
- ➢ Time Analysis Agent monitors travel time from the last inspection.

The four agents mentioned above are active throughout the existence of the database. They continuously monitor the database and interface with the agent in topical bring results.

It can be seen that the need to achieve ease of use and exclusion of detailed reports as cumbersome.

Regarding the Marketing Agent Analysis, it should be noted that it establishes increased or decreased interest by receiving from the DB Agent Monitor the number of credits in a period.

For example, let's assume the following situation:



Credit for personal needs	Mortgage	Real estate loan
Prediction: -for 300 demands in 30 days: interest is 15%	**Prediction:** - for 50 demands in 30 days : interest is 18%	**Prediction:** - for 60 demands in 30 days : interest is 15%
Possible situations: - for less than 300 demands in 30 days : interest is also 15% - for less than 200 demands interest is 14% - for less than 100 demands interest is 13%	**Possible situations:** - for less than 50 demands : interest is also 18% - for less than 35 demands : interest is 16% - for less than 15 demands : interest is 15%	**Possible situations:** - for less than 60 demands : interest is 16% - for less than 40 demands : interest is 15% - for less than 20 demands : interest is 14%

The agent will known which is the best decision so that the bank will have profit and to have the highest number of clients.

Those fluctuations can be realized without a human agent intervention.

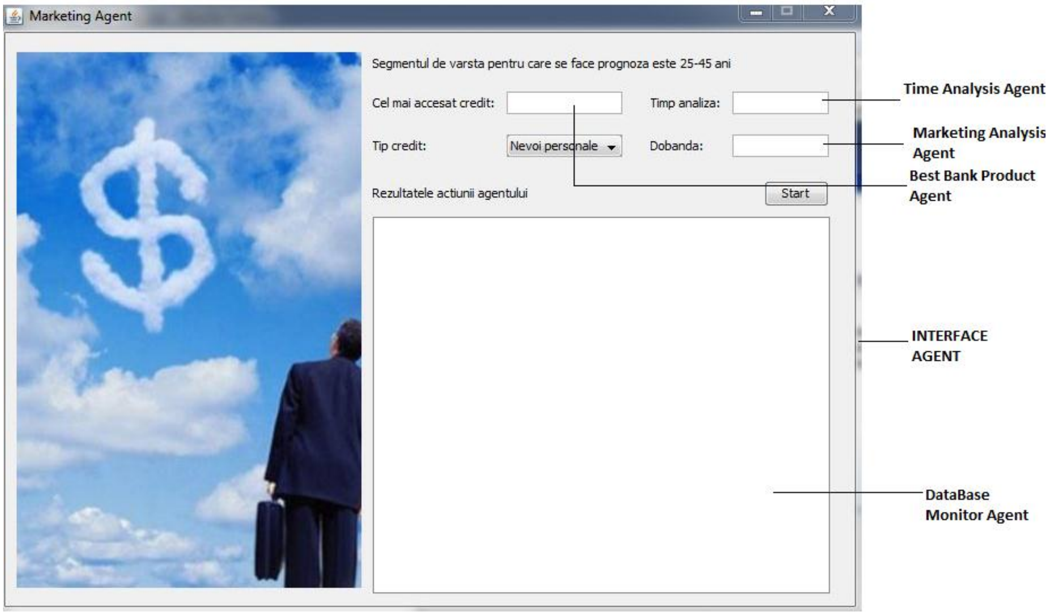
3.1 Application design

The development of the application is based on the idea of friendliness by using an interface presented in Figure 3.

An agent-oriented interface will display the results of all the other agents and they, also, can be activated sequentially or simultaneously.

A few **advantages** that an application based on intelligent agents has, comparing with the one based on threads:

1. Cooperation between agents: marketing analysis is made (*information is gathered from DB*) by an assistant agent which interacts (through ACL messages) with specialized agents: an agent who establishes, for example, increasing or decreasing rates, an agent which settles the most attractive bank product (loan), an JDBC-based *transducer agent* which serves as an interface between the database server and the other agents in the system, etc.



Segmentul de varsta pentru care se face prognoza este 25-45 ani

Cel mai accesat credit:

Timp analiza:

Tip credit: Nevoi personale

Dobanda:

Rezultatele actiunii agentului

Fig. 3. End-user interface of the multi-agent system

2. Another advantage is the continuous monitoring of the DB, which makes possible recording fluctuations and storing them for later reporting to the assistant agent. This is not possible using an application based on threads.
   Specialized agents are autonomous, and their life cycle is not related to the running time of the marketing analysis application. It may be recalled that these agents are independent of the application, so they can monitor the DB after stopping the application and even before starting the application.
3. Agents are reactive, providing information by request, and proactive, signaling when some conditions are accomplished.

3.2 Reasoning agents

The implementation of the business logic is based on a powerful inference engine provided by CLIPS.

The CLIPS inference engine is based on a very efficient algorithm, called 'Rete' and has a very brief syntax which allows defining models of very complex searches by using three main forms of data representation: facts, objects and variables

The business logic tier of the application involves chaining of IF-THEN rules to form a line of reasoning. The chaining starts from a set of input conditions (initial facts) and moves toward the provided conclusion - *forward chaining*. Thus, the reasoning agents from the application tier are based on CLIPS to provide decision support to end-user. The architecture of the application is described in Figure 4.

Because an expert system has a structure divided into two parts, one fixed, independent of the expert system - the inference engine, and one variable - the knowledge base, the behavior of the agents based on CLIPS can be changed updating only the external rules, without recompiling.

Another advantage of using CLIPS for Knowledge Representation is that it is much closer to natural language than an object oriented programing language [37].

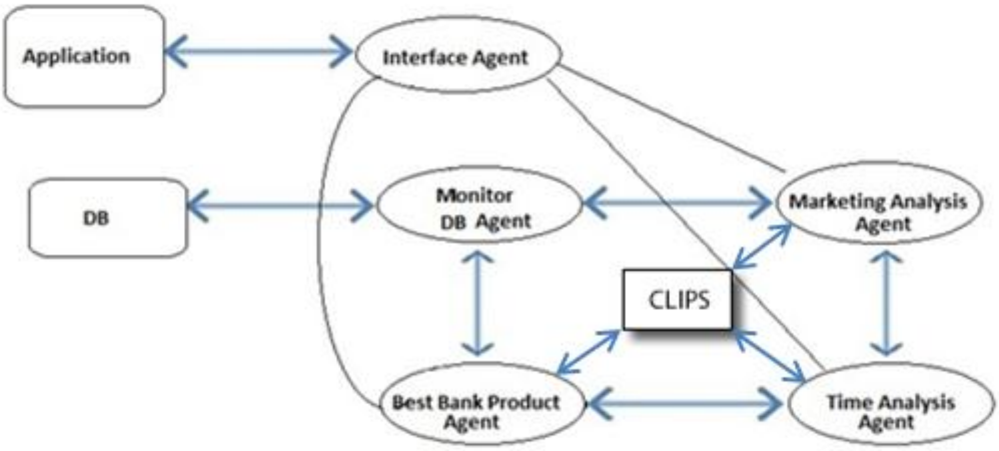


Fig. 4. Support Decision System using CLIPS

The agents are hosted by a JADE platform, and their interfacing with CLIPS was made using CLIPSJNI, a Java Native Interface for CLIPS.

4 Conclusions

In this paper is presented a multi-agent system that emulates the decision-making ability of a human expert, in the field of marketing strategies. Rule-based programming is one of the most commonly used techniques for developing expert systems. Our approach was to enhance native capabilities of JADE agents (autonomy, proactivity, reactivity, ontology-based communication) with reasoning facilities provided by powerful CLIPS inference engine and facile knowledge representation.

The three-tier architecture of the application is well suited for a marketing information system.

References

[1] Michael E. Porter, *Competitive Strategy: A manual for survival and growth of firms in market economy conditions* , Teora, Bucharest ,2007

[2] Michael E. Porter, *Competitive advantage: A manual for survival and growth of firms in market economy conditions*, Teora, Bucharest ,2007

[3] Cetină I., *Banking and Financial Marketing*, Economic Publishing House, Bucharesti, 2005

[4] Dumitru, L., *Banking and Financial Marketing*, Dimitrie Cantemir University Press, Târgu-Mureş, 2001

[5] Gheorghiu, L. , *Ways of promoting banking services in Romania, marketing view*, ASE Bucharesti, 2003

[6] Odobescu E., *Marketing of modern banking*, Sigma House, Bucharest, 2007

[7] Kotler Philip, Armstrong Gary. *Principles of Marketing.* Ed. a 3-a. Bucharesti: Teora, 2004.

[8] Cetina Iuliana, Odobescu Emanuel, *Bank marketing strategy*, Economic Publishing House, Bucharest, 2007.

[9] Matt Haig, *E-marketing guide*, "Rentrop & Straton" Publishing House, 2005

[10] Gay R, Charlesworth A., Esen R., *Online marketing, customer-oriented approach,* "All" Publishing House, 2009

[11] Strauss J., El-Ansary A., Frost R., *E-Marketing*, Pearson Education, Inc Upper Saddle River, 2003

[12] Ioan-Gheorghe Ratiu, Nicoleta David, *Working with databases in Java,* „George Bariţiu" University, Braşov

[13] David Powers, *PHP Solutions.Dynamic Web Design Made Easy*.pdf

[14] Brad Bulger, Jay Greenspan, David Wall, *Mysql/Php Database Applications*, 2Nd Ed - 2004 - Wiley.pdf

[15] David M. Geary, *Graphics Java 2 - Mastering the JFC,* 3RD Edition, vol. II, Swing

[16] Jason Hunter, (E-Book) Java Servlet Programming

[17] Andy Harrys, *PHP 6-MySQL Programming for the Absolute Beginner*.pdf

[18] Tricia Ballad, William Ballad, *Securing PHP Web Applications*.pdf

[19] Mircea Neamtu, *Distributed Programming Course*, ”Lucian Blaga”,University of Sibiu, Science Faculty 2010

[20] Ralf Fabian, *Course of Graphics and Image Precessing*, ”Lucian Blaga”,University of Sibiu, Science Faculty 2010

[21] E-marketing http://en.wikipedia.org/wiki/Internet_marketing

[22] Marketing Environment http://www.studentie.ro/Cursuri/Marketing/Curs-Marketing-MEDIUL-DE-MARKETING_i47_c1038_55121.html

[23] E-Marketing Strategies http://cerex.unitbv.ro/teze/rezumate/2010/rom/DajAlexisIon.pdf

[24] E-Marketing Plan , http://www.dotcom.ro/imarket.php

[25] Competition Analysis, http://www.scribd.com/doc/23835544/analiza-concuren%C5%A3ei

[26] Competitive Strategies http://www.preferatele.com/docs/marketing/4/analiza-strategica-a1.php

[27] Sales optimization techniques, http://www.newschannel.ro/categorii/vanzari-marketing/4/

[28] About safe online transactions http://www.talcioc.net/secure.php

[29] Advantages and disadvantages of PHP , http://kitsoftware.ro/Item/Framework-ul-PHP-care-sunt-avantajele-si-care-sunt-dezavantajele-lui/Section/Dezvoltare

[30] Working with the SQL database , http://www.ub-filosofie.ro/~solcan/minf/d06_07/minf.pdf#page=135

[31] About Web Technologies , http://www.w3schools.com

[32] Working with databases in Java http://software.ucv.ro/~aion/nad/jdbc.pdf

[33] PHP Web Security, http://www.slideshare.net/calin.iepure/securitatea-web-in-php-presentation

[34] Agent-oriented systems. Agent oriented analysis and design applications. http://profs.info.uaic.ro/~busaco/publications/articles/agenti2.pdf

[35] Florin Stoica, *Agent-oriented Systems Course*, ”Lucian Blaga”,University of Sibiu, Science Faculty 2011

[36] Valentina Lazar , *E-marketing strategies system* License Work, , ”Lucian Blaga”,University of Sibiu, Science Faculty 2010

[37] Florin Stoica, *CLIPS – Expert Decisional Systems Course*, ”Lucian Blaga”,University of Sibiu, Science Faculty 2012

Valentina Lazar
”LUCIAN BLAGA” UNIVERSITY OF SIBIU FACULTY OF SCIENCES
Adress: No. 5-7, Dr. Ion Ratiu st., 550012 Sibiu, Romania
E-mail: lyah_valy@yahoo.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Traffic Sign Detection and Recognition

Laura Nicoleta Mamina
Teacher Coordinator: Adrian Deaconu

Abstract

This paper describes a solution for detecting and recognising traffic signs. The proposed solution has two main steps: detection and recognition. For the detection part shape-based method and learning-based method are used. Shape-based method it is used for the restriction signs in daylight condition and for electronic restriction signs. Learning-based method it is used for both end of restriction signs and for the restriction signs.

It is used multi-threading for paralellizing the image processing.

For the recognition step it is used Support Vector Machine (SVM).

1 Introduction

One of the most important things when driving it is the safety. The Assistant Driving Systems can bring this safety to the driver. It can announce the driver about the traffic signs, if the car crosses the road line, it can detect the driver's face and attentionate him if it detects that the he is falling asleep.

This project is about detecting and recognising the traffic signs. In the literature there are three main approaches for detecting the traffic signs: color-based, shape-based and learning-based [1].
In this project, shape-based method and learning-based method are combined for obtaining a good accuracy.

I chose to combine these two methods because shape-based method proved to be very sensitive to the contour of the circles and it gave not the best results for the end of restriction signs because this signs have not a well defined contour like restriction signs have. The signs in dark areas were also not detected with shape-based method. Learning-based method proved to have a high accuracy in detecting those signs. But learning-based method gave not good results in detecting electronic restriction signs. Those signs might be detected by shape-based method.

For the recognition step, Support Vector Machine (SVM) and Neural Networks (NN) were compared and SVM proved to be better.

The remaining of this paper it is organized as it follows.
Section 2 describes the detection step; Section 3 describes the recognition step; Section 4 is about conclusions and future developments.

2 Detection

In literature there are three main approaches for detection step: color-based, shape-based and learning-based [1]. After implementing and testing on the same data set of images, I chose to combine shape-based method with learning based-method. One of the reasons was the speed of detection – the color-based method was very slow. Also, an important color-based method weakness is that the colors tend to be unreliable depending of the time of the day, weather condition, reflection, shadow, discoloration [1].

The goal of this project is to detect the signs in the figure 1. All of them are circle-shaped. After testing the shape-based method, I concluded that this method can easily detect the restriction signs in daylight, but it has problems when the signs are in dark areas (see figure 2) or if the sign it is an end of restriction sign (see figure 3). For detecting the circles it was used cv::HoughCircles from OpenCV library.

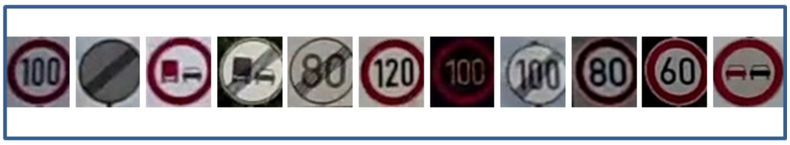


Figure 1

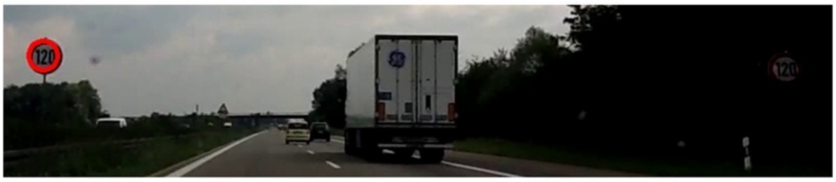


Figure 2

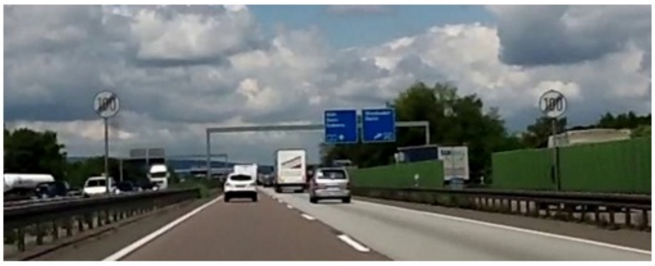


Figure 3 - Shape-based method detection

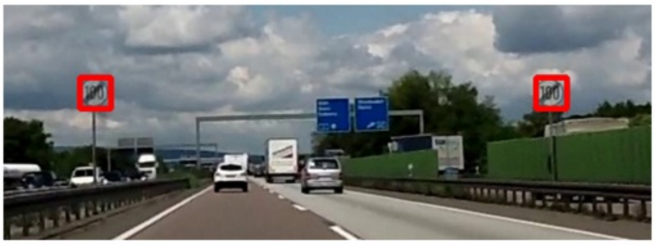


Figure 4 - Learning-based method detection

To resolve shape-based problems, learning-based proved to be a good solution. A detector was trained with end of restriction positive samples. It can detect the signs which shape-based couldn't (see figure 4). Another detector was trained for detecting restriction signs.

The regions of the image which are selected by the detector as traffic signs are analyzed and removed if they contain in a big proportion other colors than red, white or black.

Figure 5

In figure 5 there is an example of region selected by the detector. It was selected because it can be seen a dark oblique line there and this region was confused by the detector with an end of restriction sign. When it was analyzed, green was detected in a big proportion, so the region was eliminated from the signs list.

For training the detector it was used opencv_createsamples and opencv_haartraining.

Because there are two methods used for detecting the signs, processing the image with shape-based method and learning-based method is parallelized for speeding-up the detection step.

Also, the image which has to be processed it is split in two images and two different threads process in parallel each half of the main image.

Here is a pseudocode to illustrate the rationale behind the approach:

```
read the image
start Thread1
start Thread2
start Thread3
```

Thread1 detects restriction signs with learning based method. It splits the main image in two images. The current thread (Thread1) process the first half of the image and another thread process the second half of the image. Next are shown the steps for processing the image.

```
detect the signs with cvHaarDetectObjects
foreach sign detected
begin
    resize the detected sign
    reject the region if does not contain enough red, white or black, or if the
      region has another colors in big proportion
    convert the image to grayscale
    compute the descriptors
    recognise the sign
end
```

Thread2 detects end of restriction signs with learning based method; the process it is similar with the previous one, but here just black and white colors have to appear in big proportion.

```
detect the signs with cvHaarDetectObjects
foreach sign detected
begin
    resize the detected sign
    reject the region if does not contain enough white or black
    convert the image to grayscale
    compute the descriptors
    recognise the sign
end
```

Thread3 detects the electronic restriction signs with shape based method:

```
convert the image to grayscale
blur the image with cv::GaussianBlur
detect the signs with cv::HoughCircles
foreach sign detected
begin
    resize the detected sign
```

```
        reject the region if does not contain enough black or red
        compute the descriptors
        recognise the sign
    end
```

3 Recognition

In literature there are two main approaches for recognition step: Support Vector Machine (SVM) and Neural Networks (NN).

I chose to use SVM because in our tests it had better results compared with NN (90% SVM and 85% NN); also, the research for NN best parameters for training needs longer time. While for SVM the parameters needed for training are easily detected because it is a big difference between them, for NN there are more decisions to take: how many hidden layers we need and how many neurons for each hidden layer.

But NN was about 10 times faster in recognising the signs than SVM was and also the size that is occupied by the SVM classifier on the disk is approx. 5 times bigger than the size that is occupied by the NN classifier.

As parameters, for SVM training it was used the SVM type CvSVM::C_SVC and the kernel type CvSVM::LINEAR which proved in our tests to be the best choise.

For representing the signs, HOG (Histogram of Oriented Gradients) descriptors were used. OpenCV provides a compute method of HOGDescriptor class for calculating the descriptors.

For training the classifier it was used the train method of CvSVM class from OpenCV library and for predicting the class of a sign it was used predict method of the same class.

4 Conclusions and future work

Shape-based method it is a very fast traffic sign detection method, but it is very sensitive to the contour of the sign, so it cannot detect in a big percentage the end of restriction signs. Also it is unable to detect the signs in a dark area.

Shape-based method has a good accuracy in detecting restriction signs in daylight conditions.

Learning based method proved to have a high accuracy in detectiong both restriction signs and end of restriction sign. But learning based method needs a big database of signs for training. This signs have to be found in all possible conditions for a good accuracy (for example: blured, occluded, in raining conditions).

Testing NN and SVM, SVM proved to recognise more signs than NN did; But SVM occupies about 5 times more space than NN occupies.

In our tests, there were detected 129 signs out of 149 and there were 126 out of 129 correct recognised. The number of images used in testing was 88.

In a future work, the detection step it will be optimized for processing a picture in real time.

n literature there are two main approaches for recognition step: Support Vector Machine (SVM) and Neural Networks (NN).

References

[1] Karla Brkić, *An overview of traffic sign detection methods*, Faculty of Electrical Engineering and Computing, Zagreb, Croatia.

Laura Nicoleta Mamina
Transilvania University of Brasov
Applied Informatics
Str. Iuliu Maniu, nr. 50, Brasov
ROMANIA
E-mail: laura.mamina@xu.unitbv.ro

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

DCT based Digital Watermarking for Multimedia Content

Alin Manta, Arthur Mărginean
Teacher Coordinator: Ralf Fabian

Abstract

Traditional methods of ownership authentication, like stamping or signing, are getting today more and more impracticable when digital multimedia content comes to front. This paper presents a method of hiding invisible personalized information within digital multimedia material by watermarking. The technique applied here operates in the frequency domain and is based on the Discrete Cosine Transform (DCT). Experimental results show the influence certain types of attacks may have on recovering and detecting the hidden information. Conclusions and future work outline the opportunity of approaching the development of a specialized online service used for watermarking images. Among others, it's most important feature consisting in providing a free, automatic watermarking service. While the main experimental results were obtained by a MatLab application, a port on Java revealed although several partial results benefiting of a rich graphical interface which could increases the user interactivity.

1 Introduction

What exactly is "watermarking"? It could be better described as a transparent marking. This process implies the embedding of additional information in a document (may it be paper, text, audio and/or video signals, images) so that the item becomes traceable, or can be recognized, copyrighted, etc.

Depending on the application and procedure, the watermark can be for the human eye, either perceptible (e.g.: a logo on an advert, or the logo of a television, while broadcasting) or imperceptible. We shall discus the latter, because it is our particular interest to mark images so that we can identify them later.

Our target in this work is to insert information (in this case, a binary image) into a picture, so that the viewer will not be able to tell the difference between the original and the watermarked item. This mark has to be extracted by using a reverse procedure to the one which was used to insert it. Such a procedure can be used for example to identify the copyright owner or the one who created the item.

It would be ideal to be able to extract the mark without errors, even if the image has been altered. The amount in that the mark is recovered defines how well the watermarking process works. Virtually all marks can be removed eventually. Our interest is to obtain the best results when the image suffers common alterations as luminosity, hue and saturation, compressions (jpeg) and noise overlapping.

2 Rationale and Approach

2.1 Basic principles of embedding

Traditional methods of ownership authentication, like stamping or signing, are getting today more and more impracticable. Due technological advances, globalization and internet, copying and spreading of digital information has become easier than ever. Hence the question of protecting digital material from unauthorized copying, modifying ore distributing represents nowadays a technical challenge.

This paper presents a method of hiding invisible personalized information within digital multimedia material by watermarking. Digital watermarking is a process of embedding hidden information in any kind of digital multimedia content. Sounds, images and videos undergoing this process can be copied carrying the hidden information.

Most common watermarking techniques use the LSB (Least Significant Bit), the spread-spectrum or the quantization method. We will discuss the Discreet Cosine Transform method which presumes the embedding of bits of a binary image (the watermark) in some of the coefficients of the DCT of another image. The technique applied here operates in the frequency domain and is based on the Discrete Cosine Transform (DCT). The watermark pattern will be partitioned into blocks and the DCT technique is applied on each block for embedding and extracting the secret information.

Experimental results show the influence certain types of attacks may have on recovering and detecting the hidden information. Additional watermark strengthening approaches, like encryption or random noise, are outlined.

2.2 Discreet Cosine Transform

The Discrete Cosine Transform (DCT) allows image transformations form spatial to frequency domain and it often used in lossy image compression, e.g. in the JPEG standard. A major advantage is the fact that the transformation results are much better suited for image manipulation as the original colour components of in image. We shortly review the mathematical background of DCT.

The most common definition of DCT is for a 1D sequence of length N.

$$C(u) \quad = \quad \alpha(u)\sum_{x=0}^{N-1} f(x)\cos\left(\frac{(2x+1)u\pi}{2N}\right)$$

Here, f is the original function and $u = 0,1,\ldots,N\text{-}1$. All samples of f contribute the coefficient.
The invers transformation is defined similar:

$$f(x) \quad = \quad \sum_{u=0}^{N-1} \alpha(u)C(u)\cos\left(\frac{(2x+1)u\pi}{2N}\right)$$

for $x = 0,1,\ldots,N\text{-}1$. For both equations

$$\alpha(k) \quad = \quad \left\{ \begin{array}{ll} \sqrt{\frac{1}{N}} & \text{if } k = 0 \\ \sqrt{\frac{2}{N}} & \text{otherwise} \end{array} \right\}$$

Since our target in this paper is on digital images we will use the 2D DCT which is a direct extension of the 1D case. Let’s consider a block of size NxN, where N is a power of two. We may define the DCT as follows:

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

where alpha is the same as above and u,v = 0,1,…,N-1. Each image sample contributes to each coefficient and each pair (u,v) corresponds to a 'pattern' or 'basis function', values of the NxN pixel matrix.

Spatial samples can be recovered from the DCT coefficients by an inverse transformation defined as:

$$f(x,y) \;=\; \sum_{u=0}^{N-1}\sum_{v}^{N-1}\alpha(u)\alpha(v)C(u,v)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

where x,y = 0,1,…,N-1.

In order to reduce the amount of cosine function computations we can rewrite the direct transformation as matrix multiplication considering a transformation matrix T of the form

$$T_{i,j} = \left\{\begin{array}{ll} \frac{1}{\sqrt{N}} & \text{if } i = 0 \\ \sqrt{\frac{2}{N}}\cos\left[\frac{(2j+1)i\pi}{2N}\right] & \text{if } i > 0 \end{array}\right\}$$

with i and j reaching from 0 to N-1. Denoting by M the initial pixel matrix and by T' the transpose of T, we may write:

$$C = TMT'$$

Hence, a resulting matrix C consists of NxN DCT coefficients. The top left coefficient corresponds to the lower frequencies while the bottom right corresponds to the higher frequencies (see Ffigure 2). Since the transformation uses cosine function, matrix C depends in horizontal, diagonal and vertical frequencies, leading to the idea that an image with few tones has a resulting matrix with a large value at the first element. DCT provides energy compaction because most information is compacted into the lower frequency coefficients; low frequency coefficients have larger magnitude whereas high frequency coefficients have smaller magnitude

Since the matrix T is orthonormal, the inverse of the DCT can simply be determined as:

$$M = T'CT.$$

2.3 Proposed Algorithm

The block diagram of the watermark encoder is shown in Figure 1.

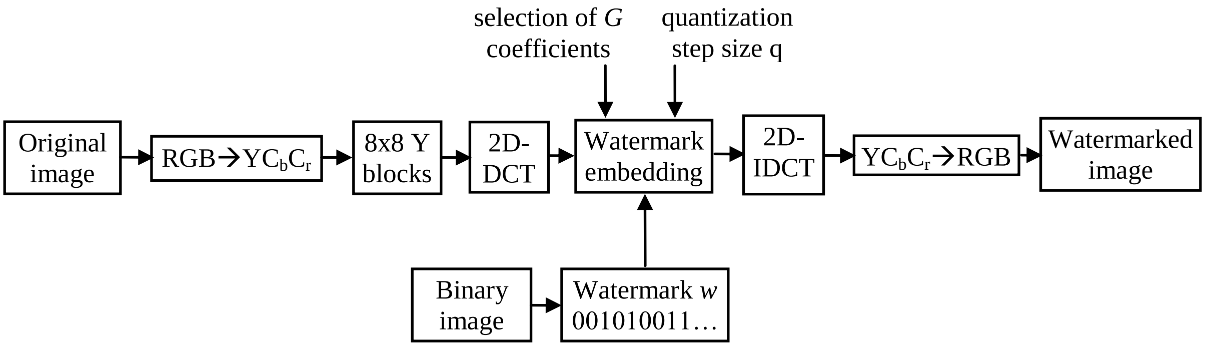


Figure 1. Block diagram of the watermark encoder

The steps of the watermarking process are described as follows:

1. The original image is converted from the RGB into the YC$_b$C$_r$ color space and just the luminance Y is further processed.
2. The Y-plain of the image is partitioned into blocks of 8x8 pixels (luminance values)
3. The 2D-DCT is applied to every luminance block, obtaining 8x8 DCT coefficient blocks.

4. From every DCT coefficient block, only the middle frequency coefficients are selected for watermark embedding. The selected coefficients are shown in grey color in Figure 2. By inserting the watermark in the low frequency coefficients, the visual quality of the original image would be significantly altered. If we insert the watermark in the high frequency coefficient, the visual quality of the image will not be altered, but the algorithm would not be robust to attacks.

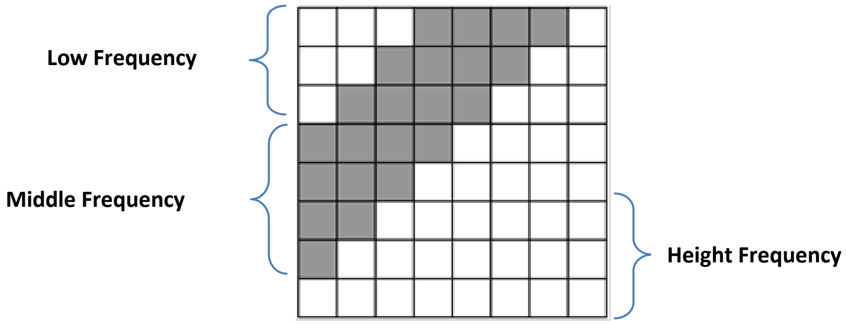


Figure 2. Frequency domains of an 8x8 DCT coefficient block. DCT coefficients selected for embedding are marked grey.

5. The watermark is a binary image of resolution $A \times B$ containing the copyright information. It is first transformed into a binary row vector *w* of size $P = A \times B$
6. One watermark bit of vector *w* is inserted in every selected DCT coefficient of a block.
7. A watermark bit *w*(*i*) is embedded into a selected DCT coefficient by rounding its value to an even or odd quantization level. Rounding to an even quantization level embeds a “0”, while rounding to an odd quantization level embeds a “1”, as shown in the next Equation

$$C^{w}_{m,n} = \left\lfloor \frac{C_{m,n}}{2q} \right\rfloor \cdot 2q + q \cdot w(i) \cdot sign\left(C_{m,n} - \left\lfloor \frac{C_{m,n}}{2q} \right\rfloor \cdot 2q \right),$$

where $C_{m,n}$ is the original DCT coefficient, $C^{w}_{m,n}$ is the watermarked coefficient, *q* is the quantization step and *sign()* is defined as:

$$sign(x) = \begin{cases} -1, & if\ x \le 0 \\ 1, & if\ x > 0 \end{cases}$$

8. After the entire watermark has been embedded, the 2D Inverse Discrete Cosine Transform (2D-DCT) is computed to obtain the watermarked Y component of the image.
9. The image is converted back from the YC_bC_r color space to the RGB colour space.
   To extract the watermarked information out of the image, we have to use the reverse process to the one we used to embed the watermark. The block diagram of the watermark encoder is shown in Figure 3.

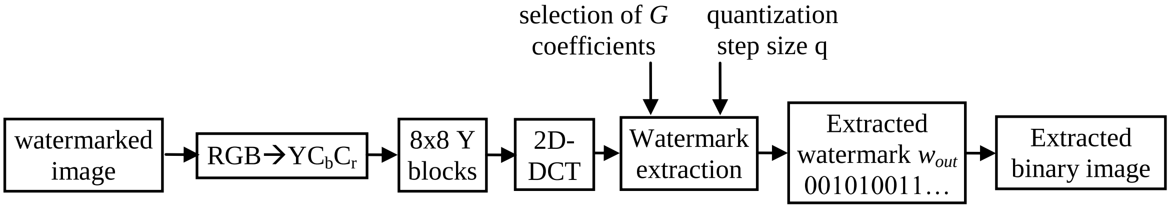


Figure 3. Block diagram of the watermark decoder.

The steps of the extraction algorithm are given in the following:

1. The watermarked image is converted from the RGB into the YC_bC_r colour space and just the luminance Y is further processed.
2. The Y-plain of the image is partitioned into blocks of 8x8 pixels (luminance values)

3. The 2D-DCT is applied to every luminance block, obtaining 8x8 DCT coefficient blocks.
4. From every block *i* of DCT coefficients we extract a sequence $s_j(i)$ of *G* bits, one bit from every DCT coefficient selected according to Figure 2 (we have chosen *G*=22).
5. We use the following decision function to determine the watermark bit extracted from block *i*:

$$w_{out}(i) = \begin{cases} 0, & if \sum_{j=1}^{G} s_j(i) \le \frac{G}{2} \\ 1, & if \sum_{j=1}^{G} s_j(i) > \frac{G}{2} \end{cases}$$

6. After extracting the watermark bits from every DCT block, we transform the resulting vector back to a binary image of resolution $A \times B$.

2.4 Experimental results

We performed the experiments on different colour images of resolution 512x512 pixels. In the following we are showing the experimental results for the “Lena” test image. Because we embed a watermark bit into a block of 8x8 DCT coefficients, the watermark payload has to be 512x512/64 = 4096 bits, so we have chosen a binary image of resolution 128x32 (4096 bits) as our watermark information, containing the copyright information (see Figure 4).

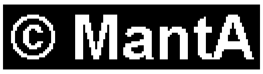
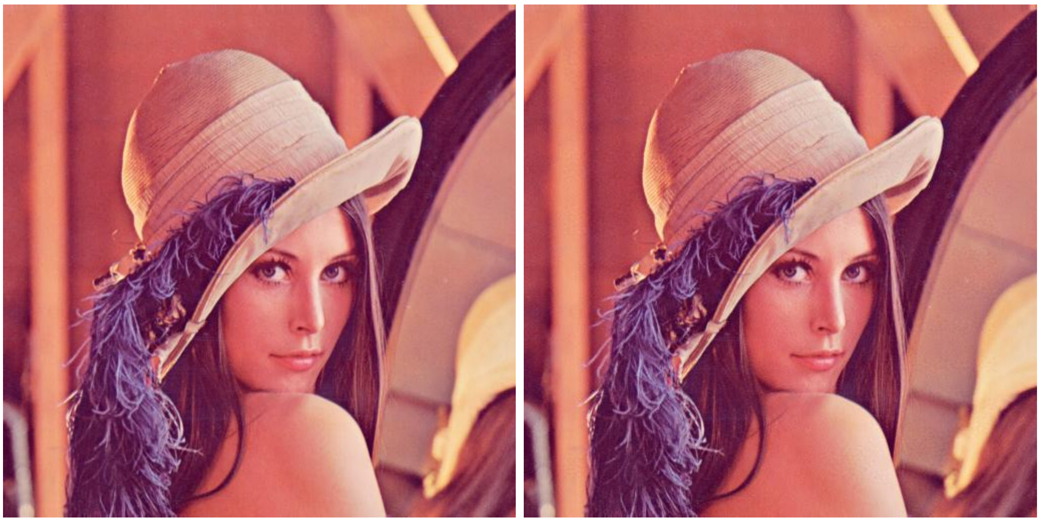


Figure 4. Binary image used as watermark

First we have determined the objective perceptual quality of the watermarked image using different quantization step sizes *q*. The watermarked images are shown in Figure 5.



a) b)

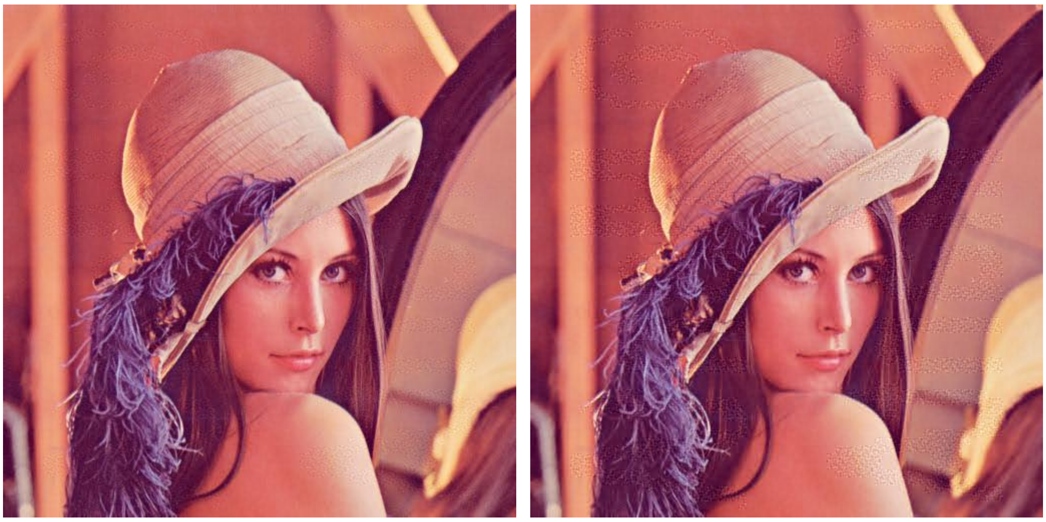


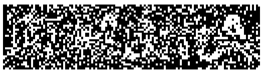
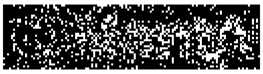
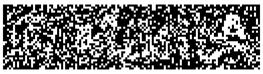
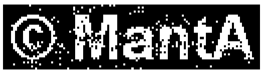
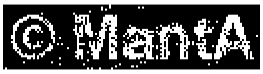
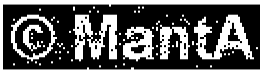
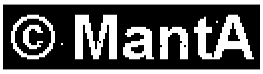
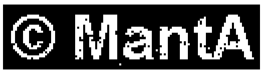
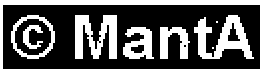
Figure 5. Original "Lena" test image and the watermarked images using a) *q*=4 and b) *q*=8 and c) *q*=16

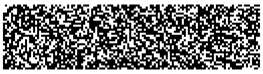
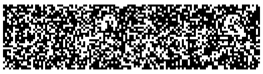
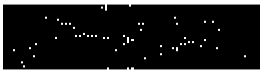
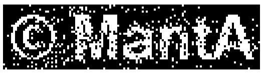
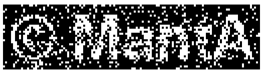
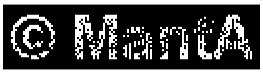
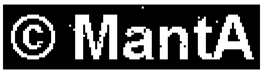
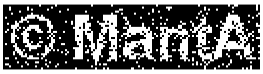
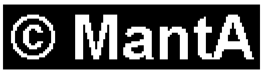
We have also tested the robustness of the algorithm to 5 different attacks:©

a) Blurring using blocks of 2x2 pixels
b) Brightening by adding Y_0=5 to the luminance of every pixel
c) Adding Gaussian noise of mean 0 and variance 0,01%
d) Adding "salt and pepper" noise with density d=0,5%
e) JPEG compression of the watermarked image with quality factor Q=80

The extracted watermark images are given in Table 1.

Table 1. Watermarks extracted after different attacks

Quantization step size q	**No attacks**	**Blurring**	**Brightening**
4			
8			
16			

Quantization step size q	**Gaussian noise**	**"Salt and pepper" noise**	**JPEG Q=80**
4			
8			
16			

3 Conclusions and Future Work

We developed and applied and technique of image watermarking which makes use of the DCT, obtaining some practical results. The developed algorithm was tested against attacks and the final application can be used in copyright protection and ownership verifications.

Watermarks have been embedded in DCT coefficients of middle frequency. Embedding in low frequency coefficients would make the watermark more resistant to attacks, but on the other hand it would degrade the image quality. Embedding the watermark in the high frequency coefficients would not degrade the image quality but would make the algorithm more susceptible to attacks. Hence an acceptable compromise to be made is inserting the watermark in the middle frequency coefficients. The choice of the quantization step size also influences the perceptual quality of the watermark images and the resilience to attacks. A higher Q improves the resilience to attacks bit produces higher visual artefacts in the watermarked image.

Our future work aims at:

- quality improvement of watermarked extraction, using small step size;
- security enhancement of the algorithm, by using a secret key for watermark insertion and extraction;
- designing a framework for additional watermarking techniques based on Discrete Fourier transform and Wavelets;
- integrating the results in a web based service oriented application with Java technologies.

References

[1] A. D. Gwenael and J. L. Dugelay, “A guide tour of video watermarking,” Signal Process. Image Commun, vol. 18, no. 4, pp. 263–282, Apr. 2003.

[2] Adrian Moise, *Algoritmi pentru prelucrarea imaginilor*, Vol. 1, Matrix Rom, București, 2011.

[3] Arockia Y., Raj and P. Alli, An Analysis and Overview of Modern Digital Watermarking, American Journal of Applied Sciences 9 (1): 66-70, Science Publications, 2012, ISSN 1546-9239.

[4] Chi-Man Pun, Ioi-Tun Lam, Fingerprint Watermark Embedding by Discrete Cosine Transform for Copyright Ownership Authentication, *International Journal of Communications*, Issue 1, Vol. 3, 2009.

[5] Ingemar J. Cox, et al., *Digital Watermarking and Steganography*, 2nd Edition, Elsevier, 2008.

[6] Ioan P. Mihu, *Procesarea numerică a semnalelor – Noțiuni esențiale,* Editura Alma Mater, Sibiu, 2005.

[7] John C. Russ, *The Image Processing Handbook*, Fifth Edition, CRC Press, 2007.

[8] M.A. Suhail, M.S. Obaidat, “Digital Watermarking-Based DCT and JPEG Model”, IEEE Trans. on Instrumentation & Measurement, Vol. 52, No. 5, Oct. 2003, pp. 1640-1647

[9] *Matlab Image Processing Toolbox User's Guide* - Version.5, The MathWorks, Inc., 2004.

[10] Nell Dale, Chip Weems, Mark Headington, *Introduction to Java and Software Design*, Jones and Bartlett Publishers, 2002, ISBN: 0-7637-2030-5

[11] Nikos Paragios, Yunmei Chen, Olivier Faugeras, *Handbook of mathematical models in computer vision*, Springer Publishing 2006, ISBN 978-0387-26371-7.

[12] R. O. Preda, N. Vizireanu, C. C Oprea, R. M. Udrea, “Highly scalable image watermarking in the wavelet domain”, European Conference on Visual Perception, ECVP 2008, Utrecht, Holland, pp. 122, august 2008.

[13] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, Second Edition, Addison-Wesley, 2001.

[14] Shiguo Lian, *Multimedia Content Encryption – Techniques and Applications*, CRC Press, 2009.

[15] Stelvio Cimato, Ching-Nung Yang, *Visual Cryptography and Secret Image Sharing*, CRC Press, 2012.

[16] Wilhelm Burger, Mark J. Burge, *Digital Image Processing - An Algorithmic Introduction using Java*, Springer, 2008.

Alin Manta
“Lucian Blaga” University of Sibiu
Department of Mathematics and Informatics
No. 5-7, Dr. Ion Ratiu st., 550012 Sibiu
ROMANIA
E-mail: mantalin@gmail.com

Arthur Mărginean
“Lucian Blaga” University of Sibiu
Department of Mathematics and Informatics
No. 5-7, Dr. Ion Ratiu st., 550012 Sibiu
ROMANIA
E-mail: name2@server2

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Robotic Arm Playing Tic-Tac-Toe Game

Kaloyan Mironov
Teacher Coordinator: Katalina Grigorova

Abstract

In the last few years of the game industry development very few games are directed towards games that require more thinking. In this paper we describe the development of a mind game – tic-tac-toe that can be played between a robot and a human. This is made in order to increase the interest in this kind of games. To create the system we use basic robotic hardware equipment, sensor information processing and developed artificial intelligence algorithms. The system has a commercial purpose. That’s why there is a suitable payment mechanism provided.

1 Introduction

Building and programming robotic arms is a main part of robotics. The term “robotic arm” originates from the Czech word “robota” and means “forced labor”. This describes a large part of common robots and their purpose.

The creation of Tic-Tac-Toe Robot was initiated by the need of development of a basic robotic system. This system was created using the modern standards main technologies from the field of: motion, sensing, real time information flows processing, decision making, etc.

The main goal of this project is to create a robotic arm that can perform a real time Tic-Tac-Toe game versus a human opponent. This kind of game was chosen as a typical example for developing artificial intelligence, and in the same time it doesn’t require expensive hardware components.

One of the main features we wanted to implement in this system is the real time feel of the game played with minimized waiting time. In order to achieve this we had to use information processing in real time and synchronization between sensors, algorithms and hardware responses.

The final goal of the project is to create a commercial system that can perform a paid Tic-Tac-Toe game. After a change of the main algorithm the system can perform more complex games like chess and be applied in the game industry.

2 Hardware

A picture of the system is shown on Fig. 1. It contains a robotic arm, a game pad and controllers that coordinate the motion of the robotic arm and the information flows from the sensors.

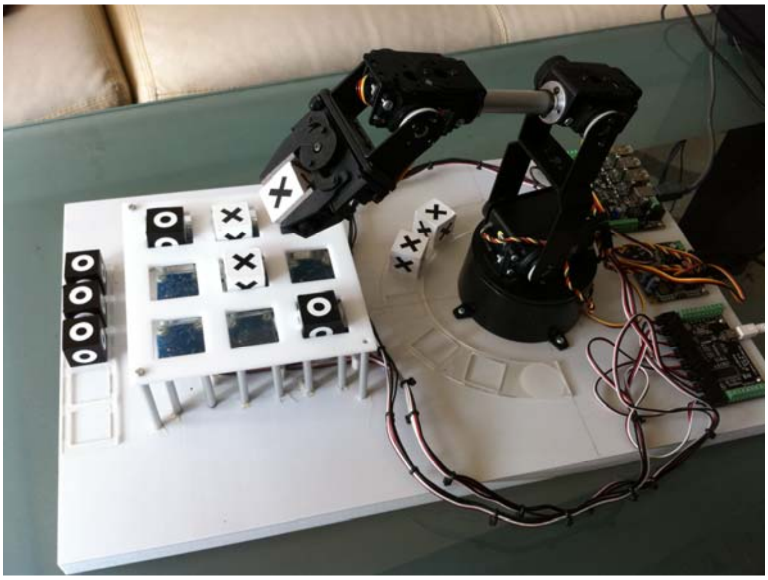


Fig. 1 – The robotic arm in action

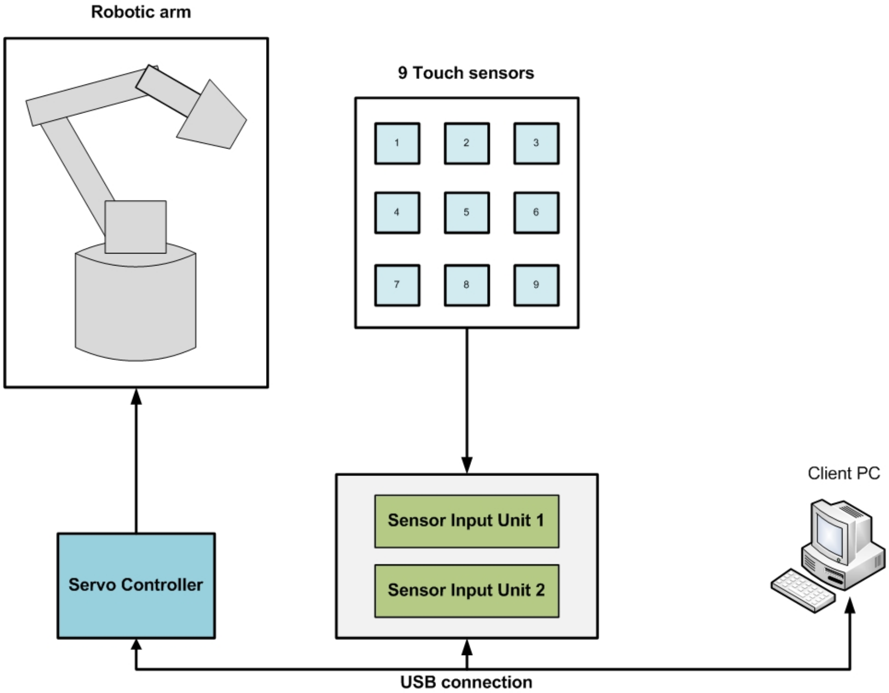


Fig. 2 – System hardware architecture

The hardware architecture of the system can be seen on Fig. 2. It consists of 4 main modules:

- Robotic arm – used to control and coordinate the motion during the game. The arm is used to show the action taken by the system;

- Servo controller – used to control the robotic arm. Depending on the game conditions the controller moves the arm in the desired direction;

- Game pad – consists of 9 (3x3) fields. It is presented by 9 touch sensors. The purpose of the sensors is to reflect the game logic. If a sensor is activated the information is sent towards the game algorithm;

- Sensor input unit – consists of 2 controllers that process the input information from the touch sensors. The main goal is to provide information for changes on the game pad.

3 Software

Several system requirements were made during the system software development. The access to the system has to be controlled and every user should have his unique identification. It is made in order to divide two groups of system users – administrators and players [2]. The six main modules developed for the system are presented on Fig. 3. They are enough to allow the system performance. Each of them has different functionality:

- RFID Security – as mentioned before, we need to identify the different kinds of system users. To achieve this we use RFID cards. This allows the game commercialization. The idea is that every player will have his own RFID card that will be charged with credits. The amount of games the player is allowed to take part in depends on the amount of his remaining credits. The card owner can also be an administrator. It gives him access to more system features. The administrator cards should be registered during the software installation process. There are 2 administrator places predicted. The system administrators can add, edit or remove players and to charge their cards with credits;

- System display module – generates the graphical user interface of the system. The module is directly connected with the RFID Security and the game algorithm. Depending on the user rights or the credits in the RFID card, this module visualizes the user menus. The purpose of the connection to the game algorithm is to present the robotic arm actions taken;

- Sensor Input Module – processes the sensor data flow and reports changes of their state. If such change of state occurs the module signals the A.I. Tic-Tac-Toe Algorithm where the game logic is controlled.

- A.I. Tic-Tac-Toe Algorithm Module – implements the tic-tac-toe game logic. This module communicates with the Sensor Input Module to get the sensor state information. Based on this information the algorithm makes a decision and sends it to the Robot Control Module, which performs the robotic arm action. The A.I. Tic-Tac-Toe Algorithm Module consists of two decision making mechanisms. First we use the tree algorithm Minimax [1]. It is a typical example of A.I. for a tic-tac-toe game. The other one is a library containing information about past games.

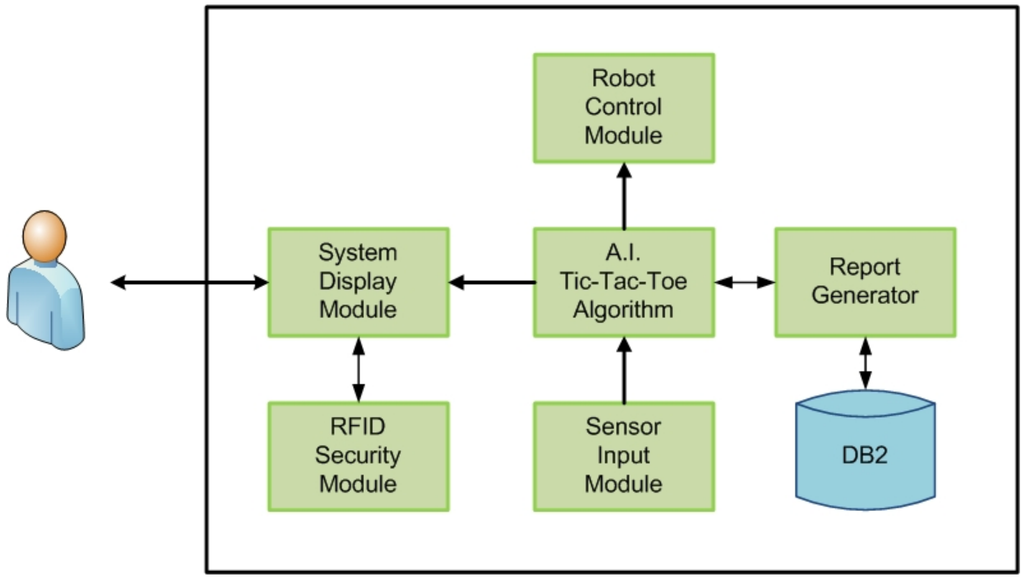


Fig. 3 – System software architecture

- Robot Control Module – controls the robotic arm movement during the tic-tac-toe game. Several functions defining different movements are described in this module. After a decision is made by the A.I. Tic-Tac-Toe Algorithm Module, the information about it is passed to the Robot Control Module in the form of movement algorithm. Each algorithm consists of exact string of kinematical coordinates which move the robotic arm. The coordinates are passed to the servo controller.

- Report Generator – stores and processes information about played games. The main goal is to save statistics about the games. The statistics are used for players' rankings, which encourages them to play more. Another important action performed by the module is storing the chronology about every game in the A.I. Tic-Tac-Toe Algorithm Module game library.

- Data Repository – a DB2 Express-C database using the option to store XML data. The information about every game is stored here by the Report Generator. Game chronology is saved in a tree form using XML files. The database also contains information about the users, rankings and RFID card credits.

4 System development

The development language was chosen considering several factors. The object-oriented approach used during the system design period made us use an object-oriented programming language. Another important factor is the possibility of software security of the used hardware. Considering these factors we chose Java as the system development language.

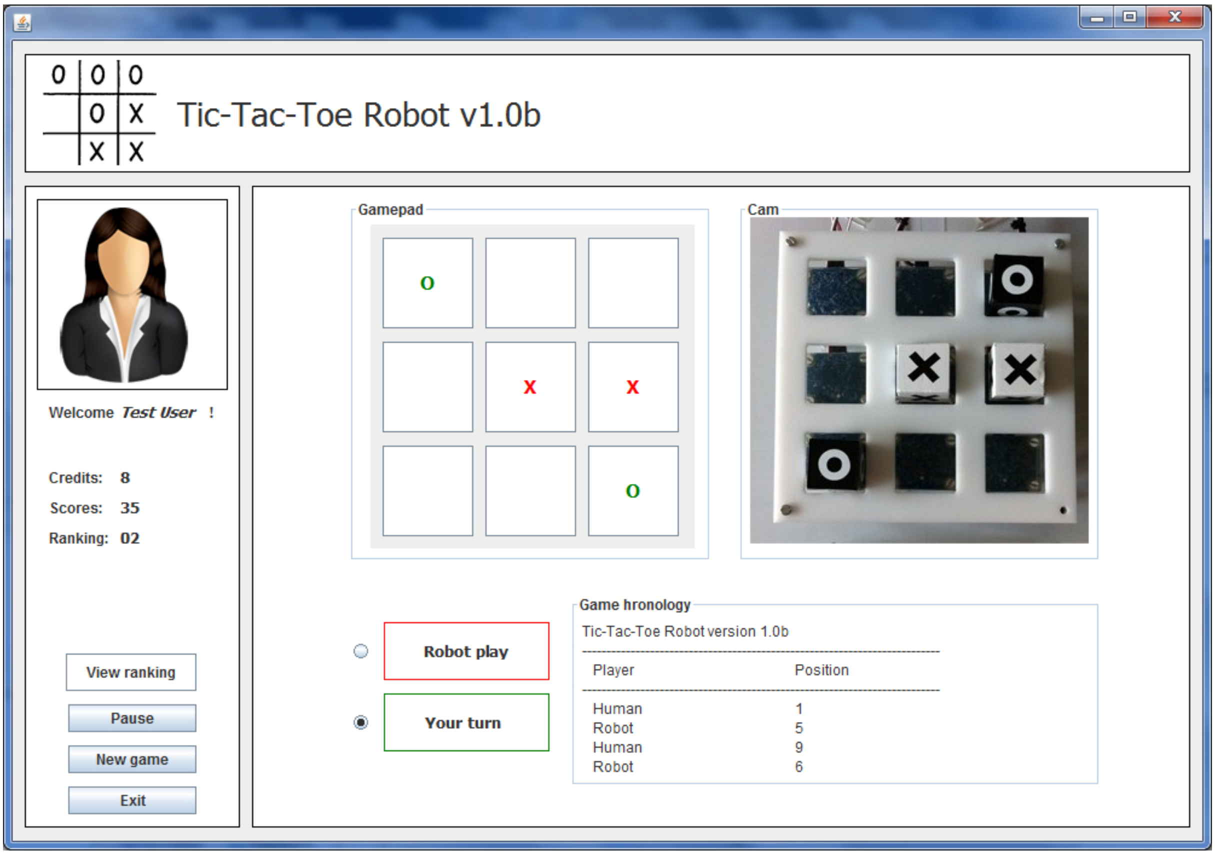
4.1 System functionality

Creating a commercial game demands a way of payment per game. Practice has taught us that using virtual currency like points, credits, etc. attracts more potential players. This is achieved by the use of RFID cards. The cards also serve as a way to divide the administrators from the players.

The original tic-tac-toe game rules say that the player should draw O or X on a chosen part of the 3x3 game field. To implement this logic we use 9 touch sensors that represent the game pad. Sensors give information about the performed actions by the player, while the system action data is taken from the game algorithm.

4.2 Graphical user interface

While creating the Display Module we aimed for a unified and easy to use interface (shown on Fig. 4). It has to be very user friendly and intuitive since most players will have low or average computer knowledge. [3]



Welcome *Test User* !

Credits: 8
Scores: 35
Ranking: 02

○ Robot play
● Your turn

```
Tic-Tac-Toe Robot version 1.0b
------------------------------------------------
Player               Position
------------------------------------------------
Human                1
Robot                5
Human                9
Robot                6
```

Fig. 5 – System graphical user interface

As you can see the data about the player (picture, name, amount of credits, amount of points and place in the ranking) is shown on the top left of the window. On the bottom left you can see the player menu including the View ranking option. The virtual game pad is shown in the middle of the screen showing the game logic. To the right the player can see a live video stream from a camera placed the real game pad. The user interface also indicates which turn should be played and the game chronology.

5 Conclusions

The developed system realizes a tic-tac-toe game between a robotic arm and a human player. It is made with commercial purpose for the game industry and has a payment mechanism implemented.

As a future plan we intend to change the tic-tac-toe algorithm with algorithms of other games such as chess, which will increase the interest in the system. In that case the sensing will be changed with video stream object recognition. An increased income will be possible if we make it possible to charge credits with some of the electronic payment methods.

The official launch of the system is planned for November 2012 during the Youth Chess Tournament. The winner will play a demonstrative game against the system. Until the mid 2013 the system is expected to become more popular and find application in some chess clubs in Bulgaria.

5 References

[1] Strong G., *The Minimax Algorithm*, February 10, 2011.
[2] Kapoor Ch., *A Reusable Operational Software Architecture for Advanced Robotics.*
[3] Bevan N., *International Standards for HCI and Usability.*

Kaloyan Mironov
Ruse University
Department of Informatics and
Information Technologies
8 Studentska Str., Ruse
BULGARIA
E-mail: mail@kaloian.net

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Solar Tracking System Using Computer Vision

Kaloyan Mironov, Vladimir Kotsev
Teacher Coordinator: Katalina Grigorova

Abstract

Solar energy is considered a renewable energy source. The sun radiates enormous amount of energy every day. Nowadays generating electricity by using renewable sources is one of the many things people can do to keep the world as we know it.

In this paper we present our solution for a solar tracking system that aims for better energy generation performance, controlled with the means of computer vision in combination with light detection resistors.

1 Introduction

Renewable energy is the energy that is generated mainly by using renewable natural resources. Such resources are sunlight, wind, rain, tides and geothermal heat. Around 16% of the global energy consumption comes from renewables – 10% come from traditional biomass and is mainly used for heating, while around 3% come from hydroelectricity. The last 3% come from the new renewables – small hydro, wind, solar, geothermal, biofuels and modern biomass. These last 3% are growing rapidly compared to the other 13%, meaning that this kind of energy gets more popular among people who feel the need to help the planet survive longer. The total share of renewable sources in global electricity generation is about 19% [1].

In time solar energy is becoming more and more powerful, thus making the usage of solar panels to create electricity more profitable. Investing in such technologies will bring huge long-term benefits. It can increase energy security, enhance sustainability, reduce nature pollution, lower the costs of mitigating climate changes and keep fuel prices from rising so rapidly.

This paper presents our solar tracking system that uses computer vision. The software helps with detecting and tracking the position of the sun. In order to achieve this we use live video stream processing algorithms. The main goal is for the solar cell to be facing the sun during the day. A small webcam installed in the middle of the cell is enough to capture the sky picture and detect the Sun’s trajectory. There are existing technologies helping solar cells with sun tracking. Except for the real time computer vision used, our system differs from that model with its compass sensor and GPS module. These help the solar cell or panel get the exact location and time of the Sunrise for every day of the year in every part of the globe.

2 Hardware architecture

In order to develop and test the solar tracking software we need to build the hardware model of what it is going to control. In this section we will explain the hardware architecture but let us have a brief look at the list of hardware components used to create the test solar panel (shown on Fig 1) [2]:

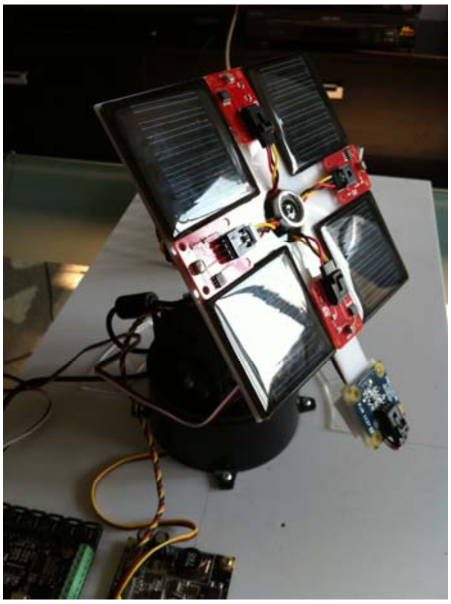


Fig. 1 – Solar Tracking System test solar panel

- Light detection resistor (LDR) – alarms whether the sunlight on the edge of the panel is more powerful than it is in the middle. Four of these are used to maintain the large amount of sun beams go to the solar panel centre;

- Web camera – with its resolution of 8 MP it delivers a high quality live video stream during the day. With its help we can track the Sun's trajectory and tune the panel position even more accurately;

- Compass – mainly used for the solar cell orientation before every sunrise;

- Light intensity sensor – placed on the solar cell this gadget helps us measure the light intensity, which is used for statistics database;

- Volt sensor and ampere sensor – used to measure the produced amount of electric power by the solar cell. Thanks to this data we can provide better reports about the quality of the system work;

- Sensor input unit – mainly used for processing sensor data and proceeding it to the system software;

- Servo motor – two of these are used for the solar cell positioning;

- Servo controller – controls the two servo motors to adjust the solar cell in order to face the sun during the day;

- RFID reader – serves as administrative login and helps creating user hierarchy with different rights.

The hardware architecture shown of Fig. 2 helps with the system structure and goals understanding. We developed the system using standard solar tracking methods and extended it with such that provide better results. The test solar panel consists of 4 connected photoelectric cells. We added the four light detection resistors (LDR) to provide basic solar tracking.

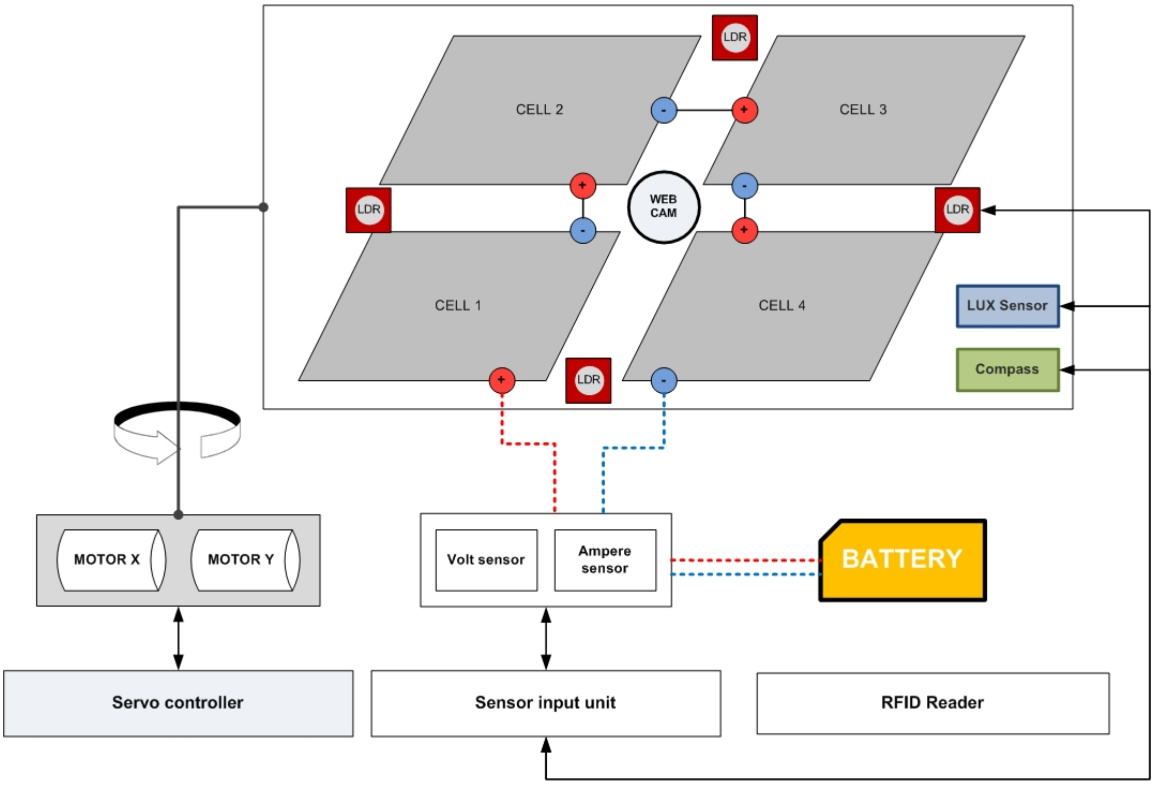


Fig. 2 – Hardware architecture of Solar Tracking System

In order to get the maximum results we also installed a webcam in the centre of the panel. With its help a more delicate panel positioning during runtime will be possible. The digital compass enables better east-to-west orientation of the installation and facing the best possible direction before sunrise.

Sensors are used in order to provide better monitoring of the system parameters. The used sensors measure the voltage and amperage of the produced electricity. We also use a lux sensor to measure the sunlight intensity. The sensor input data is processed by a system module and are visible at any time.

To ensure good mobility freedom we use 2 motors that enable both horizontal and vertical movement and initial positioning of the panel according to the position of the sun. The motors are controlled by a servo controller.

The RFID reader is a secondary device used for the administrative needs of the system. It is used as a login device for the people who want to tune the system settings.

3 Software architecture

The software architecture of the system is shown on Fig. 3 and consists of six main modules and a database. In this section their functions will be explained.

RFID Security – organizes the administrative access to the solar tracking system. This is done in order to disable unauthorized people tune the system. The access is based on RFID cards. Each card has its own unique chip that can be recognized by the reader. Having in mind that these whole installations can be big and very expensive, everyone who has access to the software should have a card of this type and each unique chip has its own system rights. In this way we can create a user hierarchy and lower the risk for the system. The RFID cards should be registered during the software installation process.

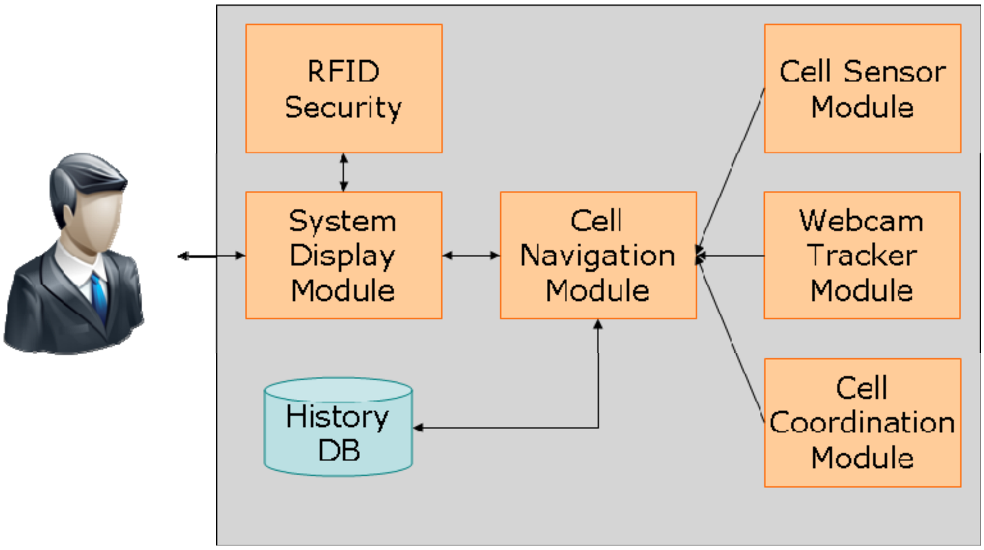


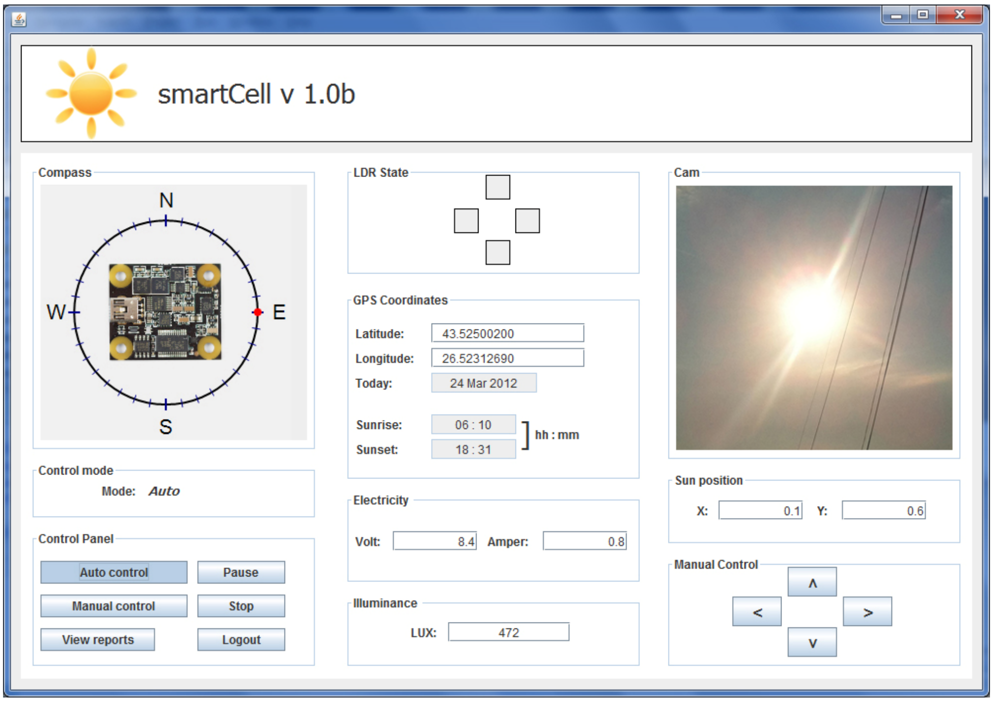
Fig. 3 – Software architecture of Solar Tracking System

System Display Module – provides the means of user interaction with the system. It visualizes the user interface needed for the system manipulations. It also displays the current sensor data. Using this module the administrator can set different options and apply changes to the system and also access data reports. The module receives real time data about the system sensors and can alarm the user, if something unexpected happens with the hardware. It is also directly connected with the RFID Security in order to enable or disable different actions for each user. The graphical user interface of the system is shown on Fig. 4.

Cell Navigation Module – controls the solar cell movement depending on the sun position in the sky. The system functions that navigate the servo controllers and set the positions of the servo motors are developed in this module. The information used for these functions is complex and is gathered from the three hardware input modules (Cell Sensor, Cell Coordination and Webcam Tracker). Another important functionality of this module is the report generation. At the end of each day all the cached data is structured in a report about the system work and stored in the database.

Cell Coordination Module – provides functionality for the initial positioning of the solar panel. In order to fulfill this task with better success it uses the help of compass data and GPS functionality. The GPS functionality provides information about sunrise and sunset exact time depending on the solar panel position on the planet. Compass data helps with east-to-west orientation and positioning. The combination of data from these sources provides better results for initial positioning than the one of ordinary solar trackers.

Webcam Tracker Module – helps providing data for the computer vision of the system. The module has functionality to process the digital video stream from the webcam. This helps us detect the exact position of the sun at any given time. In order to do this we use object recognition in real time. To recognize the sun on the sky picture we use several properties like shape, colour, size, etc. After the module recognizes the object needed, it finds its centre. In order to generate bigger amount of electric power we need the cell pointed at the centre of the sun object. The recognition is done per frame, thus making the cell positioning more precise.



smartCell v 1.0b

Control mode
Mode: *Auto*

GPS Coordinates
Latitude: 43.52500200
Longitude: 26.52312690
Today: 24 Mar 2012
Sunrise: 06 : 10
Sunset: 18 : 31
hh : mm

Electricity
Volt: 8.4 Amper: 0.8

Illuminance
LUX: 472

Sun position
X: 0.1 Y: 0.6

Fig. 4 – GUI of Solar Tracking System

Cell Sensor Module – processes the information flows from the hardware sensors [3]. The main tracked sensors are light detection resistors and the light intensity sensor. The LDR information is used for the basic control of the solar cell. In case of a webcam problem the full control of the cell is done by the LDR sensors. It is fulfilled by light detection on each of the four resistors. If one of them detect more powerful light than the others, a signal to move the cell centre towards in the given resistor direction is sent to the Cell Navigation Module. Light intensity sensor provides information which is needed for the statistics data about the system performance. Light intensity is measured in luxes.

History DB – stores data about the system performance for each day. The used database is IBM DB2 Express-C which enables storing XML data. The stored data can be divided into two groups – common solar cell information and data used for complex report generation. The first group consists of cell id, GPS coordinates and system administrators, while the second group stores information about date; sunrise and sunset exact time and coordinates (from the compass); produced electricity for the day; xml data about voltage, amperage and light intensity.

4 Conclusions

The developed project shows a new approach for solar cell control systems. Using computer vision in combination with the standard methods to track the sun position on the sky gives better results in performance compared to the ordinary systems.

The benefit of this project can be seen on Fig. 5. The testing is done on two solar cells with equal properties. The first solar cell uses the standard method based on LDR solar tracking, while the second uses the extended functionality with webcam solar tracking. When put on equal terms, the developed system achieved 8% better performance than the standard solar tracker. If all solar systems in the world use this kind of solar tracker, the extra electricity produced will be enough to power an average sized city like Sibiu.

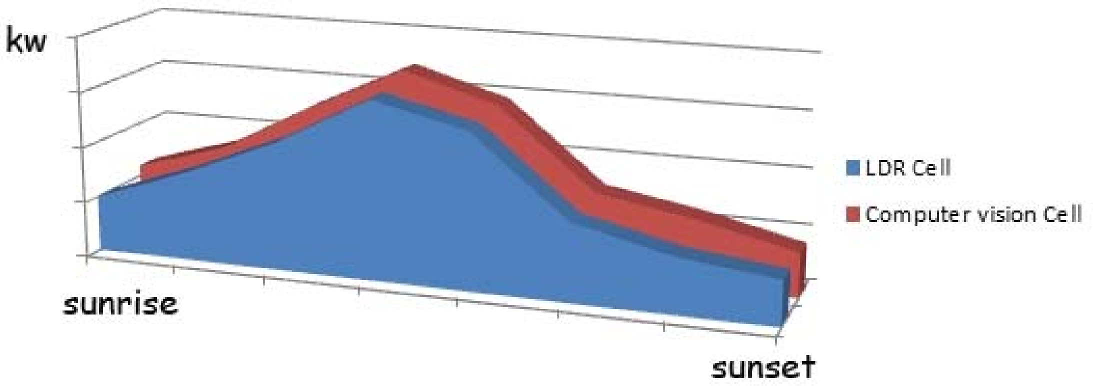


Fig. 5 – Results of testing LDR and Computer Vision solar trackers

Aiming to achieve better performance in electricity production we plan to develop an opportunity to track the sun position with a predefined trajectory. This trajectory can be calculated with a system algorithm or provided by a meteorological server. Another goal is to develop methods to track the sun on days with cloudy weather.

5 References

[1] Renewable Energy Policy Network for the 21st Century, *REN21*, Renewables 2011 Global Status Report, Paris, 2011.

[2] Paterson T., Rice J. and Valane J., *Solar Tracker*, ECE 476, Spring 2005.

[3] Wiegert P., McNeil D., Duncan M. and Levison H., *Fast Force Algorithms and Solar System Integration,* The Astronomical Journal, USA, 2004.

Kaloyan Mironov
Ruse University
Department of Informatics and
Information Technologies
8 Studentska Str., Ruse
BULGARIA
E-mail: mail@kaloian.net

Vladimir Kotsev
Ruse University
Department of Informatics and
Information Technologies
8 Studentska Str., Ruse
BULGARIA
E-mail: vladimir.kotsev@gmail.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Fast Conversion Clever Interpretation

Diana Alina Moşneag, Diana Maria Duca, Radu Marin Roț
Teacher Coordinator: Lect. PhD. Daniel Hunyadi

Abstract

For every businessman, his career is his world for at least a few hours a day. Even though our application doesn't save the world, it can save someone's world. Money has always been the thing that made the world spin. That's why our team came up with an idea that should be useful for each and every individual who is in need of exchanging money into a different currency and not just for the businessman type.

F.C.C.I. provides important information for the user such as: the exchange rate of the National Bank, the bank he should go to for the best deal at a specific time, the distance to the nearest bank and charts with the changes that occurred with the currency in the current month.

1 Introduction

Due to the fact that people have always been interested in money, we thought of an application that gives them important informations about the exchange rate at that very moment and which bank is the best option for them.

It's a fast, simple-to-use and precise currency converter, which uses the exchange rate of that day and from the bank you pick.

We live in a very busy world where things are always changing. That is why we chose to implement our application for mobile platforms. Instead of waiting for a computer to open a web page you can easily use your smartphone no matter where you are : at home, at work, even on vacation.

Fast Conversion Clever Interpretation is a really easy-to-use application with a friendly interface, appropriate for users of all ages.

2 Description

F.C.C.I. was written in C# and Silverlight for Windows Phone 7.
C# is an object-oriented programming language from Microsoft that aims to combine the computing power of C++ with the programming ease of Visual Basic. C# is based on C++ and contains features similar to those of Java, while Microsoft Silverlight is a free web-browser plug-in that enables interactive media experiences, rich business applications and immersive mobile apps.

Fast Conversion Clever Interpretation is very easy to use and intuitive for the user.

The application is optimized for both orientations, Portrait and Landscape.

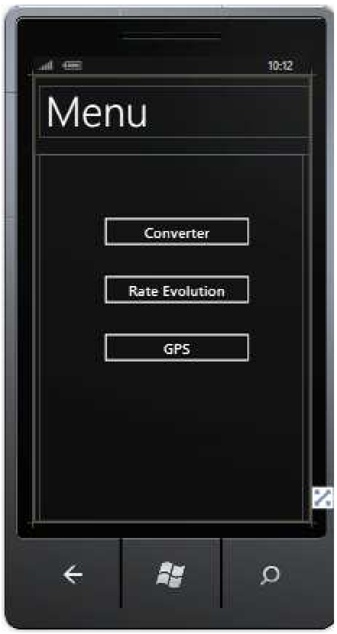


Fig. 1

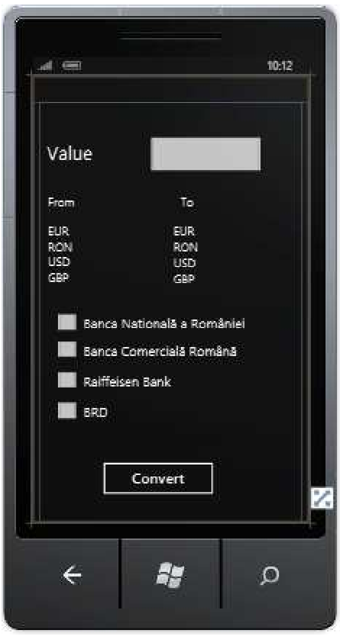
F.C.C.I. consists of three parts :

1. Conversion - the algorithm that converts from one currency to another one.

2. Rate evolution - the evolution of the currency for the last month.

3.Gps - finds the nearest banks for the current location of the user and displays the distance.

2.1 The conversion



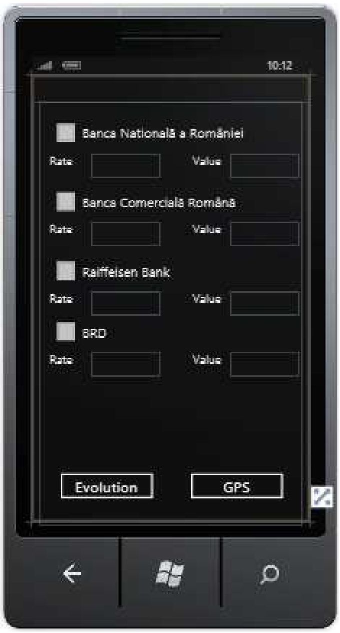


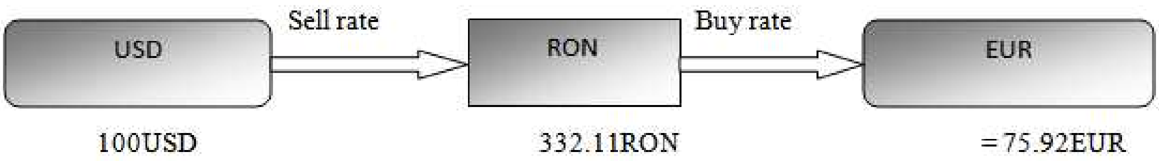
Fig. 2

Fig. 3

Here the user has to type the amount of money he wants to exchange, the currency he has and the one he is converting the money to. The application simply returns the amount of money that he would get if he would do the transaction at that moment, at the banks he checked.

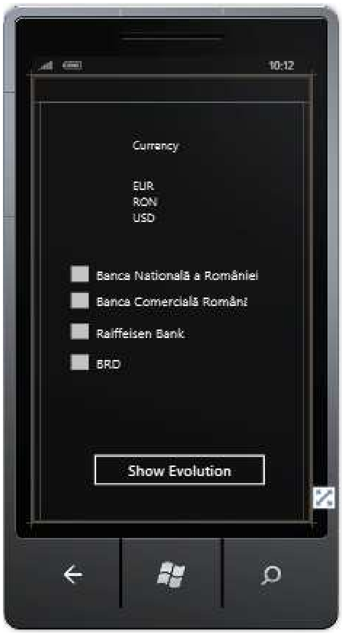
The idea behind the application is to get the source codes of the official websites of every checked bank using a webclient and to scrap the code to obtain just the necessary information that could help us build an accurate and up-to-date currency converter.

For example, if the user wants to exchange his money from RON to USD(or vice versa) the algorithm will do this directly, just consulting the rate of selling (buying) for the currency. If the user decides to use just foreign currencies (USD to EUR) the application will exchange first into RON using the selling rate and only after that (from RON) into the wanted currency.



2.2 Evolution of the rate

We tought that it would be quite helpful for the user, if the application would retrive the exchange rate for the last 30 days. Observing what happened with a currency in relation with the national currency, the user would know if the current rate is a low or a high one and decide if this is the moment he wants to make the exchange.



Currency

EUR
RON
USD

□ Banca Natională a României
□ Banca Comercială Român
□ Raiffeisen Bank
□ BRD

Fig. 4

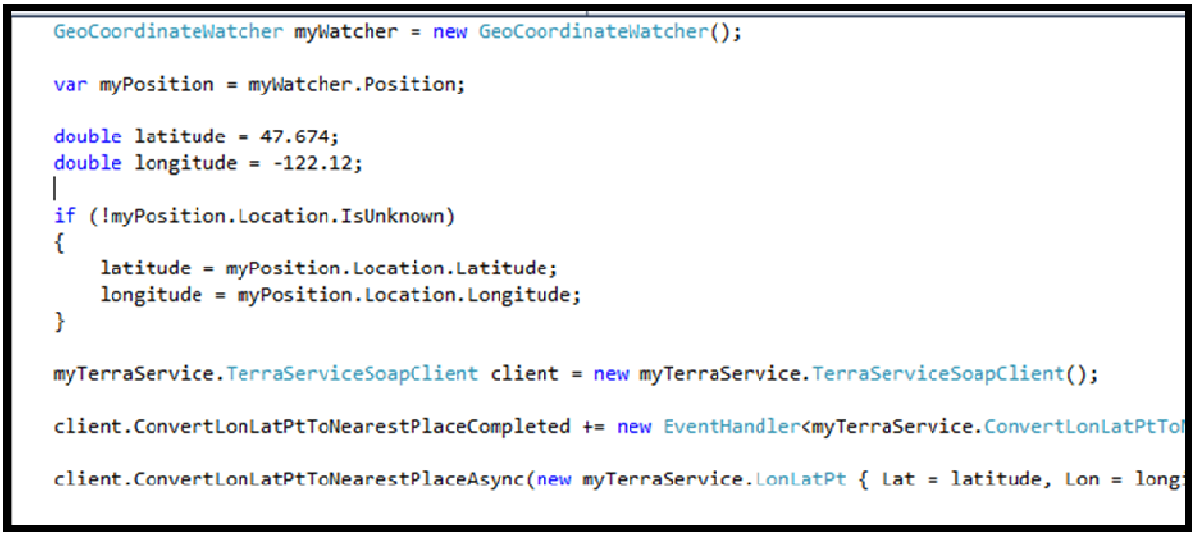
As we can observe in fig. 1, we can access the Rate Evolution page(fig. 4) from the Main Page of our application. Then the user only has to choose the currency and the banks and click the „Show Evolution" button.

An other way of accessing the evolution of the rate is right after the app has converted our money. On the results page (fig. 3) we can check the banks we are interested in, and click the „Evolution" button.

2.3 Distance

We also tought of the possibility that maybe the user is not necessarily interested in the bank with the best rate exchange, but into the one which is closer due to various factors so, we used a web service (TerraService) to retrieve the GPS coordinates of the phone (it is essential that the phone has Internet connection) and a database of GPS coordinates of the banks. The app retrives for the user a list of distances.

The buttons that take us to the page containing the distances are named intuitively „GPS" (fig. 1 and fig. 3) because our application uses GPS coordinates to calculate the distance.



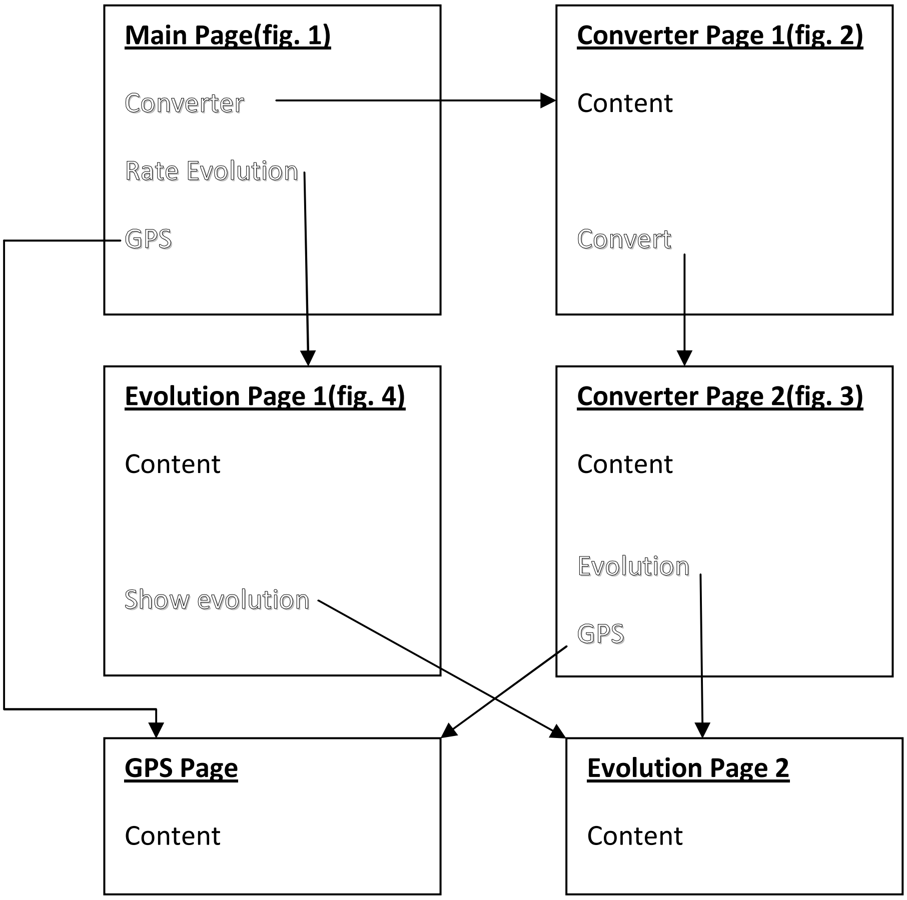
```
GeoCoordinateWatcher myWatcher = new GeoCoordinateWatcher();

var myPosition = myWatcher.Position;

double latitude = 47.674;
double longitude = -122.12;

if (!myPosition.Location.IsUnknown)
{
    latitude = myPosition.Location.Latitude;
    longitude = myPosition.Location.Longitude;
}

myTerraService.TerraServiceSoapClient client = new myTerraService.TerraServiceSoapClient();

client.ConvertLonLatPtToNearestPlaceCompleted += new EventHandler<myTerraService.ConvertLonLatPtTo

client.ConvertLonLatPtToNearestPlaceAsync(new myTerraService.LonLatPt { Lat = latitude, Lon = long
```

The latitude and longitude from above could be set to any values, in case of no connectivity or if the application is tested on a computer they will be the default values (latitude = 45.79 and longitude = 24.13 are for Sibiu, Romania).

2.4 The continuity of the application

A great aspect of F.C.C.I. is that you don't have to return to the Main Page every time you want to use a different part of the application. For example, after you do a conversion (Fig. 3), if you want to see the evolution of the chosen currency, you don't have to go back, click on the „Rate evolution" button and select againg the wanted currency and bank. You can just click on the „Evolution" button on the same page and you get the result. The same happens if you want to see the distances to the nearest banks, by clicking the „GPS" button.

The application map:



An example of code used for navigation between pages:

```
private void Button_GPS_Click(object sender, RoutedEventArgs e)
{
   NavigationService.Navigate(new Uri("/GPS_Page.xaml", UriKind.Relative));
}
```

3 Conclusion

Fast Conversion Clever Interpretation, as its name says, is a very useful mobile application because with its help, every single person could make the best choices for their money. It is different from the regular currency coverter apps especially because of the option to calculate the distance between the users position and the banks.

In the near future we want to integrate a navigation map to show the road between the users location and the bank he chooses. Also we want to make it available in other countries and we want to implement the menu in different languages.

Bibliography

[1] Charles Petzold, *Programming Windows Phone 7*, Microsoft Press, Redmond, Washington, 2010.
[2] MSDN Library, http://msdn.microsoft.com/en-us/library.
[3] Bob Tabor, *Windows Phone 7 Development for Absolute Beginners,* http://channel9.msdn.com/Series/Windows-Phone-7-Development-for-Absolute-Beginners.
[4] Windows Phone 7 Courses, http://www.pluralsight-training.net/microsoft/Courses
[5] Puja Pramudya, *Silverlight for Windows phone (e-book)*

Diana Alina Moşneag
"Lucian Blaga" University
Faculty of Sciences, Informatics
Sibiu, 5-7, Ioan Ratiu str.
Romania
mosneag.diana@yahoo.com

Diana Maria Duca
"Lucian Blaga" University
Faculty of Sciences, Informatics
Sibiu, 5-7, Ioan Ratiu str.
Romania
duca.diana92@yahoo.com

Radu Marin Ro□
"Lucian Blaga" University
Faculty of Sciences, Informatics
Sibiu, 5-7, Ioan Ratiu str.
Romania
radu_rot8@yahoo.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Optical data transfer. DATA COMPRESSION AND ENCODING USING COLOR

Vasile-Nicuşor Nechifor, Daniel-Sorin Radu
Teacher Coordinator: Prof. PhD. Dana Simian

Abstract

The article presents an original application for data transfer between computers and smart phones devices using a novel technique based on unsynchronized images. The coding formats available today like the barcodes have proved to be successful because of their various applications like coding of products by a manufacturer, coding of books by a publisher, etc. We assume that no direct connection between the devices can exist. A series of images, results of encoding process, are displayed on screen and recorded with camera equipped smart phones. With the images captured on phone device, they are decoded into the original file. Although the transfer rate is much smaller than it can be achieved with electromagnetic techniques (e.g., Bluetooth, Wi-Fi or USB), we envision to apply such a technique wherever no direct connection is available.

1 Introduction

The basic idea was to transfer data between 'devices' using 'video' image codes, i.e. by splitting the data over multiple image codes, which are then presented one after another as a video. Another device can then capture this video, split it into the separate codes, process them and recombine their contents to reproduce the original data.

Codes becomes more and more popular. They are used in advertisements, on business cards or e-tickets, or for referencing to web-pages as in Kaywa (http://qrcode.kaywa.com/). The amount of information that can be decoded robustly from a 2D barcode with ordinary mobile devices, such as mobile phones, is usu-ally restricted to several characters only. Thus, usually IDs, URLs or simple addresses are encoded. Yet, professional industrial scanners are able to decode a much larger amount of characters (several thousands) with an acceptable reliability. In this paper we present a new kind of transfer data using image code. It encodes data in four dimensions: width, height, color and time. Consequently, it cannot be printed on paper but is displayed on screens of mobile or spatial devices. Time-multiplexing colored 2D barcodes allows to transmit a larger amount of information robustly to other-shelf mobile phones without requiring an explicit synchronization (cf. figure 1(a) using QR codes)

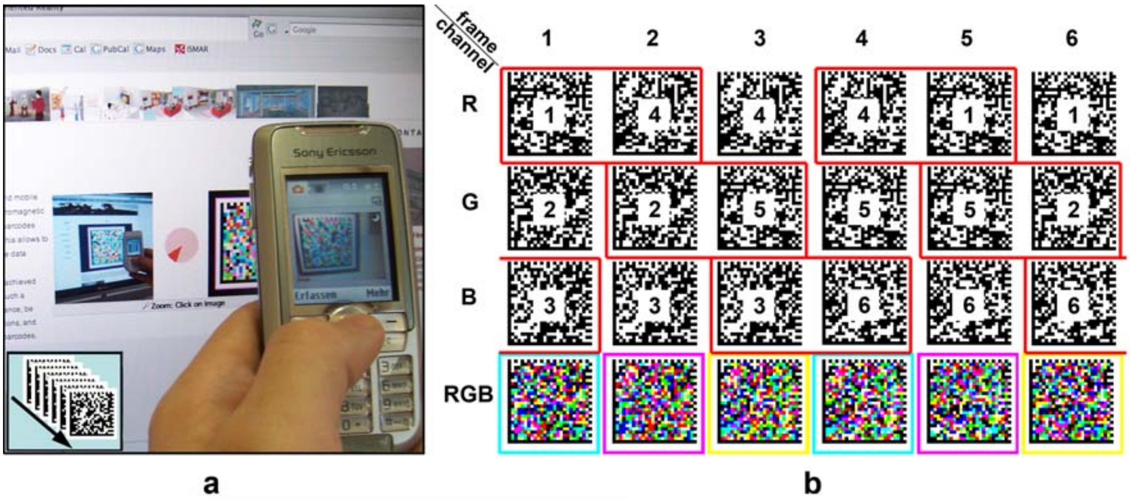


Fig. 1. 4D barcodes: (a) displaying and capturing, (b) encoding scheme for embedding 2D barcodes into a sequence of time-multiplexed 3D barcodes (barcode transitions are framed in red).

2 Related work

A large number of applications for mobile phones exist that read and decode printed QRcodes [1] as a standardized black-and-white 2D barcode. Datamatrix [2] is a similar example (yet not as common as QR-codes) and is used by applications like Kaywa (http://qrcode.kaywa.com). As mentioned earlier, only a small amount of information can be decoded robustly with consumer camera phones - limiting QR-code or Datamatrix barcodes to encode a few characters only. Han et al. [3] propose Colorcode, a 3D barcode which -in addition to a 2D matrix layout- uses colored bits as third dimension. But due to its small resolution of 5x5 cells it encodes only IDs that are resolved through a central lookup service (www.colorzip.co.jp). Besides applications in advertisement, the Colorcode is used for context aware systems [4]. 2D barcodes have also been applied to realize interaction techniques with mobile phones. Rohs [5] describes a barcode named Visual Code that stores up to 83 bits of data. Displaying it on a screen, it is used for tracking the movement and rotation of the phone relative to the screen [6, 7]. Another novel approach is presented by Scott et al. [8], who use Spotcode (a circular 2D barcode) for out-of-band device discovery and service selection - bypassing the standard Bluetooth in-band device discovery. This is applied by Madhavapeddy et al. [9] to also implement interaction techniques with Spotcodes that are displayed on a screen - using an online Bluetooth connection for data exchange. Similar techniques that apply displayed 2D barcodes for supporting mobile phone based interaction methods in combination with screens can be found in [10-12].

Besides barcodes, other possibilities for optical data transfer exist. Shen et al.[13], for example, explain how to read 7-segment digits from LCD/LED displays with camera equipped mobile phones. This system was mainly developed to support people with vision disorders by using their phones to recognize and read the digits on simple displays, such as on clocks. The system requires approximately two seconds for capturing and reading the digits on a Nokia 6630 - which is comparable to other OCR Software for mobile phones. Yet another interesting new approach for transmitting data optically is to use light sources instead of displays. In visible light communication, ordinary light sources are modulated with a digital signal. Approaches presented by Tanaka et al. [14] or Komine and Nakagawa [15] use white-light LEDs for illuminating a room and for transmitting time-multiplexed signals. The modulation frequency is high enough so that the transmitted signal remains invisible to the human eye. The embedded signal is received by photo diodes and is finally decoded. Using such a system, it is possible to transmit up to 100 Mbit/s and more.

3 Data compression and decoding using color

3.1 Color representation

Color is the result of interaction between a light source, an object and an observer. In case of reflected light, the light falling on an object will be reflected or absorbed depending on the surface characteristics such as reflectance and transmittance. For example, red paper will absorb most of the greenish and bluish part of the spectrum while reflecting the reddish part, making it appear red to the observer. Any color is the combination of three primary colors Red, Green and Blue in fixed quantities. A color is stored in a computer in form of three numbers representing the quantities of Red, Green and Blue respectively. This representation is called RGB representation which is used in computers to store images in BMP, JPEG and PDF formats. Here each pixel is represented as values for Red, Green and Blue.Thus any color can be uniquely represented in the three dimensional RGB cube as values of Red, Green and Blue.

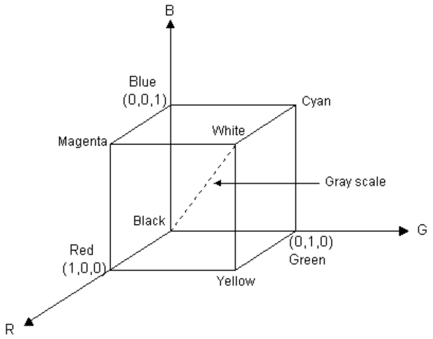


Fig. 1: The RGB Color System

The images represented in RGB color model consists of 3 constituent images, one for each primary color. The number of bits used to represent each pixel in the RGB space is called pixel depth (depth pixel). A n each constituent image (ie images that correspond to the three primary colors) of the RGB image is an 8 - bit image, then the RGB pixels of the image will have a depth equal to 24 bits.

3.2 Numeric representations of color

A color in the RGB color model is described by indicating how much of each of the red, green, and blue is included. The color is expressed as an RGB triplet (r,g,b), each component of which can vary from zero to a defined maximum value. If all the components are at zero the result is black; if all are at maximum, the result is the brightest representable white.

These ranges may be quantified in several different ways:

From 0 to 1, with any fractional value in between. This representation is used in theoretical analyses, and in systems that use floating-point representations.

Each color component value can also be written as a percentage, from 0% to 100%.

In computing, the component values are often stored as integer numbers in the range 0 to 255, the range that a single 8-bit byte can offer (by encoding 256 distinct values). These may be represented as either decimal or hexadecimal numbers.

High-end digital image equipment can deal with the integer range 0 to 65,535 for each primary color, by employing 16-bit words instead of 8-bit bytes.

For example, brightest saturated red is written in the different RGB notations as:

Notation	RGB triplet
Arithmetic	(1.0, 0.0, 0.0)
Percentage	(100%, 0%, 0%)
Digital 8-bit per channel	(255, 0, 0) or sometimes #FF0000 (hexadecimal)
Digital 16-bit per channel	(65535, 0, 0)

Table 1: Different RGB notations

In many environments, the component values within the ranges are not managed as linear (that is, the numbers are nonlinearly related to the intensities that they represent), as in digital cameras and TV broadcasting and receiving due to gamma correction, for example. Linear and nonlinear transformations are often dealt with via digital image processing. Representations with only 8 bits per component are considered sufficient if gamma encoding is used.

3.3 Data representation using colors

A bit is the fundamental unit of storage in a digital computer. There are only two numeric values that can be stored in a single bit: 0 and 1. A ’0’ bit represents the number zero, and a ’1’ bit represents the number one. To represent numbers greater than one, we need to use multiple bits in combination. For example, the number 1001001 in binary represents the number 73 in decimal (decimal is the number system that we use in everyday life, while binary is the number system used internally by computers). For a given number of bits, say eight, there is a smallest number that can be represented, and a largest number that can be represented. The smallest 8-bit binary number (in the encoding most often used for representing colors) is 00000000, which represents zero, while the largest is 11111111, which represents 255. Thus, given only eight bits, we can represent only 256 different numbers: 0, 1, 2, 3, 4, ... , 252, 253, 254, and 255. A 16-bit binary number, on the other hand, can represent the whole numbers between 0 and 65535, which is a vastly larger range than 0 to 255, even though sixteen bits is only twice as many bits as eight.

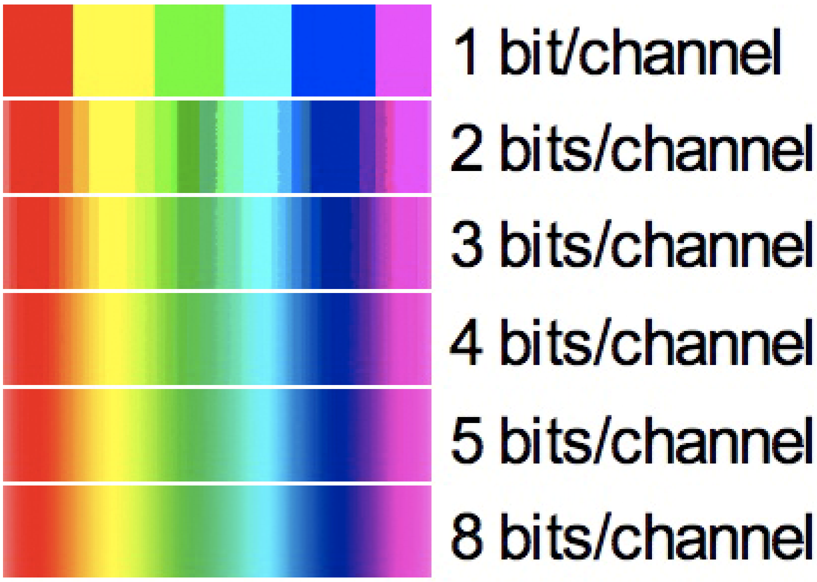


Fig. 2: Bits per channel. Each pixel is encoded via blending of three channels: red, green, and blue. Each additional bit available for encoding pixel colors doubles the number of hues that can be specified. Using more bits can result in smoother color gradients.

3.3.1 Color fading

Color fading can be a major drawback of this technology. Because of color fading, data may be represented wrongly. Color generally fades with time. Disadvantages due to fading can be minimized by selecting the resolution such that cell size is larger than maximum possible fading.

As the technology and precision of devices increases, resolution can be increased but the concept is same.

3.3.2 Color compression

In the above approach, we are not using all the colors efficiently. By using the remaining colors in the other 7 coordinates, we can use the whole cube very efficiently and even data compression can be achieved. The remaining seven sub cubes contain 256 * 256 * 1 56 * 7/8 = 14680064 colors. There are nearly 10000000 English words and templates used in MS word (Including all fonts and formats). We can make a database where each color (from these remaining seven sub cubes) represents a word. Then any word which has more than 3 letters (or characters) can be defined by a color which requires three 8 bit numbers to be represented. The words which are not in the dictionary (names, places etc) are not compressed and are represented by colors in first quadrant.

Thus by using a database, any word, no matter how many characters it has, can be represented by a color which requires only 24 bits to be represented. Thus data can be compressed to a large extent.

By using the above concepts of encoding and compression, large amounts can be compressed and transmitted in a more secured way. Even if the data is hacked by unauthorized person, he cannot decode it unless he has the same database an d knows the key.

Using the concept of compression, more amounts of data (word documents) can be printed in very less space (as images). Printing data is very easy because an ordinary printer can print all the 256 mode colors. It prints colors by mixing magenta, cyan and yellow in subtractive color mixing. Cyan, Magenta and Yellow are opposite to Re d, Green and Blue in the RGB cube. A printer uses CMY subtractive mixing because white paper is used for taking printouts and ink need ot be wasted to print white.

3.4 Time-Multiplexed Colored 2D Barcodes

One possibility to enlarge the data volume that can be embedded into a 2D barcode is to increase the code matrix resolution. However, the optics used for consumer cameras set clear limitations. Consequently, this is not an option when off-the-shelf mobile phones are used. The main idea of 4D barcodes is to split

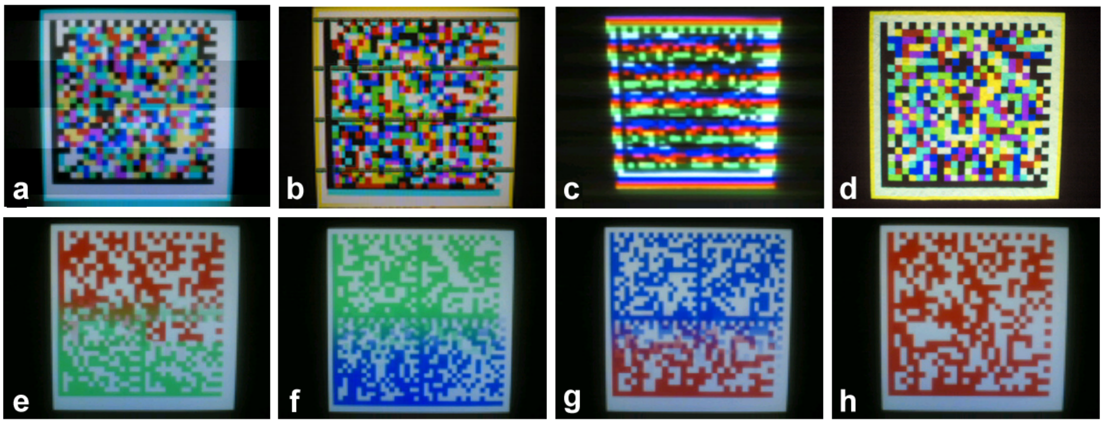


Fig. 3: Captured unsynchronized 3D barcodes from (a) 120Hz and (b) 60Hz CRT monitor, (c) DLP projector with white color wheel segment, and (d) LCD projector. Time-multiplexed R,G,B 2D barcode sequence captured form LCD monitor, (e-g) with and (h) without frame transitions.

the data into smaller chunks that are embedded into a series of decodable 3D barcodes. Thereby, the color dimension is used for increasing the robustness of the transmission. The animated 3D barcodes are displayed in an endless-loop on screens, and can be recorded by mobile camera phones. The looping duration and state is visually indicated on the display to give a feedback on how long the code sequence needs to be captured. After recording, individual barcodes are extracted, assembled and decoded on the phone to reconstruct the entire data content. Thereby, the challenge is the missing synchronization between displaying and recording. Our system is able to support LCD panels (or projectors) and Plasma screens. CRT monitors (or projectors) can only be applied if the decay rate of the utilized phosphor and the display's refresh rate ensure no full blank regions during the integration time of the camera chip.

We found that fast 120Hz CRT monitors (cf. figure 3a) are sufficient, while most slow (e.g. slower than 85Hz) CRT monitors (cf. figure 2b) are not. Due to an image generation via time-multiplexing (color and gray levels), DLP-based displays (i.e., projectors or back-projected screens) are not supported (cf. Figure 2c). We use Datamatrix barcodes in our prototype for encoding and decoding since decoders are freely available. Yet, it is extended to carry nested color bits. Animated GIFs are used to display the sequences of color codes. They can be easily embedded into web-pages. The following sections describe the encoding and decoding process in more detail.

3.5 Encoding

As mentioned above, the whole data set is split into smaller portions. They are encoded into a series of 2D Datamatrix barcodes having a size and resolution that can be decoded robustly by consumer phones. Binary data is preconverted into a sequence of 6-bit characters that is supported by Datamatrix. Therefore, we apply a similar technique as proposed by Josefsson [16]. After decoding, the reconstructed 6-bit character sequence is converted back to its original format.

Furthermore, the data can be compressed before encoding and is uncompressed after decoding to achieve a possibly higher throughput. The sequence of 2D barcodes are then converted into an animated GIF for presentation and recording.

Due to the missing synchronization between camera phone and display, however, such a simple approach would be very vulnerable to failures. The reason for this is that during the integration time of the camera, the screen content can change.

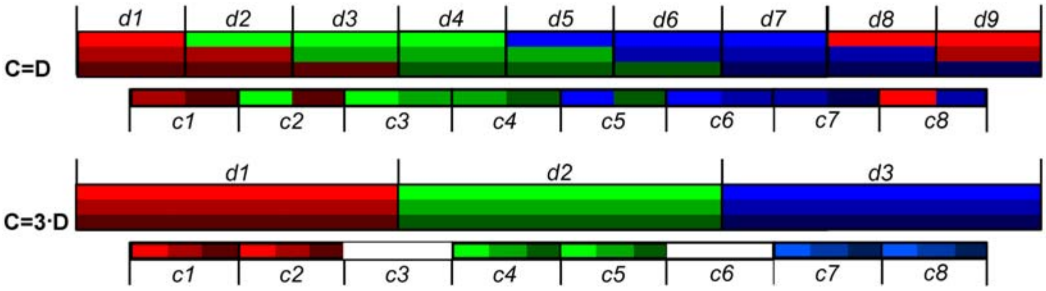


Fig. 4 Encoding: shifted (top) and non-shiftet (bottom) encoding scheme (c_i and d_i are captured and displayed frames respectively, the individual frame-embedded and captured/transmitted barcodes are color-coded).

This effect is illustrated in figures 3e-f for an LCD display. Here, full red, green, and blue 2D barcodes are displayed sequentially. Recording the sequence might show two different frame portions (and consequently two different barcode portions) in the same captured image. We solve this synchronization problem with a new encoding scheme. Instead of encoding one 2D barcode, we encode three different 2D barcodes simultaneously into each frame of the displayed sequence.

Each of them will be embedded into the red, green and blue color channels making it a 3D barcode. This, however, is not being done to triple the transfer throughput, but to increase the robustness of the system by adding redundancy.

Every 2D barcode of the original sequence is embedded exactly three times ones in each of three subsequent 3D barcodes, and it is always encoded into the same color channel. This is illustrated in figure 3(b). Only one 2D barcode is replaced between two subsequent frames. Combining the three color channels in each frame leads to the displayed colored 3D barcodes in lower row of figure 3(b). In addition, we surround each 3D barcode by a colored border. This is necessary for detecting if a captured frame was recorded while the barcode was replaced.

Therefore the border color is alternating between yellow, magenta and cyan colors that are complementary to the RGB code colors. Furthermore, the border color allows detecting which barcodes are encoded and which one will be replaced in the next frame. We use the complementary border color for indicating an upcoming barcode transition within a particular color channel. For example, if a barcode will be replaced in the next frame's blue channel, the border color for the current frame is chosen to be yellow.

Our encoding scheme (figure 4-top) applies equal capture (C) and display (D) rates and adds a two-fold redundancy. It shifts each barcode to three subsequent display frames and ensures that it can be captured completely in atleast two frames. The same result (i.e., redundancy and transmission rate) could be achieved with an unshifted encoding scheme and with $C = 3 * D$ (figure 4- bottom), for example. Shifting, however, increases the recognition probability during code resolution transitions and for non-constant capturing times (caused by online JPEG compression in our case). Note that both cases satisfy the Nyquist-Shannon theorem.

3.6 Decoding

After the sequence of 3D barcodes have been recorded on the mobile phone, each captured frame is analyzed for extracting the individual 2D barcodes and finally the encoded information. This task can be split into two preprocessing steps and one final decoding step, as illustrated in figure 5.

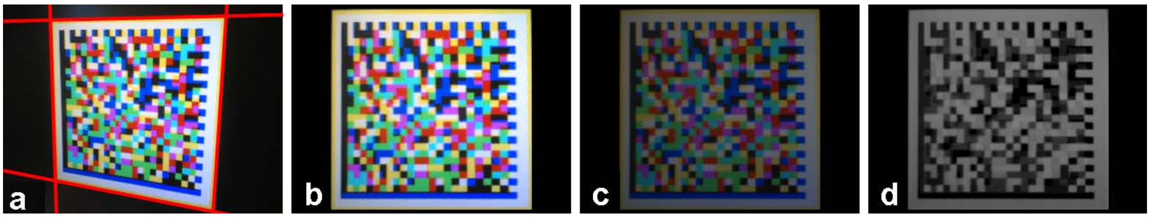


Fig. 5: Preprocessing steps for decoding: (a) captured frame and detected corners, (b) rectified image, (c) contrast and brightness adjusted image, (d) extracted 2D barcode (red channel) in gray scale.

Preprocessing: During the preprocessing steps, the 3D barcode in each captured frame is rectified to compensate for perspective distortions and is then contrast and brightness enhanced to compensate for noise. For rectification, the edges of the colored borders are detected through a conventional scan line algorithm. Having found multiple points on each edge, the corresponding line equations can be determined by solving a linear equation system. The intersections of the four edge lines lead to the corner points of the border that can be used to estimate a homography for rectification [17]. In our current prototype, the rectification works well for small barcode resolutions, but our Datamatrix decoder (we applied the Symbian Semacode library) fails often for rectified high resolution codes. However, this has not been critical in our case, since we have to limit the barcode resolution for mobile phone decoding anyway.

$$col_{new} = a \cdot col_{old} + b, a = 255/(255 - 2 \cdot \Delta), b = a \cdot (l - \Delta), \Delta = 127 \cdot c/100 \quad (1)$$

Following this step, the contrast and the brightness of the rectified images are adjusted using equation 1 to reduce image noise. Experimentally we found that a constant brightness reduction of l=20% and a constant increase in contrast of c=50% was optimal in combination with the (unknown) build-in white-balancing function of our mobile phones. This pushes the recognition rate up by a maximum of 20% (compared to no adjustments).

Handling Code Transitions: After optimizing the captured frames, the embedded 2D barcodes can be extracted and decoded. The first step is to detect whether or not a barcode transition happened within a frame. This can be detected by analyzing the border color (which has already been found during rectification, as explained above). If the colors of the upper and the lower border edges are the same, the barcode recorded in the frame is consistent. In this case, all three 2D barcodes that are encoded into the RGB color channels are completely captured. They can be separated, converted into gray scales, and decoded by the Datamatrix decoder. If the colors of the upper and lower edge are unequal, an inconsistency is detected. However, due to our encoding scheme it is possible to guarantee that always two barcodes are consistent (and completely recorded) in one frame. The reason for this is that only one of the three 2D barcodes is replaced between two subsequent 3D barcodes images. By analyzing the color of the upper border edge, we can determine in which color channel a 2D barcode is replaced in the following 3D barcode image (and is consequently recorded inconsistently in the current frame), and which ones are completely recorded. In correspondence to the coding example from section 3.5, a yellow upper border indicates a code transition in the blue color channel. Thus, two different barcodes are captured in the upper and in the lower portions of the current frame's blue channel, while the barcodes in the red and green channelsare consistent and complete, in this example. The same applies for the other two possible variations. The complete 2D barcodes can be decoded after converting them into gray scales (cf. figure 6). Note, that the intensity variations of the code bits are not critical for decoding.The inconsistent 2D barcode is discarded, but our encoding scheme guarantees that it will be complete in at least two of the three frames in which it was encoded (i.e., in the best case the same barcode is consistent in all three 3D barcodes; in the worst case it is only consistent in two 3D barcodes). After decoding the individual 2D barcodes, the encoded data packages from each one have to be rebuilt in the primal order of coding. Since it is possible that entire 2D barcodes cannot be decoded at all, and it is likely that recording the 3D barcode sequence does not start with the first frame (users might start recording at an arbitrary point during the looped image sequence), a correct order of reconstructed data packers is not given by the order of decoding.

To overcome this problem, we add a unique frame ID into a header section of each 2D barcode. This allows rebuilding the entire data set in the correct order. The data might have to be uncompressed and transformed back into the original representation, if necessary (in analogy to the encoding, as explained in section 3.5).

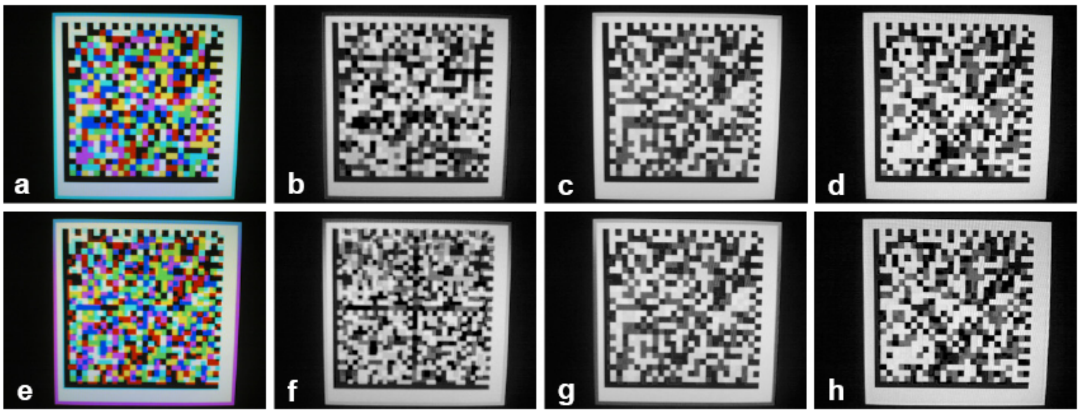


Fig. 6: Decoding: (a) captured 3D barcode with all three embedded 2D barcodes (bd) completely recovered, (e) example with inconsistent 2D barcode in red channel (f) while green and blue channels can be recovered (g+h).

4 Summary and Future Work

In this paper we presented the concept and an implementation of unsynchronized 4D barcodes. With our technique, we are currently able to transmit 1400 characters per minute (23 characters per second) with a success rate of 82% (95% for experienced users) from LCD, Plasma, and fast CRT displays to unsynchronized mobile phones. A user study has shown that such a technique would be accepted, if the decoding speed can be improved. Our technique has a much smaller transmission rate than established electromagnetic techniques, such as Bluetooth or WiFi. But it can be used in cases where such connections are are not established per se. Furthermore, it transmits significantly more data than corresponding 2D or 3D barcodes. Besides transmitting data from location- and device-independed public web-pages, we envision applications for recorded and broadcasted video content, for advertisement with billboard displays (as being done already with 2D barcodes) or in movie theaters, for information displays (e.g., transmitting updated schedules at airports or in trains), or for electronically displayed e-tickets (2D barcodes are already accepted to be displayed on mobile phones instead on printed paper). In future, the decoding time has to be decreased. The Kaywa-Reader (reader.kaywa.com) for example, offers a more robust decoding and a speed-up by a factor of two compared to Semacode. Porting our system to newer Android versions (e.g., Android 2,3,4) allows benefitting from improved camera control functions. This may increase the quality of the captured frames and might open the door to embedding six or more 2D barcodes instead of only three by using discrete intensity variations in addition (cf. figure 7). We have implemented and tested these 5D (height, width, color, time and intensity) barcodes, but deferred them due the too low image quality provided by the utilized mobile phones.

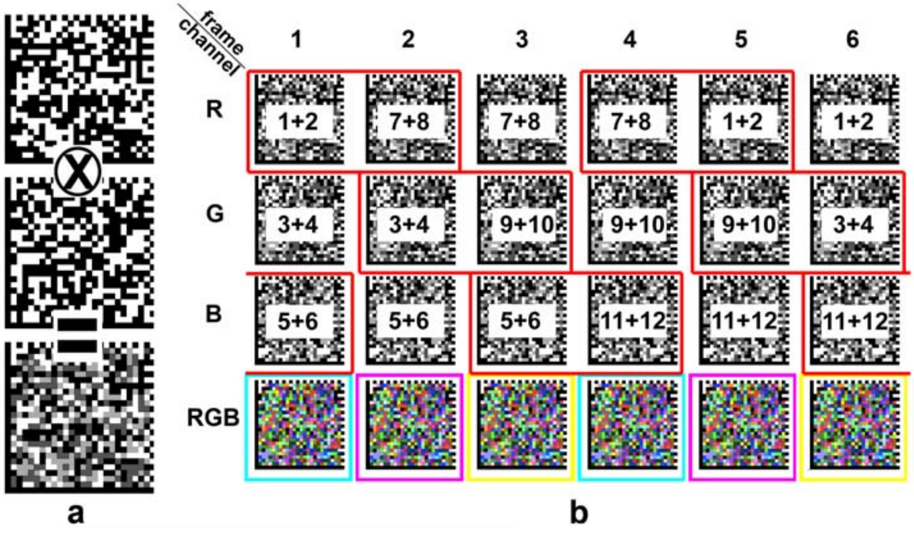


Fig. 7: 5D barcodes: (a) intensity coding of two 2D barcodes, (b) encoding scheme for embedding intensity coded 2D barcodes into a sequence of time-multiplexed 3D barcodes (barcode transitions are framed in red).

References

[1] ISO/IEC: *International Organization for Standardization*: QR Code.ISO/IEC 18004. (2000)

[2] ISO/IEC: *International Organization or Standardization*: DataMatrix. ISO/IEC 16022. (2000)

[3] Han, T.D., et. al.: *Machine readable code and method and device of encoding and decoding the same*, japan patent 3336311 (2002)

[4] Han, T.D., et. al.: Implementation *of personalized situation-aware service. In: Proceedings of the First Internaltional Workshop on Personalized Context Modeling and Management for UbiComp Applications (ubiPCMM 2005).* (2005)

[5] Rohs, M.: *Real-world interaction with camera-phones. In: 2nd International Symposium on Ubiquitous Computing Systems* (UCS 2004). Number 3598 in Lecture Notes in Computer Science (LNCS), Tokyo, Japan, Springer-Verlag (2004) 74-89

[6] Ballagas, R., Rohs, M., Sheridan, J.G.: *Sweep and point and shoot: phonecambased interactions for large public displays.* In: CHI '05: CHI '05 extended abstracts on Human factors in computing systems, New York, NY, USA, ACM Press (2005) 1200-1203

[7] Rohs, M.: *Visual code widgets for marker-based interaction*. In: IWSAWC'05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems - Workshops (ICDCS 2005 Workshops), Columbus, Ohio, USA (2005)

[8] Scott, D., S.R.M.A., Upton, E.: *Using visual tags to bypass bluetooth device discovery.* ACM Mobile Computer Communications Review 9 (2005) 41-53

[9] Madhavapeddy, A., S.D.S.R., Upton, E.: *Using camera-phones to enhance humancomputer interaction.* In: Adjuct Proc. of Ubicomp 04, Springer-Verlag (2004)

[10]Toye, E., Sharp, R., Madhavapeddy, A., Scott, D., Upton, E., Blackwell, A.: *Interacting with mobile services: an evaluation of camera-phones and visual tags.* Personal Ubiquitous Comput. 11 (2007) 97-106

[11]Vartiainen, P., Chande, S., Ramo, K.: *Mobile visual interaction: enhancing local communication and collaboration with visual interactions.* In: MUM '06: Proceedings of the 5th international conference on Mobile and ubiquitous multimedia, New York, NY, USA, ACM Press (2006) 4

[12]Ballagas, R., Borchers, J., Rohs, M., Sheridan, J.G.: The *smart phone: A ubiquitous input device.* IEEE Pervasive Computing 5 (2006) 70

[13]Shen, H., Coughlan, J.: *Reading lcd/led displays with a camera cell phone*. In: CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern RecognitionWorkshop,Washington, DC, USA, IEEE Computer Society (2006) 119

[14]Tanaka, Y., et. al.: *Indoor visible light transmission system utilizing white led lights.* In: IEICE Transactions on Communications, vol.E86-B, no.8. (2003) 24402454

[15]Komine, T., Nakagawa, M.: *Performance evaluation of visible-light wireless communication system using white led lightings.* In: ISCC '04: Proceedings of the Ninth International Symposium on Computers and Communications 2004 Volume 2 (ISCC"04), Washington, DC, USA, IEEE Computer Society (2004) 258-263

[16]Josefsson, S.: *The Base16, Base32, and Base64 Data Encodings*, RFC 3548, United States. (2003)

[17]Heckbert, P.S.: *Fundamentals of texture mapping and image warping. Technical report*, University of California at Berkeley, Berkeley, CA, USA (1989)

[18]Deutsch, P.: DEFLATE *Compressed Data Format Specification version 1.3*, RFC1951, United States. (1996)

[19]Tobias Langlotz, Oliver Bimber: *Unsynchronized 4D Barcodes, Coding and Decoding Time-Multiplexed 2D Colorcodes*, Bauhaus-University Weimar

Vasile Nechifor Nicușor
"Lucian Blaga" University of Sibiu
Department of Mathematics and Informatics
Dr.I.Rațiu Street No.5-7, Sibiu,
550012,ROMANIA
Vasile.nechifor@ulbsibiu.ro

Daniel Radu Sorin
"Lucian Blaga" University of Sibiu
Department of Mathematics and Informatics
Dr.I.Rațiu Street No.5-7, Sibiu,
550012,ROMANIA
radu_s_o_r_i_n@yahoo.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Droid monitor

- Android monitoring software -

Dragoş Iulian Obancea

Abstract

In this paper, I focused my interest on an Android application, which aims at monitoring the position of a device, the received and delivered calls and messages, and other activities that can be used by means of such operating system. When using this application user has the possibility of sending special messages (Check-In), or performing authentications/registrations. The events are transmitted to a server, which makes it possible to login and to save them in a database. The amount of data that has been stocked may be accessed on www.droidmonitor.com in different formats. The collection of applications (Android, server, and web-site) represents a product on its completing merge, which can be purchased on www.market.android.com or on the previous mentioned link.

1 Introduction

Since ancient time, the evolution of the society was grounded on the thick foundation of information. Far away in the past the quantity of information was rather poor, and the ways of transferring it, was even more inadequate, which lead to a slow, developing process. About a century ago, however, the discovery of new more rapid ways of manipulating and transmitting the information, has brought to the development of technology, that has accelerated this process and so a new cycle was created between information and development as such [1]. All in all, we can make the following statement: information is a determining factor in the evolution of mankind; moreover the quantity and speed of transfer, determines the level of evolution of the technologies, and also the standards of living. All these lead to the concluding idea: information is the essential factor that determines an exponential growth of the level of society.

In the previous paragraph, it has been insisted upon the importance of the impact that information has upon society. The next step is to classify the amount of information into categories and certain steps, in order to point out the place, that the product being described, occupies. So, we can order information on a historical scale:

- Basic knowledge about plants and animals, that have been useful for the survival of mankind;
- Basic knowledge about the elements that are to be found in nature, and that help in building up a place to live in and survival tools;
- Advanced knowledge about plants and human body have brought ways of curing diseases;
- Knowledge about the human way of thinking has improved war strategies that would bring victories;
- Different knowledge that has led to the explosion of technological development from the last century. One of these technologies is that of the maps and GPS locations, that further on forms the basic ground of other more complex and sophisticated technologies, et cetera;

This paper is focused especially on the monitoring process of two kinds of information:

- GPS location
- Conversations between two or more persons

If we are to use our imagination, the process in question would resemble to visualizing a war, where one of the opponents foresees all the possible moves of the other. In this case, it is easy to foresee the winning party. Some more realistic examples are:

- The user always knows where the personal mobile device is;
- As a parent, one can always check where the child actually is, or with whom is he/she talking with;
- The employer always knows where the employee is;
- The leaders of the army will always keep an eye to where the solders are located;

The advantages of such a tracking system are more than obvious, some of which would be: avoiding stealing and kidnapping, avoiding disasters or injustice by means of anticipating them with the help of recorded phone calls et cetera; Such methods are already in use, others are due to evolve, and some have remained at the stage of being a concept.

The amount of information previously talked about is highly sensitive data and needs special care, because they can become risky. The means of manipulating this type of information will be discussed in the following chapters.

2 Software Requirements Specification

The chapter on the software requirements specification is built on the concept of obtaining and manipulating data. The first task on solving the problem would be: **to create a product that has the capacity of monitoring the mobile devices that use an Android operating system.**

After having analyzed the given task, one is lead to think, that the first step consists in actually producing software that would fulfill the requirements. The result of the monitoring process constitutes in a set of information that is to be stocked. In consequence a server is needed, in order to receive data packages and store them. Moving a step forward, one can notice that without the existence of a module for visualization and manipulation of the data, the entire system is of no use. This third system has to work independently from the first and the second one.

Now, if we are to update the above mentioned quest, it will be modified as follows: **You have to create a product that is formed out of three independent systems, and that fulfill certain functions:**

- The first system is to be dedicated to mobile Android devices. Its task will be to collect data permanently, regarding the events occurred in the devices, and it will periodically verify their location. The data is to be moved in packages that will be sent further on to a server.

- The second system consists in a server placed on an independent device, which completes the charge of receiving the data packages via the internet. The data packages will be analyzed, and then stored.
- The third system has to allow users to visualize and manipulate data concerning the personal mobile devices. In order to have an easy and fast access to the information, the system will be a web application. Besides having a „friendly" data display, the software has to offer the possibility of generating reports, statistics, E-mail services et cetera.

If we are to examine the present market in this field, we will encounter the existence of a relatively small number of products that describe similar functionalities. Acquiring some of these products consists in actually buying them, others do not work properly, and the percentage of those that fulfill the tasks mentioned above is rather small. In this order of ideas we are lead to the third, and final, objective: **to analyze and brush the functionality offered by similar products and implement new functionalities, whose concepts will be built on the way of developing the systems.**

The concept described in the paper, is often referred to, as a *product*, because it represents more than a simple project in the field of informatics, but rather a system that is meant to be perceived as any other product on the market. So, the Android application will be available on *Android Market* while the web application will be at hand on www.droidmonitor.com.

Apart from the objectives that have been covered and that are to be dealt with, the application has to offer a series of qualities, out of which we remind the more significant ones: correctness, fiability, extensibility, reusability, performance, easiness in usage, universality and accessibility.

2.1 Correctness

Correctness is a property of maximum priority of software's [2]. One of the main necessities of the common user is to be furnished with 100% correct information. Although achieving this goal is also influenced by the hardware of the selected devices, it is demanded that the acknowledged pieces of information to be as concise and complete as possible.

2.2 Fiability

It is obvious that fiability completes itself with correctness [2]. No user would wish that the Android application he has been using to stop working unexpectedly or the site meant for checking the data to fail down. Again, such inconvenience might be caused by the hardware, but one has to analyze all the problems that can interfere. This process can be correctly accomplished, if a detailed testing stage is introduced, which implies the initial publishing of a beta application.

2.3 Extensibility

Extensibility is the ability of a system to easily adapt to changes in the specification phase [2]. The process of development of the product is to be attained within the frames of this concept. To be more precisely, any system out of the three might be replaced. Moreover, the two applications that interact directly with the final user can be added different applications dedicated to other types of devices. For instance, an application of the same kind can be built for IOS, and can easily be integrated in the product.

2.4 Reusability

Reusability is the ability of system components to be able to be used at developing more different applications [2]. In this matter, it is required that the systems are independent, and more than that, each system will be built on layers, that can be, at any time, replaced or integrated with other components.

2.5 Performance

The quantity of data that is to be manipulated by the software is really large, and it also shows tendencies of exponential growth. Out of such reason, it is highly important that the transfer, access and data manipulation to be performed by means of the most rapid processes that are acknowledged. Again, the efficiency of the process it depends on the performance of the hardware, which sometimes might be limited out of financial reasons.

2.6 Easy to use

There are quite a large number of users that can use this application. On such grounds it is necessary that the Android application to be used with little effort, and with not much intervention from the user's side. Actually, after having the user's application installed, the system might as well be left to run in the background. As far as the web application is concerned, the access has to be as rapid as possible. All the functionalities that are set at disposal will be easy to understand and handle. It is preferable to introduce more languages in the application for an increased comfort of the user.

2.7 Universality

The product will be available to any Android user, regardless of the country of provenience or the type of the device.

2.8 Accessibility

Accessibility is an objective that has been already handled, because the user has access to the product directly on the internet by using *Android Market* or www.droidmonitor.com.

3 Main results

It is a shared opinion that a paper has to be schematic presented, with short and concise ideas, rather that long and tiresome ones, in order to determine a higher receptivity of the listener/reader. In the chapters that follow, the paper will be structured in several sections of middle and small length:

- Describing the product;
- Describing the process of evolution of the three systems;
- Describing Android application;
- Describing the server;
- Describing the web application.

3.1 Describing the product

Droid Monitor is an Android application that permits monitoring mobile devices by means of:

- Tracking GPS locations at time intervals chosen by the user, or simply whenever changes occur.
- Tracking the incoming and outcoming calls and messages.
- Transmitting special messages by the user (Check-In).

The collected data are sent to a server that stocks them in a database. In order to visualize the information in different formats, one can access the website, described in the following chapters.

The product is dedicated to any Android users. He/she is able to download the application from Android Market or from www.droidmonitor.com. Installing and using the system requires basic knowledge in the field of informatics, such as: the capacity of installing an Android application and the capacity of navigating on an internet browser.

3.2 Describing the process of evolution of three systems

In the process of developing the systems attaining the selected objectives was the main goal, despite the type o the methods that have been practiced. Nevertheless, the initial problem is in fact the result of another one, that was raised by the author and that implies learning and gaining knowledge in a larger scale of fields. Consequently, the following graphic shows the use of various technologies for each system:

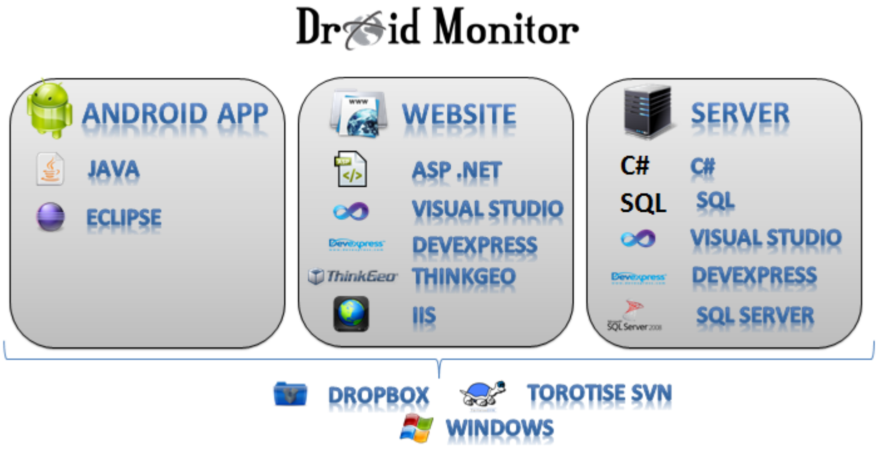


Figure 1. Used technologies for each system

3.3 Describing Android application

3.3.1 Generality

Android application was built as a service that runs permanently, without requiring a repeated intervention of the user. Although Android system does not permit the applications to run uninterrupted, it places at disposal a series of services, which allow the application to be informed whenever an event occurs in the mobile device. In this manner, an independent system was achieved, which succeeds to attain all the objectives discussed above. The main aspects in regard to the mode of functioning of the application will be presented in the chapters that follow.

3.3.2 Accessibility

The application will be released in the shortest time on Android Market and on www.droidmonitor.com. Initially, it will be purchasable in a beta format, and then, depending on the feedback received from the users, it will be provided a 1.0 version. The beta version could be downloaded for free, and the final one will have a reference price between 5$ - 10$.

3.3.3 Monitoring and data transfer

Monitoring the events is undertaken by means of receivers. These inform the application every time an event appears for which an existing code has been implemented. At the moment when such an event occurs, one should appeal the respective code section, where the data is collected. These activities take place in a separate thread from the main activity of the application. After the data has been received, they are sent via another thread in packages towards the server.

It will not be necessary to leave the GPS device open, in order to prolong the durability of the battery. Precise locations can also be found by using the network connection. When both devices are open, the application will select the more efficient way of obtaining data.

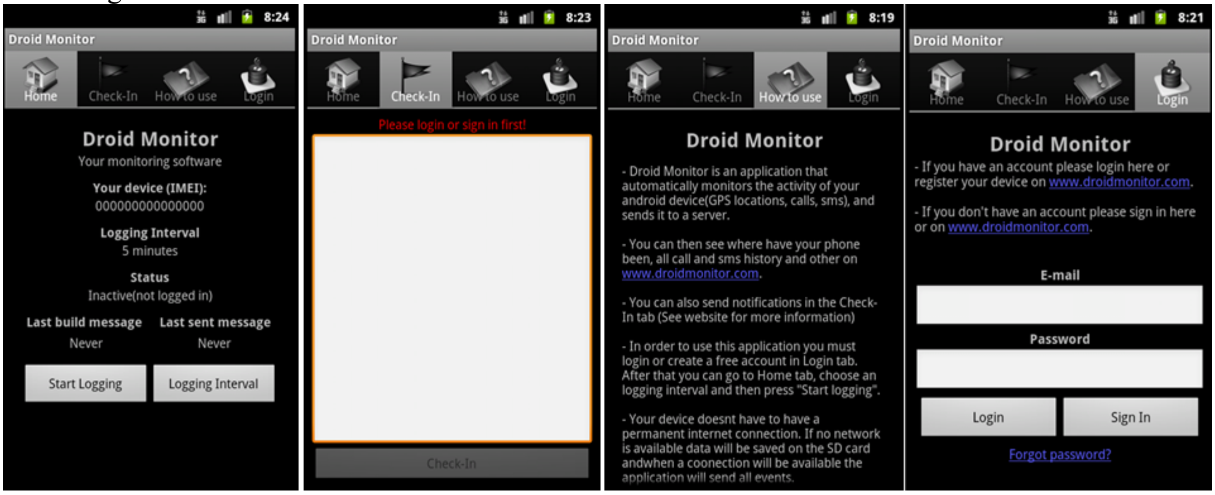
The transfer is made using the TCP protocol. It is relevant to point out the fact that the application does not require a permanent internet connection. Whenever an event is identified, the internet connection is verified, and if it works, the transfer is completed. Otherwise, the data are saved in a folder on the external card of the device, and they will be sent exactly at the moment the internet connection is made. Despite this fact, the average amount of data sent every month is 5-10 Mb, which equals a low traffic input. Consequently, no piece of information is lost, only the transfer process will be delayed. This approach is profitable not only for the durability of the battery (fewer connections means more lastingness for the battery), but also for controlling the network traffic (generally, the traffic is limited for the mobile devices).

3.3.4 User

The application was created in such a way, that the number of interventions per user is minimal. All that needs to be done is to install the application and to authentify/register using the tools from the interface. After that the application will be accessed only when the user would want to send a message (check-in) or to verify the data status.

3.3.5 User's interface

As mentioned in the previous chapter, this application is based on services, and the user's interference is minimal. Nevertheless, the structure and complexity of the interface is the most important one, so that the application to be accessible to any type of user, irrespective of the knowledge he/she has in the field of informatics.



Droid Monitor
Your monitoring software
Your device (IMEI):
000000000000000
Logging Interval
5 minutes
Status
Inactive(not logged in)
Last build message
Never
Last sent message
Never

Please login or sign in first!

Droid Monitor

- Droid Monitor is an application that automatically monitors the activity of your android device(GPS locations, calls, sms), and sends it to a server.

- You can then see where have your phone been, all call and sms history and other on www.droidmonitor.com.

- You can also send notifications in the Check-In tab (See website for more information)

- In order to use this application you must login or create a free account in Login tab. After that you can go to Home tab, choose an logging interval and then press "Start logging".

- Your device doesnt have to have a permanent internet connection. If no network is available data will be saved on the SD card andwhen a coonection will be available the application will send all events.

Droid Monitor

- If you have an account please login here or register your device on www.droidmonitor.com.

- If you don't have an account please sign in here or on www.droidmonitor.com.

E-mail

Password

Figure 2. Android application tables

3.3.6 Java language

The choice of selecting Java language was mandatory, due to the fact that Android applications are created with Java libraries. The IDE was Eclipse, because it places in hand all the tools that are necessary for the exploitation and testing of Android applications.

3.4 Describing the server

3.4.1 Generality

In order to make the connection between the Android application and the one on the website an intermediate system was required, that had to fulfill the task of transferring and stocking the data. In this matter, a server (Windows Form Application) was built, that was programmed to run on the same device as the website server. The server is permanently running, and it detects the signals sent by the mobile devices on a specific port. After the signals are processed and the pieces of information are saved, the users can check their information by means of the web application.

3.4.2 Data transfer

Data transfer is processed using the TCP protocol. When starting the server, it automatically launches a thread specialized in capturing the signals on certain port. The instant a message is received, another thread is activated, in order to verify and save the information. This manner of implementation permits the server to receive a large number of messages simultaneously.

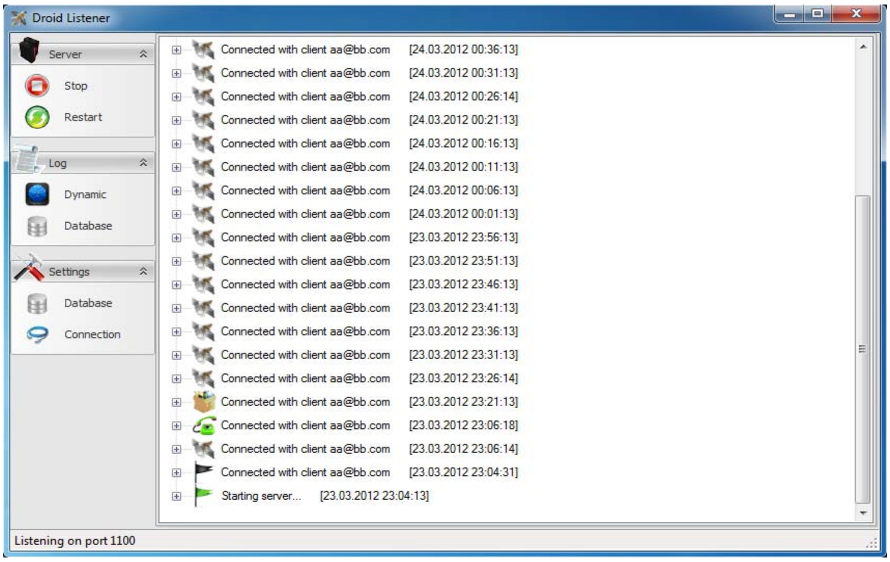
3.4.3 Database

The database server that was selected is SQL Server 2008 R2, because it offers a series of tools that are essential to the development of an enterprise application. Also, *Express Edition* version is mandatory, for it stipulates an exponential growth of the volume of information received from the mobile devices. *Express Edition* version owns a license that permits the existence of unlimited space for the database.

3.4.5 User's interface

The server application can only be accessed by developers and/or the administrator. This was the main reason for adopting a friendly interface, that will be helpful for:

- Immediately detection of the problems encountered during the developing and testing process of the product.
- A thoroughly control of visualizing data transfer.
- Changing the settings of connectivity and those belonging to the database are easy to handle.
- Giving right instructions to anybody who wants to fulfill the task of administrating the server.



```
Connected with client aa@bb.com    [24.03.2012 00:36:13]
Connected with client aa@bb.com    [24.03.2012 00:31:13]
Connected with client aa@bb.com    [24.03.2012 00:26:14]
Connected with client aa@bb.com    [24.03.2012 00:21:13]
Connected with client aa@bb.com    [24.03.2012 00:16:13]
Connected with client aa@bb.com    [24.03.2012 00:11:13]
Connected with client aa@bb.com    [24.03.2012 00:06:13]
Connected with client aa@bb.com    [24.03.2012 00:01:13]
Connected with client aa@bb.com    [23.03.2012 23:56:13]
Connected with client aa@bb.com    [23.03.2012 23:51:13]
Connected with client aa@bb.com    [23.03.2012 23:46:13]
Connected with client aa@bb.com    [23.03.2012 23:41:13]
Connected with client aa@bb.com    [23.03.2012 23:36:13]
Connected with client aa@bb.com    [23.03.2012 23:31:13]
Connected with client aa@bb.com    [23.03.2012 23:26:14]
Connected with client aa@bb.com    [23.03.2012 23:21:13]
Connected with client aa@bb.com    [23.03.2012 23:06:18]
Connected with client aa@bb.com    [23.03.2012 23:06:14]
Connected with client aa@bb.com    [23.03.2012 23:04:31]
Starting server...    [23.03.2012 23:04:13]
```

Listening on port 1100

Figure 3. Server interface example

3.4.6 C# Language

C# language was prefered to be used, due to the fact that it constitutes an object-oriented programming language, with a C++ oriented syntax, which includes aspects of Delphi, Visual Basic and Java languages; a special emphasis is placed on the simplification of the matter in discussion (fewer symbols than in C++, fewer design tasks than in Java et cetera) [5, 6].

3.5 Describing web application

3.5.1 Generality

Nowadays, the most efficient way of visualizing data about a certain entity is websites. During the planning process of the concept it has been decided to develop a web application in order to make the mobile devices information available. The way of visualizing data, the functionalities and other important details are discussed in the following subchapters.

3.5.2 Functionalities

The web application is available on www.droidmonitor.com, and is currently in a continuous expanding process. One part of the functionalities that have been selected is in course of expansion, and another part lies on a list of near away future objectives. The major objectives are:

- Displaying GPS locations by means of interactive maps (Google, Yahoo, Bing, and Open Street).
- Displaying all events using GPS localizations in a similar way as stated above.
- The possibility of authentication or registration of the users.
- Choice of language.
- Administration of the user's account, personal devices or contacts.
- Chance of generating personalized reports.
- The possibility of generating statistics in regard to personal devices.
- E-mailing service that permits the user to send an appropriate feedback to the developers (in progress).
- Broadcast service that will furnish reports to the user about his/her devices (in progress).
- Facilitating friendships/partnerships between the users, such as Facebook (in progress).
- Other functionalities that exist on the list of submitted objectives.

3.5.3 User

The user represents the central entity of the project. The functionalities of the web application have been built depending of his necessities. As reminded when stating the problems in question, the pieces of information we are working with are quite delicate and they may become unsafe. Implementing an authentication and registration method was compulsory.

In order to register the user has to follow some elementary steps. The data introduced by the user are: username and password. Additional information, such as name or address, are optional, and when required they serve in making statistics.

3.5.4 User's interface

Within a web application functionality is as significant as the interface. This has to be as simplest and easy to handle as possible, because the product is dedicated to any users, despite their experience in the branch of informatics. Pages with suggestive names and descriptions were created for each action. All buttons have personalized icons and noticeable writings/names, in order to be at fast reach when accessing it on a mobile device. The main page contains a map that locates personal devices, as in the image bellow.

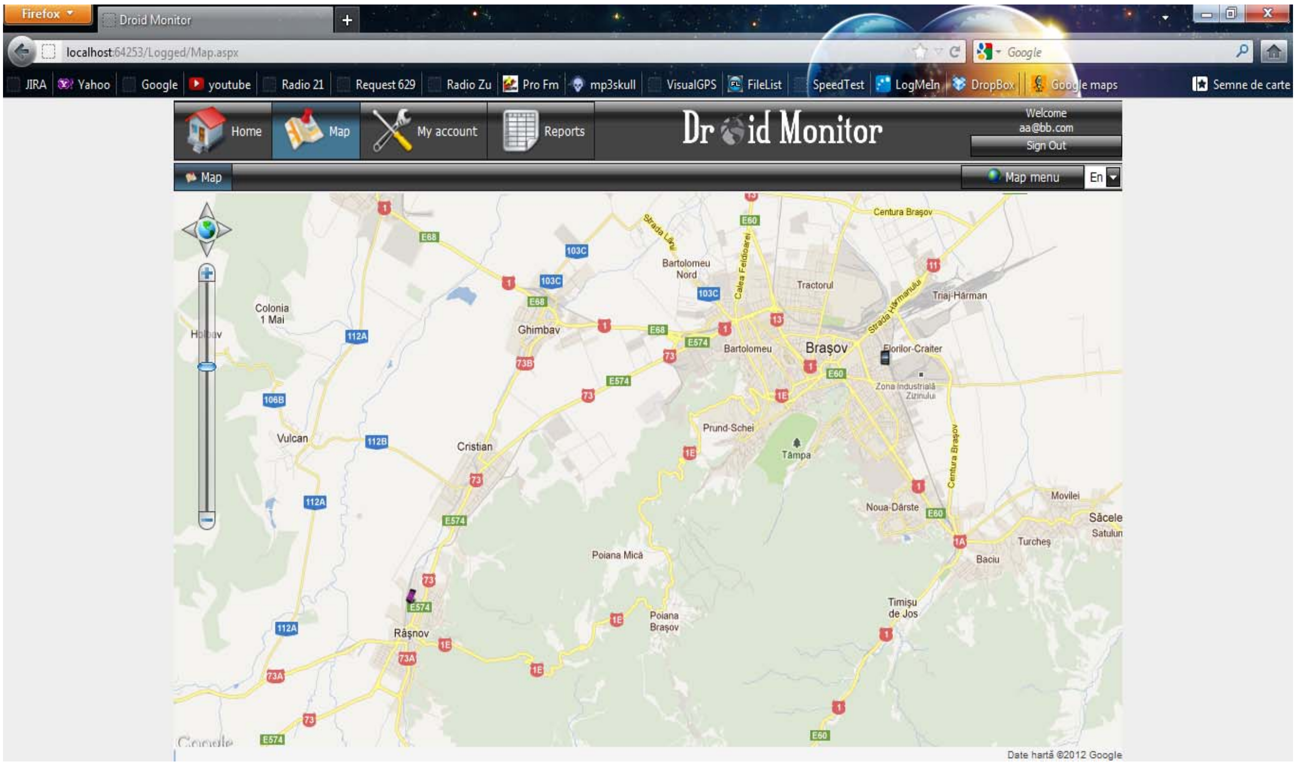


Figure 4. Website main page

3.5.5 Maps

The application is dedicated to a large number of users, so the number of maps preferences also increases. For simplifying the matter, there are more types of maps that have been adopted, which are to be modified from the user’s settings:

- Google satellite
- Google maps
- Yahoo satellite
- Yahoo maps
- Bing satellite
- Bing maps
- Open street

3.5.6 Graphics

The web application offers a series of statistics tools. It has been ascertained that the best method of presenting the results of statistics are graphics. They have been implemented in this type of websites, for better illustrating the data. One type of graphic is shown in the picture below:

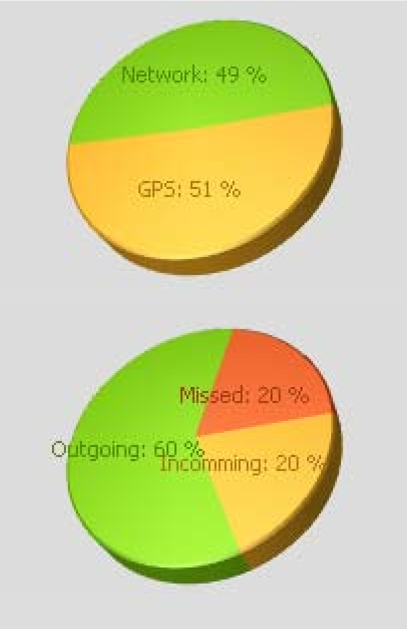


Figure 5. Example of statistics

3.5.7 Reports

This model of application can be also utilized by companies. For instance, when an administrator wishes to monitor the activity of employees, they can use the product on Android mobile devices. In this case, different reports will be officially/unofficially generated at the end of the day/month, depending on the company's needs. The manner of generating reports is for the moment in a developing phase.

4 Conclusions and future developments

There are two main reasons for creating this project: one is didactical and the other productive. When analyzing the results, the conclusion has been drawn, that the majority of the above mentioned objectives were accomplished. If in the didactic sector the objectives were finalized at the productive level, then the last step will remain on measuring the success of the product on market. Consequently, only the final user may conclude in a realistic way upon this project.

Even if the majority of the above mentioned objectives have been solved, others might appear from the feedback received from the users themselves. Nevertheless, the next step of developing the project is publishing the application on Android Market and administrating correctly both website and database.

5 Notes

The current form of this paper was completed on 25/03/2012. There is great probability that the application might have suffered changes until the actual moment of presenting the product. Mainly the modifications that may occur consist in adding new functionalities to the website application.

Bibliography

[1] James Gleick, *The information: A History, A Theory, a Flood (Vitage)*, Kindle Edition, 2012

[2] Dorin Bocu, *Ingineria Softului,Cur*s, 2008

[3] Adrian Florescu, *Sisteme inteligente*, Editura Noua, 2003

[4] Sebastian Corbos, *Java for dummies*, Kindle Edition, 2001.

[5] Lucian Mircea SASU, *Introducere în limbajul C#,* Transilvania University of Bra□ov Press, 2009

[6] Dragoş Iulian Obancea, *Primii paşi cu C#*, Editura Noua Bucureşti, 2009.

Dragoş Iulian Obancea
Transilvania University from Braşov
Facultay of Mathematics and Informatics
Informatics Department
Street Iuliu Maniu, Nr. 50, Braşov
ROMÂNIA
obancea_dragos@yahoo.com

Simulation of information influence of social networks on client`s loyalty by the AnyLogic package tools

Svetlana A. Panasenko
Teacher Coordinator: I. A. Ushakova

Abstract

The main directions of social networking influence on the customer relationships, specifically the formation of customer loyalty are determined. Details of modeling of social networks and agent-based modeling of information influence of the social networks on customer loyalty are considered. Multiagent model of customer loyalty influenced by the dissemination of information in social networks is built. The trend of the potential customers` changing state in real time within the model is defined by the model.

1 Introduction

1.1 Problem definitions

The global network is well known and popular throughout the world and a lot of people come to the Internet every day and attend social networks. Business can`t ignore such popular structure, and therefore it needs to participate in this conversation. But nowadays it is very difficult to choice the way of achieving it. Therefore, the areas of formation of customer loyalty through the social networks is one of the perspective areas of business and researchers and business pay more attention to them (social networks). The advantage of participating organizations in communication with customers in the social networks is a large number of them on the WWW. For example, it is easier and cheaper for information technology companies to download fresh videos of updates to Facebook, than to join a conference, and for developers of games to distribute news about the new version to fans via Twitter, and for manager of sales to communicate with colleagues and customers via Linkedin.

The modern century of new information technologies has changed the understanding of communication. Recently companies separated and even neglected social networks and Internet at all, but now they develop their own similar Internet projects. Nowadays many companies are joined social networks, but they do it for advertisement. However, monitoring the customer relationship and the formation of their loyalty can be increased in social networks. This is new management district called Social CRM (SCRM - social customer relationship management).

Companies can form and increase customer loyalty via the Social CRM as certain category of potential customers spend a lot of time in social networks [6].

1.2 Article purposes formulation

Accordingly, the purpose of the article is to analyze the impact of social networking information on the relationship with clients to use it to create a multi model of customer loyalty.

2 Main Part

2.1 Social Networks developing

Social Network - social structure, which consists of groups of nodes, such as social objects (people or organizations), and relations between them [5]. The term "social network" was introduced long before the Internet and Internet networks in 1954 by James Barnes (the sociologist from the "Manchester School").

Modern concepts mean a range of familiar man, where man himself is the center of the social network, its familiar - branches of it and relationships between people - relations. If the social network is reviewed more deeply, it will be found that relations are divided by type: unilateral and bilateral; network of friends, colleagues, classmates, etc.

In the second half of the 20th century, social networks have been actively developed as a scientific concept, they first became popular in the west, later came to us. Then this common concept of professional sociologists has become a fashionable concept, which is a central concept in Web 2.0.

The first computer social networks are all the same group of people that used tools of computer communication to create and maintain social relations, which became e-mail. It happened on the 2nd of Oct.ober, 1971 - the day when the first message was sent to a remote computer, and the first social network users were military in ARPA Network.

The next step was the invention of the IRC (Internet Relay Chat) - server system to communicate in real time. IRC or "Irka" as it is called, was established in 1988 by Finnish student Vibrant Oykarinenom. These were more "advanced" social networks, but still far from modern [1].

Internet was invented on the 7th of August, 1991 with the invention of computers, email, IRC and many other. That day the British scientist Tim Berners-Lee published first online pages and did the same the next step with modern social networks.

In 1995 appeared the first, close to the modern, social network Classmates.com, which made Randy Conrad, who is owner of Classmates Online, Inc. This website helped registered visitors find and keep in touch with friends, classmates and other acquaintances. Now there are more than 40 million people in this network mostly from the U.S. and Canada.

Classmates concept was successful and since 2005 it has been developing, and not only within this network such world giants as MySpace, FaceBook, Bebo and LinkedIn or giants Runet Odnoklassnyky.ru, Twitter, and Moymyr MoyKruh are appeared.

All social networks are divided by type. There are networks to find people: classmates, colleagues and others. There are business networks for job search partners, professional communication and others. Some networks are based on video, some on audio and music specifically, and some on the photo. There are networks that can take some isolated niche and can`t be on the above categories [1], [4].

Also conditionally networks can be divided by geographic orientation: global or country specific.

In addition, there is different attitude to the policy of openness of information in the different networks. Most networks currently open, but there are closed, where people come by invitation only. Closed networks are just beginning to appear, but you can expect their popularity a few years because people like all forbidden and inaccessible.

Social networks can be divided by the developing level into Web 1.0 - the first social network of basic functional, Web 2.0 - advanced social network with a wide functionality for communication and Web 3.0 - socio-scattering potential gave rise future networks that solve specific problems.

Social networks are the Internet phenomenon of the 21 century. In October 2011 the number of unique visitors to Facebook has made 750 million and the number of page views - 1000000000000. Revenues for Facebook in 2011 for their own assessment of the company amounted to 5 billion U.S. dollars. And this is just one, although the most popular social network, with hundreds of English-speaking social sites that continue to developing incredible rate. Let only imagine the number of users of global network in Asian region and developing countries.

Hundreds of millions of users who speaks on dozens of languages and thousands of companies who simply have no choice but to place their advertisements on the sites of social networks - and this is not real, and in the virtual world of computers, networks with the relevant instrumental tools of communication, storage and dissemination of information.

2.2 Formation of Customer Loyalty

Loyalty is characteristics of the client, the buyer, which define its commitment to a particular vendor, brand, etc. The purpose of the modern strategy of customer relationship management (CRM) is building a strong and lasting relationships with customers, shaping their loyalty. These relationships should be beneficial to both parties - and the company itself and its customers. To achieve this companies should makes for the customer all that is required, best in the minimum period for a reasonable price, but not at a loss.

Impersonal character "Customer" animates only if a specific person. If the customer acts as a company, the case will be carried out with different people performing different roles and have different, even opposing interests. The most important are the following roles within the company-customer: buyer - one who can buy or acquire goods or services, the customer - who pays for the purchase of goods and services, user - someone who uses goods and services which was purchased in your organization, customer - who is somehow related to your company (this includes all previous roles). If the "greedy" user wants to get bigger and faster, the "stingy" customer limited material benefits and other features.

The rapid development of social CRM is caused primarily by the possibilities offered by social networks for attracting potential clients and formation of their loyalty. The main advantages of social networks with traditional CRM are the follow:

availability of information about potential customers in social networks. Social network is filled with information about potential customers. For example in the social network "in contact" in January 2012 daily audience exceeding 35 million users, in social network Facebook - over 800 million users. Thus, each resident of the planet, which is present in the social network is taken into account in it with all contact information, preferences, friends, etc. This does not mean that automatically you can get all information about the user social network. However, modern search tools allow you to solve these problems, which are quite developed in the same social networks. For example, by e-mail can be automatically tracked all the user registered contacts. It is much easier and faster than the complete customer base in CRM;

availability of appropriate communication tools to work with customers in social networks. In social networks a special mailer to send messages do not need. Tools correspondence network users built into the same network;

guaranteed delivery of messages to the recipient. The recipient just sees a message from the network, it will not be removed or detained in spam filters, as links in a network are friendly, and block friends` messages is mistakenly;

easiness of access to information. If you want to see the history of correspondence from specific people - just go to his profile and watch. There is also no sense to withhold information from friends;

speed of information dissemination. If a marketing event is planned it will be written in a social network, invite friends to visit, a list of contacts that responded. All this in just a few clicks, much easier than in traditional CRM-systems. To extend the news about company, product, new marketing initiatives, etc., simply put information online, and invite friends to visit her. Established companies have no corporate site, but there are several pages on Facebook. Indeed, why should create and maintain a website where you can immediately promote information in an environment where there are potential customers;

low cost. There are no competitors for social networks. Registration and most of the features are free. Compare the cost of any, even the cheapest, CRM-systems. Interface is intuitively simple and understandable, and not have to pay a specialized company for implementation. But more importantly, that does not need to care for and bear the cost of maintenance of the information system. It is obvious that its service specialists rather high level, and servers that cost is likely to significantly exceed the annual turnover of small businesses;

technological reliability. Social Network is available 24 hours a day, 7 days a week, has interferfaces in all common languages, it can work with virtually any device connected to the Internet and a browser. Of course, no social network, at least, is free, does not guarantee that your data will never get lost, there will be failures, hackers do not break your information, but practice shows that incidents of this type almost never happens. And other things being equal opportunities reliability CRM-system in which looks small IT department or simply a system administrator is likely to be lower than in any social network;

low costs of maintaining customer information up to date. In the case of CRM-systems need to spend at least this time employees. In the case of social network users update the information yourself, or at least have the opportunity to upgrade it yourself, if you ask them;

wide coverage of the youth audience. This class of potential employees who are familiar with work in social networks and are likely to be interested to continue to do what and still get paid for it.
So the obvious direction of forming customer loyalty can be the development of CRM-systems and their integration with social networks, that use Social CRM.
Social CRM is a tool that contributes to a better, more effective interaction with the client and uses the collective intelligence of a broader client community with predictable improved contacts between the organization and its prospects and customers. The purpose of Social CRM is to build closer relationships with customers and bind them to the company by creating public-ecosystem, for better understanding of what they want and how they interact with different points of contact of the company, for example, sales, customer service and etc.

2.3 Social networks modelling

Modelling the behaviour of potential customers in the distribution of advertising in social networks is an integral part of improving customer loyalty through Social CRM is. Network analysis is a natural distribution of the technical concept of "structure", understood as a set of elements and connections between them, to research methods in anthropology, social psychology, sociology, economics, geography and political science.
The initial intuitive understanding of the network were gradually expressed in the language of algebra, graph theory and probability theory. Specific networking methods (Based on mathematical methods in sociology) that facilitate the inclusion of sociological concepts in the conceptual core of other disciplines have developed [3].
There are three key areas in the theory of social networks:
- Search statistical properties that characterize the behaviour of systems with network structure;
- Creating network models;
- Predict the behaviour of systems with network structure based on the measured structural properties and local rules of the individual structures.

Simulation - the study of objects of knowledge in their models, building models and studying the real existing objects and phenomena (living and nonliving systems, engineering designs, various processes - physical, sings, chemical, biological, social) and constructed objects (for identify, clarify their characteristic of the characteristics, rationalization of their construction methods, etc.).
Model (from the Latin modulus - a measure, sample) - simplified representation of phenomena or objects actually related to nature and society in the form of diagrams, pictures, descriptions, mathematical formulas of any real object (phenomenon or process), seen as their equivalent.
The model performs the following functions:
- Cognitive - allows to look into the essence of the phenomena which are being studied to better understand them;
- Forecasting - to predict the future of real object model;
- Decision-making;
- Improvement of measurement.

All these processes are carried out by communication and information technologies, which are technical tools of interaction. Carriers interactions are information flows which are distributed in the information environment of social networks.
In virtual space, which is the basic medium for interacting with customers, communication between people are made as follows:
through the distribution in space of information flows;
as collecting and processing information received,
through direct exchange of information between the participants of the economy.

2.4 Agent modelling of social networks information effects on customer loyalty

Agent modelling techniques can be used to simulate the impact of social networks information on customer loyalty. Agent Modeling (AM) - a new approach to modelling systems that contain autonomous and interacted agents. AM can be used in business decision-making. AM can give good results when it is used by researchers for computer simulation of processes in social networks [2].
Agent is entity that owns the activity, autonomous behaviour, can make decisions respectively with some set of rules, can interact with the environment and other agents and may change (evolve). Multiagent (or simply agent-

based) model used to research decentralized, dynamic operation of them is not determined by global rules and laws, but these global rules and laws are the result of individual group members activity. The purpose of agent models is an overview of the overall system behaviour based on knowledge of behaviour of its individual active objects and interaction of these ones in the system. Agent model can contain dozens and even hundreds of thousands of active agents.

package AnyLogic has been used for a better understanding and visualization of information influence modelling of social networks on the formation of customer loyalty [3].

To analyze the impact of social network information on Customer Loyalty agent model will have two states - a potential customer (Potential Adopter) and existing (real) client (Adopter). The potential client - a customer who could buy products, but certainly his intentions aren`t known. The real client - a buyer who made a purchase. The next model parameters will be used: the size of social network, type of social network, trust degree and agent modeling time.

Size of network is set for 1000 agents. It is sufficient to obtain clear results for social network. As a parameter of the transition from potential to actual state of the client was accepted trust degree to agent as one of the main factors of influence on customers` loyalty is thought the agent to whom they trust. Modelling carried out for 8 units of model time. The model is able to transition from a potential to real state, but not vice versa (Fig. 1).

On the Fig. 2 the model which is built by the tools of package AnyLogic is presented at the first and last day of the dissemination of information via social network, respectively. Blue people are potential clients, red - real.

During modeling the graph was constructed (Fig. 3), which shows that the number of potential customers is decreased (top chart) against that the real number of clients is increased (bottom chart). Due to influencing of information dissemination by agents of social networks share of real customer is increased each unit of model time.

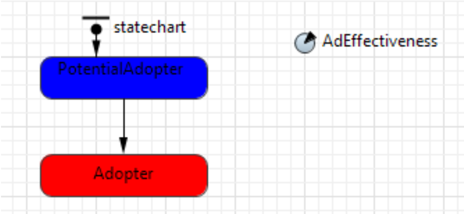


Figure 1 Statechart diagram of customer loyalty which is built by the tools of AnyLogic package

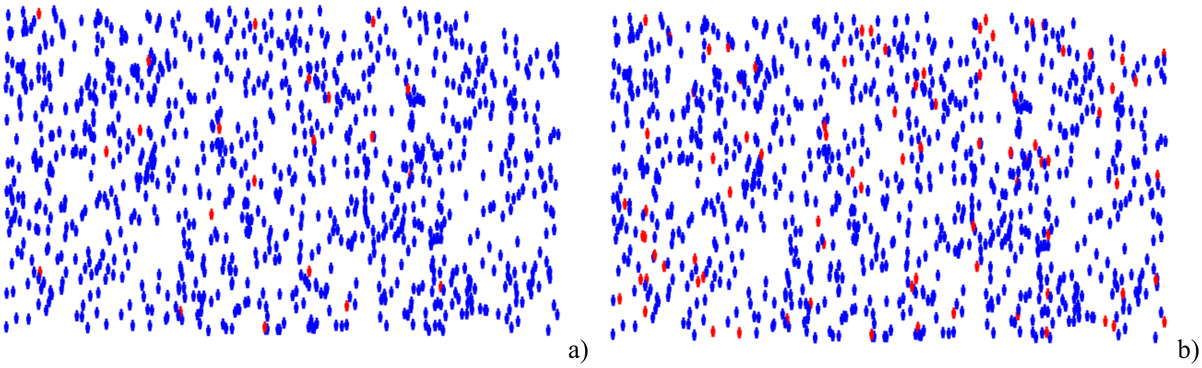


Figure 2. Model of Customer Loyalty on the initial (a) and final (b) moments of model time

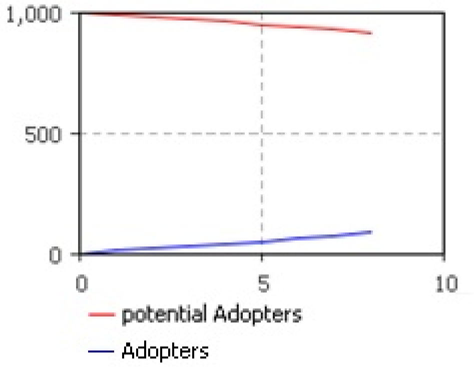


Figure 3. Quantity dynamics of potential and actual clients graph

3 Conclusion

Modern marketing strategy is directed to attracting and keeping customers, finding opportunities to increase sales. That is why modern business is interested in converting potential customers into real ones.

Important impact on attracting new clients is made by social networks. It is needed to simulate the impact of information on potential customers to predict future customer engagement through social networks.

Modern tools of simulation, which have many features are package AnyLogic. The model of customer loyalty has been built with the help of this package. The model will help to analyze the formation of customer loyalty by distributing any information via social network. The result modelling information will allow to predict the growth trend of real clients which is influenced by the distributed information by the social networks.

References

[1] D.A. Gubanov, D.A.Novikov and A.G.Chhartishvili. *Социальные сети: модели информационного влияния, управления и противоборства*. Phizmatlit Moscow, 2010.

[2] K.N.Mezentsev and A.B.Nikolaeva. *Учебное пособие «Моделирование систем в среде AnyLogic 6.4.1»*. MADI Moscow, 2011.

[3] S.I.Parinov. *Новые возможности имитационного моделирования социально-экономических систем.* Internet Magazine "Искусственные сообщества" 2007.

[4] *Свойства социальных сетей.* Electronic resourse, *http://www.sd-company.su/article/computers/social.*

[5] *Социальная сеть.* Electronic resourse, *http://termin.bposd.ru/.*

[6] *What is Social CRM?* Electronic resourse, 2011, http://thesocialcustomer.com/leighdow/36230/what-social-crm.

Svetlana A. Panasenko
Kharkiv National University of Economics
Information Systems
61001, Kharkiv, Lenina Avenue 9A
UKRAINE
pansvetik6@yandex.ru

Multi-User Drawing Board

Iulia Paraicu
Teacher Coordinator: Alexandru Radovici

Abstract

The **Multi-User Drawing Board** [4] is a JAVA application that allows the user to connect a conference using Laptop or Desktop. The participants can all draw and send text messages to each other on a shared whiteboard. A real-time communication between users is possible with the help of XMPP [7]. The concept of this project can apply to virtual project meetings, real-time collaborative problem solving, and remote visually oriented mathematics tutoring.

1 Introduction

In 1996 Mark Weiser [1] defined the computer-human interaction trajectory. It can be thought of in three waves:

- The first computing wave tied many people to a single mainframe computer. Users of such computers had highly specialized skills that were not representative of average citizens.
- The second wave connected individuals to desktop and laptop computers, providing a one-to-one computer-to-human ratio.
- The third wave is the era of ubiquitous computing, whereby many computers interact with one person, or many computers interact with many people.

In my opinion, nowadays we reached the third wave. Even if some people are still manifesting one-to-one interaction with their computers, future comes with the ubiquitous computing [10], hundreds of wireless computing devices of different sizes in the same room. Contemporary devices that lend some support to this latter idea include mobile phones, digital audio players, GPS, and interactive whiteboards. We notice that the grand companies and prestigious universities are more and more interested in using this concept, to improve their efficiency.

The main objective of our project is to develop a drawing board that can communicate with similar applications from other platforms to create a conference. This application facilitates the communication between users and has the following functions:

- allows the client to enter, send and receive the data at the same time;
- offers a friendly drawing environment;
- gives the user the possibility of expressing himself utilizing the various tools;
- is flexible, so users can join or leave the conference anytime they want.

2 Related Work

This whiteboard [5] allows people to easily collaborate and solve problems, by sharing visual information in real-time. The program can be used to improve areas in business, communication, education, and recreation.

Firstly, the drawing-board application is helpful for companies. By using this program, the communication with their employees, their customers, and possible recruitment candidates is improved. Many business groups communicate using a form of voice conference. When they have to share visual information, it has to be sent to each participant or to be stored in a place where everyone has access. Then the image is explained as the participants talk. The explanation process is easier due to this program and its applicability. They can use the drawing tools to emphasis the part of the image that they need to focus on or to highlight a potential problem. The whiteboard can also be used to interview possible candidates for employment.

Students and pupils can also benefit from the multi-user drawing board, which perfectly suits the education system. Subjects, such as geometry and physics, use a lot of images and diagrams to describe problems. Many subjects also require advanced equations that contain many foreign letters and symbols. Explaining these images and equations in words can be a difficult task for teachers and the students don't always understand the phenomena. This drawing board provides a place to draw out images and equations for students, who would work together and better understand their schoolwork. Students will also be able to collaborate to solve difficult problems. A system can also be set up to provide tutoring in various subjects. Anyone that is having trouble would then be able to use the program to connect to someone that can help. They can then see as the problem is explained to them visually and in real time.

The drawing-board program can be use not only for work and education, but also for recreation. People spend an important part of their spare time chatting, but they are often limited to just text. By using this whiteboard they are able to add images to further express themselves and to share memories or life experiences. There are some applications on the Internet that allow people to play simple games such as Tic Tac Toe, Pictionary, and Hangman. This application gives user the opportunity to draw original game environments and play all kind of games.

3 Technologies

All the applications in this project are developed using JAVA [2]. This is because JAVA is object oriented [3] and platform independent. One of the most significant advantages of Java is its ability to move easily from one computer system to another. By using Java, one program can be run on many different platforms· This means that you do not need to put your efforts on developing a different version of software for each platform.

For the graphical user interface (GUI) [8] were used Java Swing [6] and AWT [9] libraries. Swing is an advanced GUI toolkit. It has a rich set of widgets. From basic widgets like buttons, labels, scrollbars to advanced widgets like trees and tables. Swing itself is written in Java. The main characteristics of the Swing toolkit are: platform independent, customizable, extensible, configurable, and lightweight.

The communication between clients is realized using XMPP [7] (Extensible Messaging and Presence Protocol). XMPP is a protocol based on Extensible Markup Language (XML) and intended for instant messaging (IM) and online presence detection. It functions between or among servers, and facilitates near-real-time operation. The protocol may eventually allow Internet users to send instant messages to anyone else on the Internet, regardless of differences in operating systems and browsers.

3 Drawing-Board Implementation

The whiteboard consists of a toolbar on the left of the window and a drawing canvas in the center. The user can draw lines, various shapes, move them and change their dimensions. The tools also contain a free hand option; prefigured shapes and the user can add images to the canvas. The application has a large range of colors and offers the freedom of using more than one page.

Even if similar applications exist at the moment, the multi-user drawing board is an innovation because it comes up with solutions to previous problems. Therefore the clients can enjoy a functional and easy to use whiteboard that has some obvious advantages.

The program was developed in Java, which is platform independent, so the application can be used by any operating system. In this way the clients can benefit of the whiteboard without conditions or restrictions.

The XMPP technology allows the application to connect to similar programs from different platforms without the help of a server, so the user is able to make a conference anytime and from anywhere using his PC, cell phone or tablet.

We want our clients to be able to solve their problems fast and easy, therefore we managed to let them move or resize each drawing without doing any other changes on the page. That because all drawings (lines, shapes, images) are objects. Every time the user draws something, the object is added to a vector, so if the user wants to move or resize the drawing, the object will be accessed from the vector.

Nowadays working at a school or business project requires a lot of space to express different ideas. This application provides the client a large drawing space organized in pages, in this way everybody can show his own opinion without destroying others work.

4 Future Improvements

Though this application provides many functions, it can be improved. I am thinking of adding the following options: copy-paste, shape rotation and change color for shapes and background. My colleagues and I are also thinking of developing a similar application for browser.

5 Conclusions

This application is developed to facilitate the process of sharing information for work, education or recreation. The users can join or leave a conference whenever they want to. The whiteboard provides them a large range of shapes, colors and options. They can draw free hand, lines, and shapes, add prefigured shapes and images, resize or move them, add text and hide their work from others.

The main advantage of this whiteboard is that it can communicate with similar applications for tablets and mobile phones that were developed by my colleagues.

References

[1] Mark Weiser, *The Computer for the 21st Century*. Scientific American, September 1991
[2] Doug Bell, Mike Parr. *Java for Students* (5th Edition). Prentice Hall
[3] Bertrand Meyer, *An Object-Oriented Environment: Principles and Application*
[4] Sailaja Rudraraj, *DrawingBoard Conference between Multiple Clients*
[5] Dr. Ramesh Yerraballi. *Multi User Collaborative Whiteboard*. The University of Texas at Austin, Spring 2009
[6] ***, http://zetcode.com/tutorials/javaswingtutorial/, accessed on 25.03.2012
[7] ***, http://xmpp.org/about-xmpp/, accessed on 28.03.2012
[8] ***, http://cs.nyu.edu/~yap/classes/visual/03s/lect/l7/, accessed on 29.03.2012
[9] ***, http://docs.oracle.com/javase/1.4.2/docs/api/java/awt/package-summary.html, accessed on 29.03.2012
[10] ***, http://en.wikipedia.org/wiki/Ubiquitous_computing, accessed on 28.03.2012

Iulia Paraicu
Universitatea Politehnica Bucuresti -Facultatea
de Inginerie in Limbi Straine
Calculatoare si tehnologia informatiei - franceza
ROMANIA
iulia.paraicu@gmail.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Tank::Shooter

Popa Adrian, Wandschneider Josef-Alexander
Teacher Coordinator: Simian Dana

Abstract

In the following we will describe the basic structure of representing 2D and 3D objects in OpenGL and the basic idea of how to structure a game.

1 Introduction

In order of testing the possibility of representing graphical elements on the PC while using our C/C++ knowledge we acquired at our university, we made a basic and interactive game, utilizing the OpenGL [1] graphics libraries. It allows two players to control one tank each with which they interact with one and other. The game's scope is to obtain more points than the other.

Furthermore the game uses a 3D-model imported from a 3D-modelling tool, in this example 3DSMax.

2 Construction

2.1 Components

2.1.1 C/C++

The code is a combined structure of procedural, modular and object-oriented programming style. There are four big components of our design structure: main, control, draw and model. Our aim was to keep the main-function clear and easy-to-understand. As a result of this it is now very easy to add and modify certain parts of the whole program.

The structure made it simple for us to work as a team on the same project without having to be in the same location or work at the same time.

2.1.2 OpenGL

OpenGL provides the programmer with the necessary graphics base for drawing, placing and displaying the objects you want in a 3D space. It is an open source library which is widely used in modern industry and the source code can be compiled for Unix/Linux and Mac systems. OpenGL doesn't depend on a specific environment or platform like DirectX does. OpenGL uses different data types and functions than standard C and C++. The creation of a black, cleared window in Windows needs a lot of lines of code. Luckily, there are a lot of projects - like GLUT/FreeGLUT [2] - which provide the programmer with some functions for creating the cleared window by writing 2 lines of code.

A basic object like a cube consists of 6 squares. Each square is made out of 4 points with the specific x, y, z coordinates.

2.2 How to play

The first player, on the left, controls his tank with the keys: WASD and shoots with space. The second one, on the right, controls his tank with the arrow keys and shoots with the right control key.

In addition it is very easy to modify certain elements in the game. By pushing E or R the map size is changed. F and G change the size of the obstacles and bullets. T changes from shooting one single bullet to a row of bullets.

The camera's position is changed by pressing I, J, K or L.

Finally, the keys ESC or Q exits the program.

In the upper left hand corner you can see a FPS indicator which shows the current frames per second. Below that there is an indicator showing the time since the game was started. In the middle, you can see a score. The bottom of the game contains the controls.

Our collision control makes it impossible for tanks and bullets to pass through walls and obstacles.

We have implemented an Easter Egg which changes certain things in the Christmas month.

3 Program Code

3.1 main

As usual the source code contains one basic main function, a must-have in every C code. You can see the glutInit()-functions which are responsible for creating the window and drawing the scene. The scene will be redrawn on certain amounts of time for displaying the events. These are for example object movement like shooting a bullet or driving the tank through the map.

In addition to the main function, there is the drawScene()-function which is responsible for drawing the objects we want to see. The object information and its drawing methods are stored in the draw.cpp file.

3.2 draw.cpp

This file contains some very important functions.

On the one hand there are the functions for creating basic shape objects like cubes and a simple plane surface. We generated the entire map using these objects.

The plane is considered to be a filled area between four points. The point is called "vertex" and is characterized by its coordinates. The following code-excerpt will demonstrate this procedure:

```
glBegin(GL_QUADS);
    glTexCoord2f(1, 1);
    glVertex3f(-size, 0, size);
    glTexCoord2f(0, 1);
    glVertex3f(size, 0, size);
    glTexCoord2f(0, 0);
    glVertex3f(size, 0, -size);
    glTexCoord2f(1, 0);
    glVertex3f(-size, 0, -size);
glEnd();
```

As you can see glBegin(GL_QUADS) sets the whole following instructions for vertexes to be considered as parts of the a plane, which is basically a filled rectangle in space. glTexCoord2f() is a texture-binding instruction. glEnd() ends the drawing method initialized by glBegin().

Now the next step was building whole 3D-objects like cubes. Cubes are the product of 6 rectangles with the vertexes put in a specific order.

The tank which is moved by the player is a .3ds-model. Therefore it was necessary writing the source code in C for importing the data from a .3ds file. The lib3ds-library was a big help.

On the other hand there is a function – drawText(const char *text, int length, int x, int y) which makes it possible for us to present strings where we choose just by transmitting the x and y coordinates. By this function we created the basic tool for a HUD.

3.3 control.cpp

This file is important for handling user input for moving a player and for shooting of a bullet. Generally by using the Windows-libraries [3] the keys have certain codes [4]. Conditionally checking if a certain key is pressed allows then changing for example a Boolean variable to “true”. With this information the drawing procedure may change the color, size or position of an object.

```
if((GetKeyState(VK_LEFT) & 0x80))
    {
        if(pl2.x > -map_size + cube_size + 0.5)
        {
            aux.x-=cube_size + 0.5;
            if(!player_collision(aux, map_size, cube_size))
                pl2.x = pl2.x - speed;
            aux.x+=cube_size - 0.5;
        }

    }
```

In this excerpt you can see the left movement of the left player. Firstly, there’s a check if the left key is pressed or not. Secondly, there are some conditions, checking if the tank collided with an obstacle or the map border. The term “collision” defines the meeting of two objects. Finally, the x-coordinate of the object is just changed by “speed”.

Every other key-handler has the same structure. Pressing the Q or ESC button quits the game and you will return to windows’ desktop.

3 Conclusion

Contrarily to our belief that building a game from scratch would be easy, now we understand that there is a specific structure we had to implement. Despite the fact that OpenGL has a very user-friendly and easy-to-understand documentation and a big community, designing a game starts by designing a way to get the players interest in that game.

Having a modular/object-oriented design – basically having created a so called engine – it allows creating other programs or games just by calling the predefined functions of the engine.

Another idea was to implement the gaming over the internet, so that players could duel without being in the same place. In addition, gamers high scores and acquired goals could be seen on Tank::Shooters website.

References

[1] http://www.opengl.org/
[2] http://nehe.gamedev.net/
[3] http://msdn.microsoft.com/en-us/library/windows/desktop/ms646301%28v=vs.85%29.aspx
[4] http://msdn.microsoft.com/en-us/library/windows/desktop/dd375731%28v=vs.85%29.aspx

Popa Adrian
ULBS
Stiinte - Informatica
Str. Dr.I.Raţiu, nr. 5-7
ROMANIA
ad_robotics@yahoo.com

Wandschneider Josef-Alexander
ULBS
Stiinte - Informatica
Str. Dr.I.Raţiu, nr. 5-7
ROMANIA
alexander91@versanet.de

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Denoising algorithms for processing images corrupted by additive noise

Nikola Rankovic
Teacher Coordinator: Milan Tuba

Abstract

One of the most important tasks in the field of image processing, itself a branch of the field of image restoration, is the removal of noise from an image. Noise may arise in the process of image acquisition, its transmission and also in the reproduction of an image. There are several approaches regarding the task of denoising, depending on the nature of the noise. Some image restoration techniques are best formulated in the spatial domain, and in this paper we will be analyzing the various algorithms used and will be comparing their effectiveness.

1 Introduction

Every image can be thought of as a representation or, even better, an approximation of an object, person or scene from the world at large, be it real or imaginary. When it comes to digital images this observation is even more important, since we are dealing with images that are acquired via some imaging apparatus or sensor, stored on a digital medium and are very likely transmitted across networks. At each of these steps noise can be acquired, though the principal sources of noise in digital images arise during image acquisition (digitalization) and/or transmission [1].

The field of denoising is still very active today, with research involving many local and non-local algorithms (in terms of the processed images), working on both spatial and frequency domains. There are also machine learning algorithms that are effectively used in this process. Image denoising still remains a challenge for researchers because noise removal introduces artefacts and causes blurring of the images [2].

The applications of imaging denoising are many. For instance, many algorithms developed for tasks in computer vision, such as object recognition, segmentation and others, assume that the input images contain little or no noise [3]. In medicine for example, the fundamental problem of ultrasound images is the poor quality, mainly caused by multiscale speckle noise [4].

In this paper we will be providing an overview of three most common noise models that are used to simulate image degradation (Gaussian, uniform and impulse noise) and also the various local algorithms, including an example of an adaptive algorithm, that tackle the process of denoising an image.

2 Software

The software that has been developed alongside this paper allows experiments with images and provides implementation of the various algorithms there mentioned. The application was developed in the C# language, as the language is easy to understand and provides fast prototyping and development speeds.

The image data is handled with custom functions and classes that operate on the low-level (as it would be done in a language such as C for instance), without the use of high-level libraries that are available, to better illustrate the algorithms that are implemented and to provide better portability and higher speed.

Among the software features are some basic image manipulation functions, adding of different types of noise to an image and of course, the implementations of different denoising algorithms. There are also some example images included, on which the various experiments were undertaken in this paper, that were publicly available from the Digital Image Processing website: www.prenhall.com/gonzalezwoods and from Wikipedia, available under the creative commons licence (Unequalized Hawkes Bay NZ.jpg by Phillip Capper).

3 Image representation

Representing images in the spatial domain means keeping record of each pixel individually, while keeping information about each one of its components, according to the used colour model - for instance RGB (which we implemented in our software), HSV and others. More formally, an image may be defined as a two-dimensional function [1] f(x,y) where the x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x,y) is called the intensity or gray level of the image at that point. When x, y and the amplitude values of f are all finite, discrete quantities, we call the image a digital image.

Images can also be represented in the frequency domain, using a Fourier transform. This is based on the fact that any function that periodically repeats itself can be expressed as the sum of sines and/or cosines of different frequencies, each multiplied by a different coefficient (this sum is called a Fourier series). We can further convert an image from its spatial representation to the frequency domain and back, with no loss of information. The frequency domain is nothing more than the space defined by values of the Fourier transform and its frequency variables (u,v). In computer graphics, as well as electronics, statistics and many other disciplines, frequency domain is a term used to describe the domain for analysis of mathematical functions or signals with respect to frequency rather than time [5].

4 Noise

Image noise is random variation of brightness or colour information in images. It is an undesirable by-product of image capture that adds spurious and extraneous information.

The noise embedded in an image manifests in diverse varieties. The noise may be correlated or uncorrelated; it may be signal dependent or independent, and so on. The knowledge about the imaging system and the visual perception of the image helps in generating the noise model. Estimating the statistical characteristics of noise embedded in an image is important because it helps in separating the noise from the useful image signal [6].

There are various techniques that are used in the process of image restoration which attempt to reconstruct or recover and image that has been degraded by using a priori knowledge of the degradation phenomenon [1] and in this paper we will be dealing only with degradations due to noise.

We can represent a degraded image in the spatial domain with the following equation:

$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y) \tag{1}$$

where *h(x,y)* is the spatial representation of the degradation function, *η(x,y)* is an additive noise term and the ‘*’ symbol indicates convolution. Since convolution in the spatial domain is equal to multiplication in the frequency domain, we may write the model in an equivalent frequency domain representation:

$$G(u,v) = H(u,v)F(u,v) + N(u,v) \tag{2}$$

where the terms in capital letters are the Fourier transforms of the corresponding terms in the previous equation for the spatial domain.

In this paper we will be only concerned with case when H is an identity operator and will be dealing solely with degradations caused by noise.

4.1 Noise models

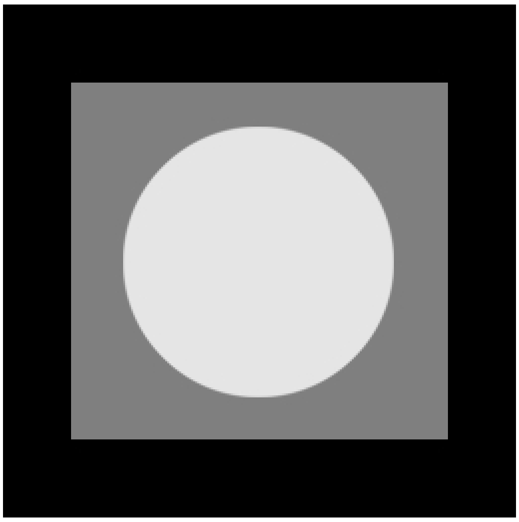


Figure 1 *Test image used for testing out different noise models.*

Various factors are involved in the generation of noise in the image acquisition process, including environmental factors and the properties of the sensing elements used. During transmission, noise can occur due to interference in the transmission channel.

We assume that noise is independent of spatial coordinates and that there is no correlation between pixel value of the image and the value of noise components. We will also be concerned with the spatial noise descriptor, i.e. the statistical behaviour of the gray-level values in the noise component of the model, as characterized by a probability density function (PDF) or random variables. The following are some important PDFs regarding image processing that are covered in this paper and implemented in the supplied software.

4.1.1 Gaussian noise

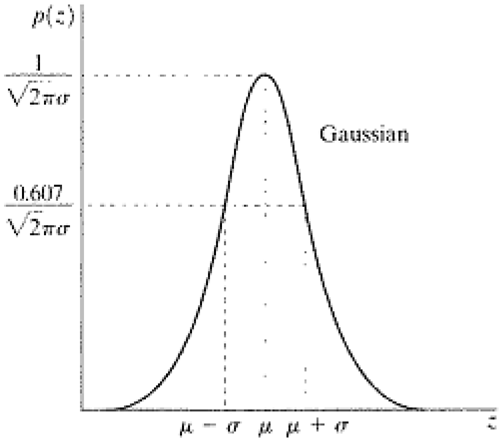


Figure 2 *Gaussian noise probability density function*

Gaussian (or normal) noise models are frequently used in practice, due to their mathematical tractability in both domains.

The PDF of a Gaussian random variable z is given by:

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{(z-\mu)^2/2\sigma^2} \tag{3}$$

where z represents gray level, η is the mean average value of z, and σ is its standard deviation.

Gaussian noise arises in an image due to factors such as electronic circuit noise and sensor noise due to poor illumination and/or high temperature.

4.1.2 Uniform noise

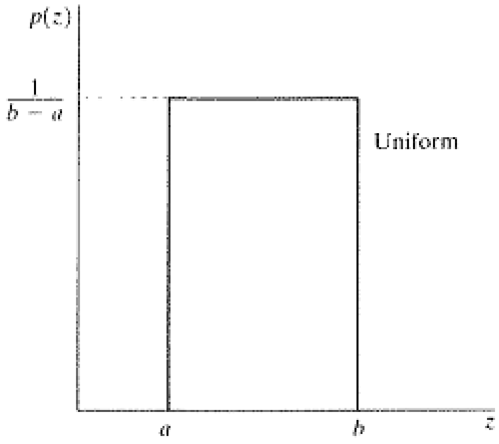


Figure 3 *Uniform noise probability density function*

The PDF of exponential noise is given by:

$$p(z) = \begin{cases} \frac{1}{b-a}, & a \leq z \leq b \\ 0, & \text{else} \end{cases} \tag{4}$$

where a and $b > 0$. The mean is given by

$$\mu = \frac{a+b}{2} \tag{5}$$

and its variance by:

$$\sigma^2 = \frac{(b-a)^2}{12} \tag{6}$$

The uniform density is perhaps the least descriptive of practical situations. However, the uniform density is quite useful as the basis for numerous random number generators that are used in simulations.

4.1.3 Impulse noise (salt-and-pepper)

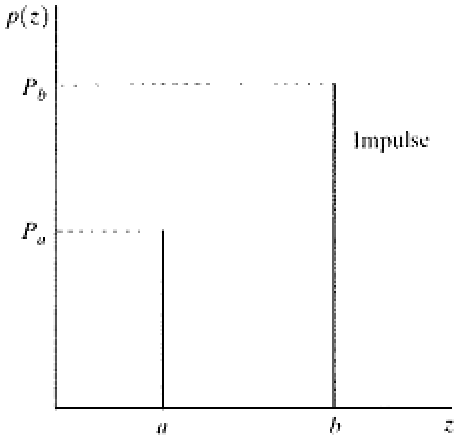


Figure 4 *Impulse (salt-and-pepper) noise probability density function*

The PDF of (bipolar) impulse noise is given by:

$$p(z) = \begin{cases} P_a, & z = a \\ P_b, & z = b \\ 0, & \text{else} \end{cases} \tag{7}$$

If $b > a$, gray-level b will appear as a light dot on the image. Otherwise, level a will appear like a dark dot. If either *Pa* or *Pb* is zero, the impulse noise is called unipolar. If neither probability is zero (especially if they are approximately equal), impulse noise value will resemble salt-and-pepper granules randomly distributed over the images (hence the name).

Noise impulses can be negative or positive and impulse noise is generally digitized as extreme (pure black or white) values in an image.

Impulse noise is found in situations where quick transients, such as faulty switching, take place during imaging.

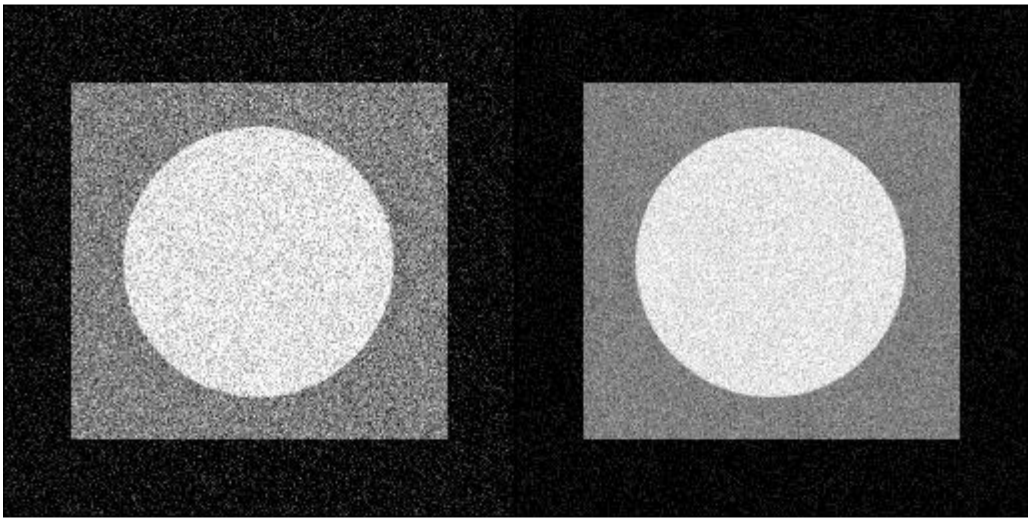


Figure 5 *Results of our software implementation of Gaussian and uniform noise models*

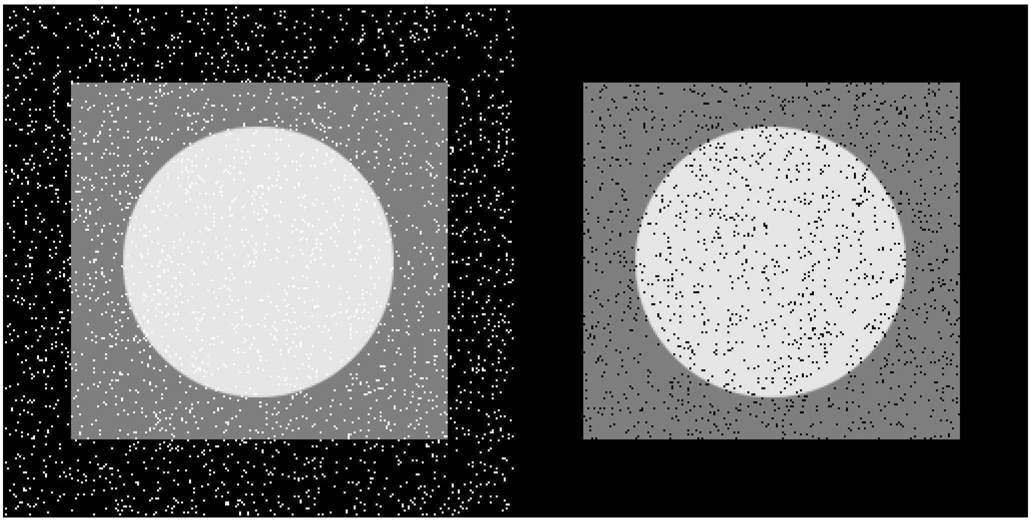


Figure 6 *Results of our software implementation of salt and pepper noise models*

4.1.4 Periodic noise

Periodic noise in an image typically arises from (electrical) interference during image acquisition. It is spatially-dependent and can be reduced significantly by filtering in the frequency domain.

5 Denoising overview

The parameters of a noise PDF can often be estimated from the arrangements of a particular image.

Often, it is possible to use small areas of the image, with a constant gray level, to estimate the parameters of the PDF. By analyzing the histogram shapes of these regions, we may identify the closest PDF match. We use the mean and variance for estimating the parameters of Gaussian and uniform noise models. Impulse noise parameters are estimated by selecting a small patch of the image, of constant mid-grey colour, and retrieving the probability of occurrence of black and white pixels.

6 Denoising algorithms

Various factors are involved in the generation of noise in the image acquisition process, including environmental factors and the properties of the sensing elements used. During transmission, noise can occur due to interference in the transmission channel.

We assume that noise is independent of spatial coordinates and that there is no correlation between pixel value of the image and the value of noise components. We will also be concerned with the spatial noise descriptor, i.e. the statistical behaviour of the gray-level values in the noise component of the model, as characterized by a probability density function (PDF) or random variables. The following are some important PDFs regarding image processing that are covered in this paper and implemented in the supplied software.

6.1 Spatial filtering

Spatial filtering is the method of choice in situations when only additive noise is present [1].

6.1.1 Arithmetic mean filter

The simplest of mean filters, the arithmetic mean filtering process computes the average value of the corrupted image *g(x,y)* in the area defined by *S_xy* (a rectangular subimage window of size *m*×*n*, centred at the point *(x,y)*.

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(x,y)\in S_{xy}} g(x,y) \tag{8}$$

Noise is reduced as a result of blurring.

6.1.2 Geometric mean filter

The geometric mean filter uses the geometrical mean as a basis of blurring, to a similar effect of the arithmetic mean filter, but with a tendency to lose less image detail in the process.

$$\hat{f}(x,y) = \left[\prod_{(x,y)\in S_{xy}} g(x,y) \right]^{\frac{1}{mn}} \tag{9}$$

The following are the results of using our application to apply a Gaussian filter and then filtering it with an arithmetic and then geometric filter, with *S_xy* of dimension *3x3*.

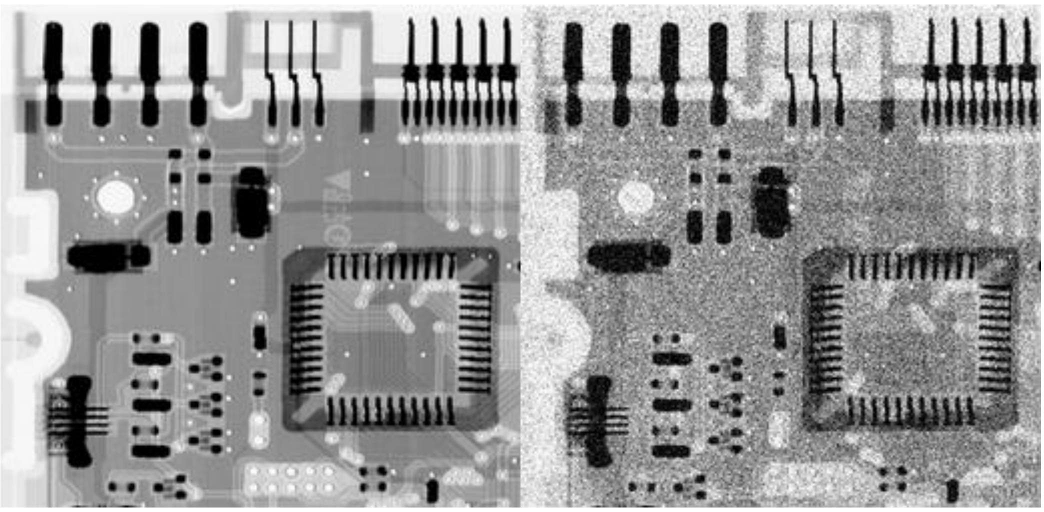


Figure 7 *The original X-ray image of an electronic circuit (left) and the result of adding Gaussian noise (right).*

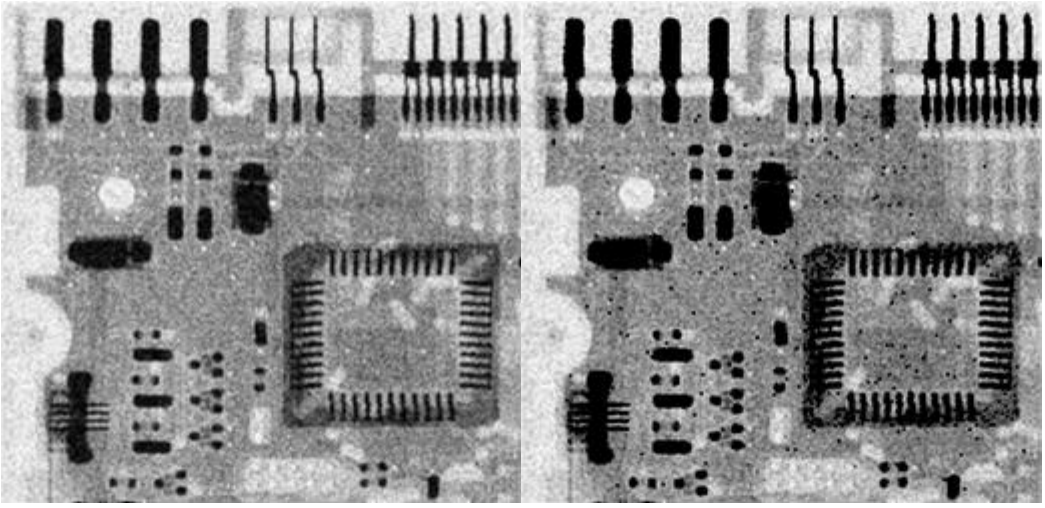


Figure 8 *The results of applying the arithmetic (left) and geometric (right) filters on the image corrupted by Gaussian noise.*

6.1.3 Harmonic mean filter

The harmonic mean filter works well for salt (and other, like Gaussian) noise, but fails for pepper noise. It is defined by the expression:

$$\hat{f}(x,y) = \frac{mn}{\sum_{(x,y)\in S_{xy}} \frac{1}{g(x,y)}} \tag{10}$$

6.1.4 Contraharmonic mean filters

Where Q is the order of the filter, the contraharmonic filter is defined by the expression:

$$\hat{f}(x,y) = \frac{\sum_{(x,y)\in S_{xy}} g(x,y)^{Q+1}}{\sum_{(x,y)\in S_{xy}} g(x,y)^{Q}} \tag{11}$$

It is well suited for eliminating salt noise (when *Q* is negative) or pepper noise (when *Q* is positive).

The following are the results of applying salt noise on the electronic circuit image, and then filtering it with a harmonic filter and contraharmonic filters (for *Q* = 1.5 and *Q* = -1.5).

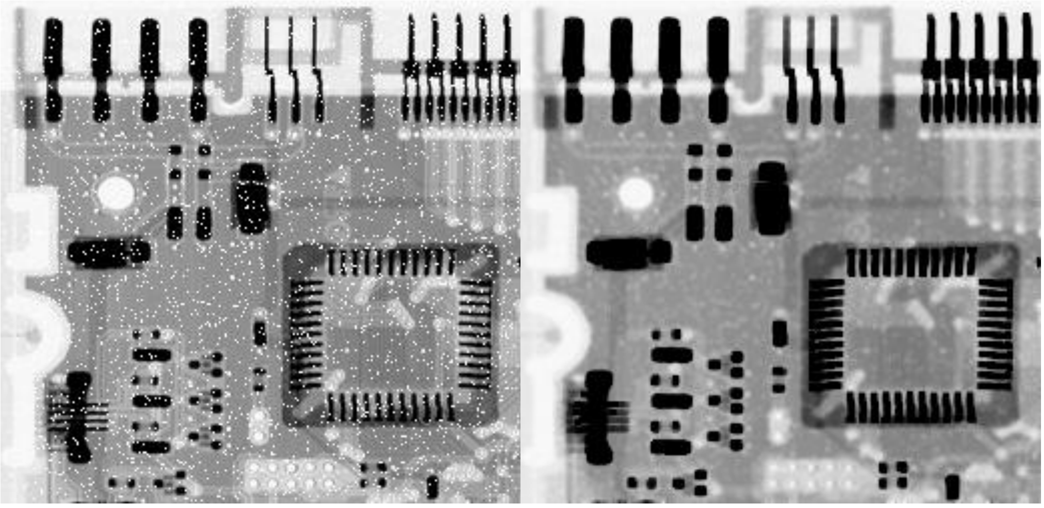


Figure 9 *Addition of salt noise (left) and the result of a harmonic filter (right)*

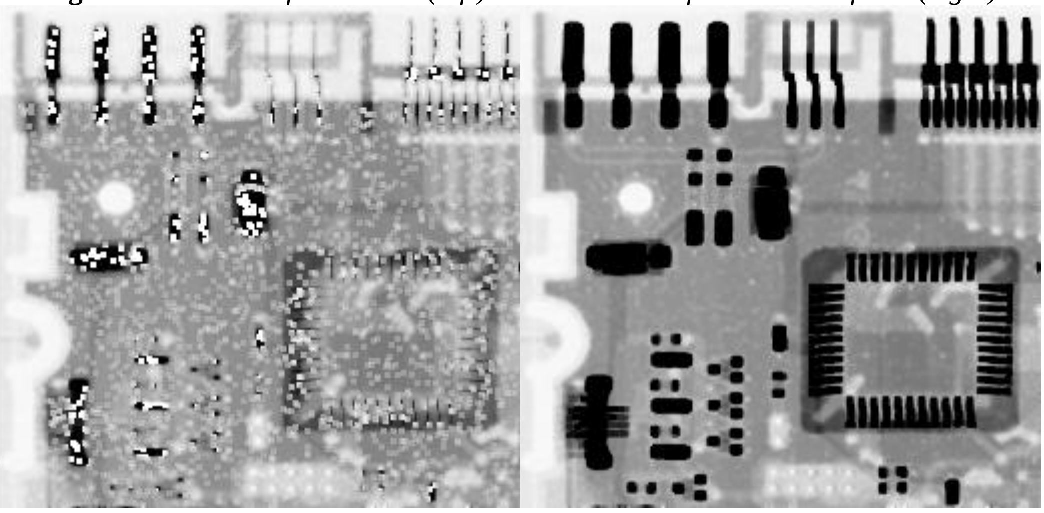


Figure 10 *The results of a positive Q = 1.5 (left) and negative Q = -1.5 (right) contraharmonic filters on the image degraded by salt noise.*

As can be seen, harmonic and a negative *Q* contraharmonic filter both work well with salt noise.

The following are the results of applying pepper noise on the electronic circuit image, and then filtering it with a harmonic filter and contraharmoic filters (for *Q* = 1.5 and *Q* = -1.5).

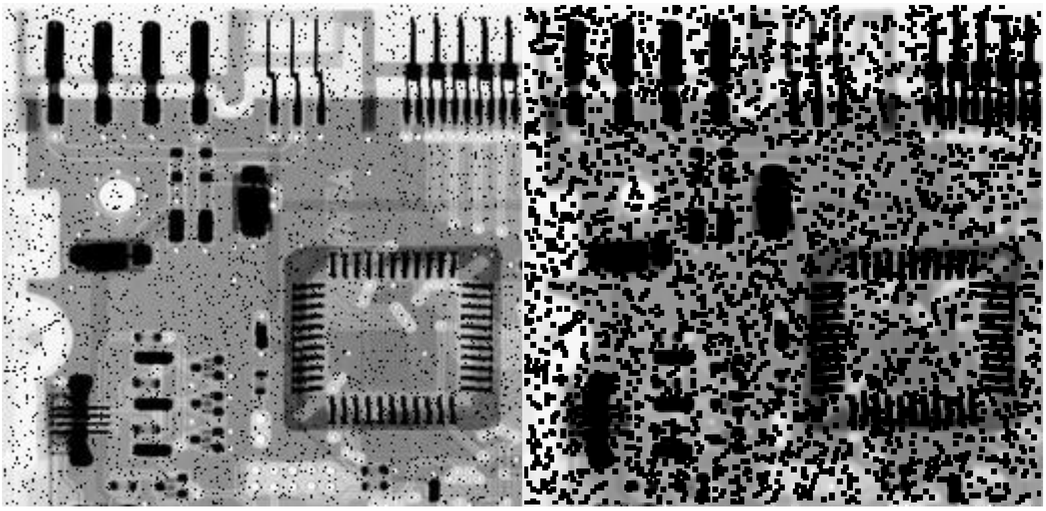


Figure 11 *Addition of pepper noise (left) and the result of a harmonic filter (right)*

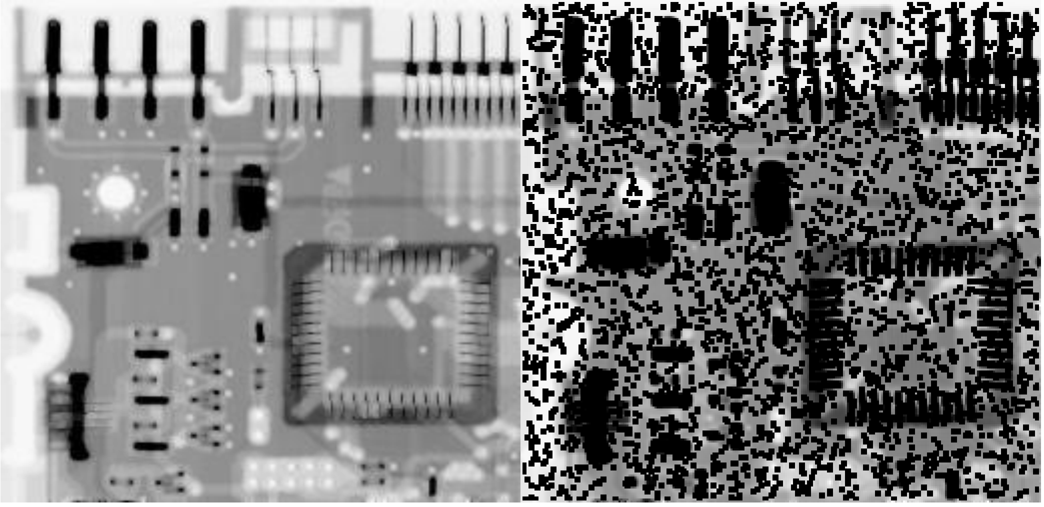


Figure 12 *The results of a positive Q = 1.5 (left) and negative Q = -1.5 (right) contraharmonic filters on the image degraded by salt noise.*

For pepper noise, we find that only a positive *Q* contraharmonic filter worked well.

6.1.5 Median filter

This is the best-known order-statistics filter, which replaces the value of a pixel by the median of the grey levels in the neighbourhood of that pixel.

For certain types of noise, these filters provide excellent noise-reduction, and are particularly good at dealing with both unipolar and bipolar impulse noise.

The following is the circuit image corrupted by both salt and pepper noise, and the median filter applied to it.

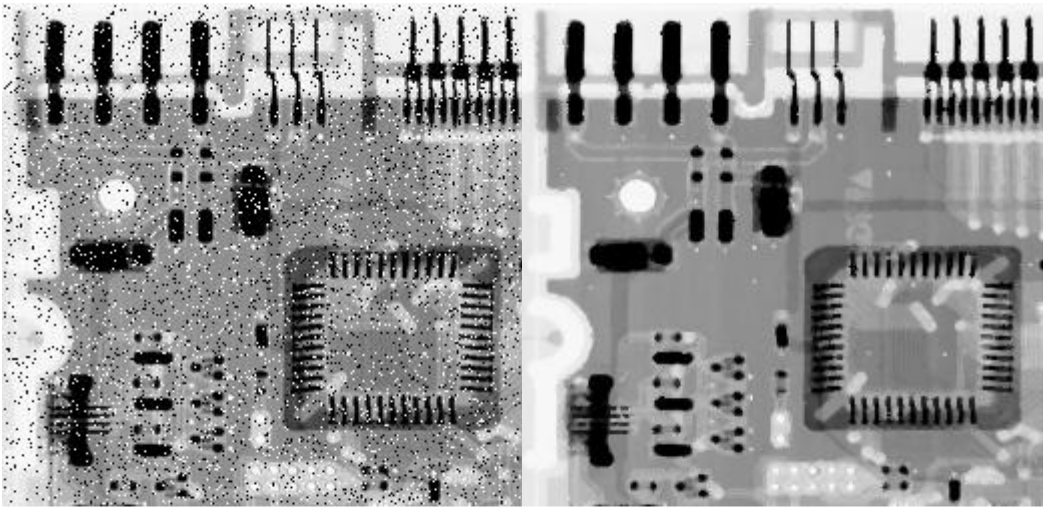


Figure 13 *The results of applying both salt and pepper noise (left) and the median filter (right).*

As can be seen, the median filter was successful in dealing with both salt and pepper noise.

6.1.6 Max and min filters

These are also order-statistics filter, useful at identifying the brightest and the darkest points of an image, by replacing a pixel's value by the maximum (or minimum) value in its neighbourhood. Max filter can help reduce pepper noise, while min filter reduces salt noise.

The following are the results of filtering pepper noise with a max filter and salt noise with a min filter.

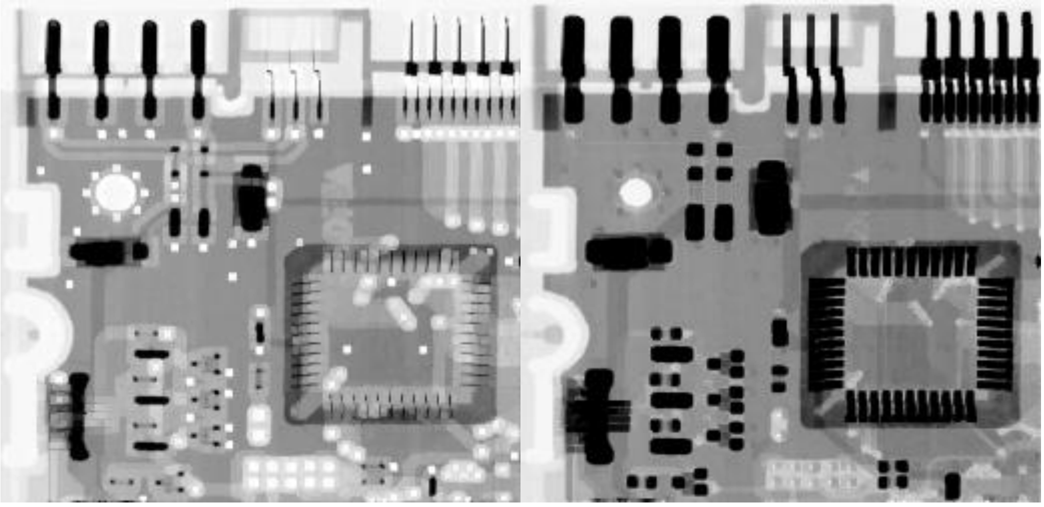


Figure 13 *The results of applying the max filter on the pepper-noise corrupted image (left) and the min filter on the salt-noise corrupted image (right).*

The max and min filters were successful in removing the corresponding noises, with a noticeable loss of quality and details of the image.

6.1.7 Midpoint filter

A combination of the max and min filters, the midpoint filters work best for randomly distributed noise, such as Gaussian or uniform noise.

6.2 Adaptive filters

An adaptive filter's behavior changes based on statistical characteristics of the image inside the filter region, as defined by the *mxn* region *S_xy*. These filters are of a greater complexity and analyse how image characteristics vary from one point to another.

6.2.1 Adaptive median filter

The adaptive median filter preserves detail while smoothing impulse noise. It changes the size of the working window *S_xy* during execution, according to specified conditions. First, we define the following:

$$\begin{gathered} z_{min} = \text{minimum grey level value in region } S_{xy} \\ z_{max} = \text{maximum grey level value in region } S_{xy} \\ z_{med} = \text{median of grey levels in region } S_{xy} \\ z_{xy} = \text{grey level at coordinates } (x, y) \\ S_{max} = \text{maximum allowed size of } S_{xy} \end{gathered} \tag{12}$$

The adaptive median filtering algorithm can be represented with the following pseudo-code:

```
A1 = z_med - z_min
A2 = z_med - z_max
if A1 > 0 and A2 < 0
        B1 = z_xy - z_min
        B2 = z_xy - z_max
        if B1 > 0 and B2 < 0
                return z_xy
        else
                return z_med
else
        increase window_size
if window_size <= S_max
        continue
else
        return z_xy
```

The values of *z_min* and *z_max* are considered by the algorithm to be impulse-like noise components. If the median of the working region falls between these two values, then we do not consider it an impulse. If so, then we check whether the point in the centre of the region is itself an impulse, by seeing whether *z_xy* is between the minimum and maximum value. If this condition holds, then the algorithm returns the original value of the pixel. If not, then the value of the pixel is extreme and the algorithm outputs the median value.

If we had previous found an impulse (for the median of the working region), we increase the size of the window and restart the algorithm from the beginning.

For smaller noise probabilities or larger maximum working region size *S_max*, it is less likely that we will exit the algorithm prematurely. The choice of the maximum value can be estimated by experiment with the "standard" median filter first.

Every time the algorithm returns a value, the window is moved to the next location in the image and the algorithm is reinitialized, working with the pixels in the new location.

The following are the results of applying noise with the uniform distribution and the application of the adaptive median filter.

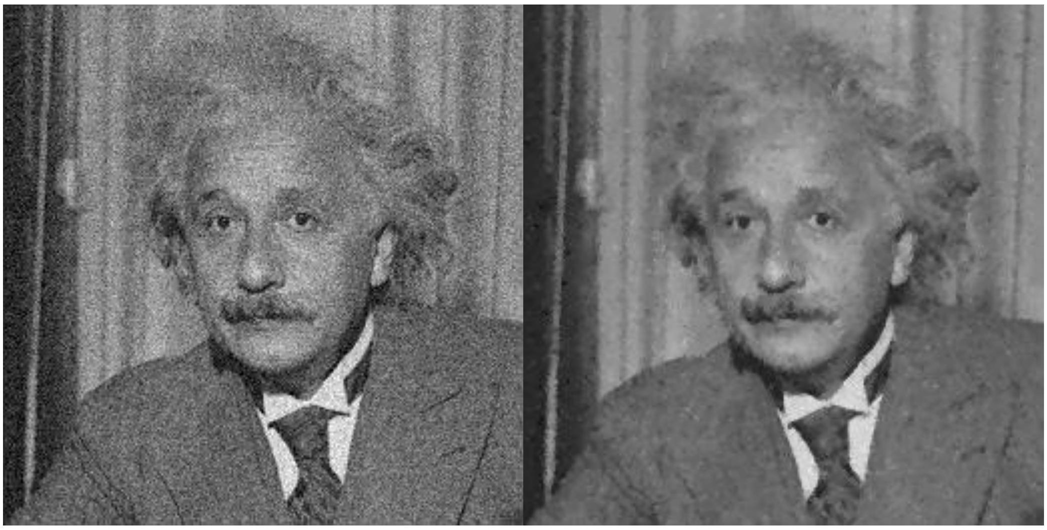


Figure 14 *The results of applying the noise with uniform distribution (left) and the subsequent applying of the adaptive median filter (right).*

7 Summary

In this paper, we worked with the assumption that image degradation can be modelled as a linear, position-invariant process with the addition of (additive) noise that is not correlated with image values. We can obtain useful results by simulating various types of additive noise and applying the various filters (working on the spatial domain) that were provided in the previous sections. Each filter works best for certain types of noise and performs not so well on others. An observer's preferences and capabilities must be taken into account since the denoising process has to rely on subjective interpretations of an individual image's "enhancement" or "restoration". Therefore, the area of denoising remains a challenge for further improvements in the field.

References

[1] R.C. Gonzalez and R. E. Woods. *Digital Image Processing (2nd Edition)*. Pearson Prentice Hall, New Jersey, 2002.

[2] M. C. Motwani, M. C. Gadiya, Rakhi C. Motwani and F. C. Harris. *Survey of Image Denoising Techniques*. Proceedings of GSP 2004, Santa Clara (CA), 2004.

[3] D. Gong, F. Sha and G. Medioni. *Locally linear denoising on image manifolds*. Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, Italy, 2010.

[4] Y. S. Kim and J. B. Ra. *Improvement of Ultrasound Image Based on Wavelet Transform: Speckle Reduction and Edge Enhancement*. Proc. of SPIE Vol. 5747. Korea, 2005.

[5] S.A. Broughton and K. Brian. Discrete *Fourier Analysis and Wavelets: Applications to Signal and Image Processing*. Wiley, New York, 2008.

[6] T. Acharya and A.K. Ray. *Image Processing: Principles and Applications*. Wiley, New Jersey, 2005.

Nikola Rankovic
University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: nikola.s.rankovic@gmail.com

Prof. Milan Tuba
University Megatrend of Belgrade
Faculty of Computer Science
Bulevar Umetnosti 29, Belgrade
SERBIA
E-mail: tuba@ieee.org

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Image rendering using parallel ray tracing algorithm

Filip Slavov
Teacher Coordinator: Associate Professor, PhD Milena Lazarova

Abstract

The paper describes the development of parallel implementation of ray tracing algorithms for scene visualization. A software system based on backward ray tracing rendering algorithm comprising several modules is designed and developed. The system architecture and its main modules are presented. The parallel computational models for shared memory and distributed memory platform and their parallel programming implementation using message passing (MPI) and multithreading (OpenMP) are discussed. The performance benchmark of the parallel image rendering using the developed ray tracing system are evaluated using three test scenes. The experimental results show very good scalability of the parallel image rendering both in view of the workload and the hardware platform.

1 Introduction

Ray tracing is a rendering technique, used to create high quality images and videos of a given 3D virtual scene. It is based on tracing the light rays travelling in space that reflects and refracts with objects. Simulating the natural way of how light is spreading leads to more realistic images, but also leads to a lot more heavy calculations.

In this method for image synthesis separate light rays are traced between any light source and the viewer, including rays intersections with scene objects [1]. Among the advantages of this rendering approach, is its straightforward parallel computation. The tracing of each ray can be done concurrently by separate treads as the scene remains unmodified. This means that the heavy calculations can be easily divided among multiple workers, taking a lot less time than sequential tracing.

The efforts for speeding up the image rendering using ray tracing algorithm are aimed at utilization of the existing high-performance computer platforms with different architecture. Both distributed memory and shared memory parallel implementations are suggested using several parallel programming models and interfaces [4, 5, 6]. Recent attempts for real-time rendering based on ray tracing are aimed at the utilization of GPU processing power based on nVidia CUDA architecture [7, 8, 9]. A prototype implementation of a programmable chip Ray Processing Unit (RPU) is presented in [10].

The paper describes the development of parallel implementation of ray tracing algorithms for scene visualization. A software system based on backward ray tracing rendering algorithm comprising several modules is designed and developed. The system architecture and its main modules are presented.

The parallel computational models for shared memory and distributed memory platform and their parallel programming implementation using message passing (MPI) and multithreading (OpenMP) are discussed. Some experimental results evaluating the performance benchmark of the parallel image rendering using the developed ray tracing system are given.

2 System architecture

2.1 Ray tracing algorithm

2.1.1 Types of ray tracing algorithms

According to the ray tracing algorithm, light rays should be traced from any light source until they exit the scene or hit the “eye” of the viewer. Two stages of ray tracing are usually adopted – forward ray tracing (Fig. 1) and backward ray tracing (Fig. 2). If a ray is being traced from a point light, which generates light rays in every direction, the recursively reflection and refraction of the ray by the scene objects makes the calculation task of ray tracing extremely difficult. Even not taking into account the complex mathematical calculations of ray and objects interactions most of the rays emitted by the light source won’t hit the viewer screen and therefore won’t be visible at all. That’s why the backward ray tracing is applied: limited number of rays is generated from the viewer point, through each pixel of the screen and these rays are followed to check if the hit any light and which objects they intersect.

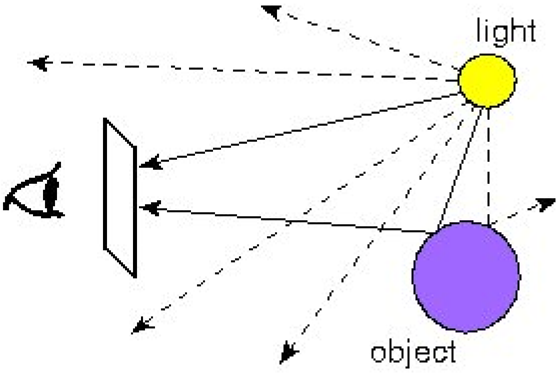


Fig. 1. Forward ray tracing: physics based, rays start from the light sources

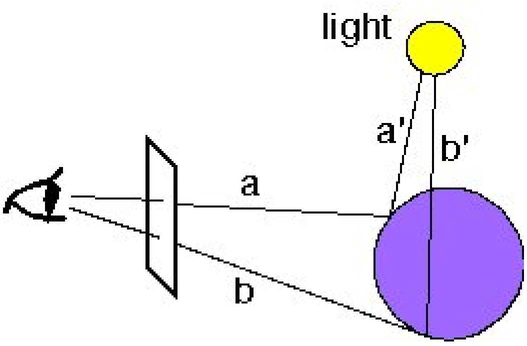


Fig. 2. Backward ray tracing: source of the rays is the “eye” of the viewer.

2.1.2 Types of rays in backward ray tracing

There are four types of rays that are being traced using the backward ray tracing (Fig. 3):

- Eye ray – primary ray, starting from viewer position through each pixel of the screen. The result returned by this ray is stored as the new color of the corresponding pixel.

It is traced to the first hit object, or outside the scene, where the background color is used for return value.

- Reflected ray – secondary ray, spawned recursively from another ray intersecting given object surface. Its direction is calculated according to the angle between the source ray and the surface.
- Transmitted (refracted) ray – secondary ray, spawned again recursively from another ray, below the surface of an object, representing refracted rays. Their direction is result of the snell's law for refraction.
- Shadow ray – at each intersection point of a given ray with an object, shadow rays are cast to every light source, to test if this point is lit directly, and if so, accumulate its influence to the total result.

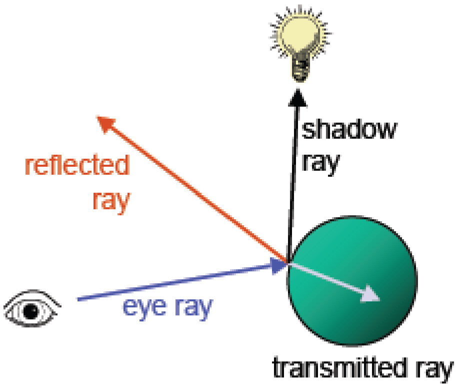


Fig. 3. Types of rays

2.2 System modules

The developed ray tracing rendering software system has 3 base modules shown in Fig. 4:

- RayTracing – static representation of the current state of the scene to be rendered.

The ray tracing algorithm works directly with this module.

- Engine – controls time and logic in the scene.

It contains the RayTracing module and modifies it according to objects logic in the world.

- Main program – initializes the engine and keeps synchronized multiple engines being executed on different machines, to divide the work for rendering.

These modules work together in a hierarchical manner, as higher level modules control lower level modules. Each of them exposes an abstract interface to be used with. They can work on their own, without knowing who is using them.

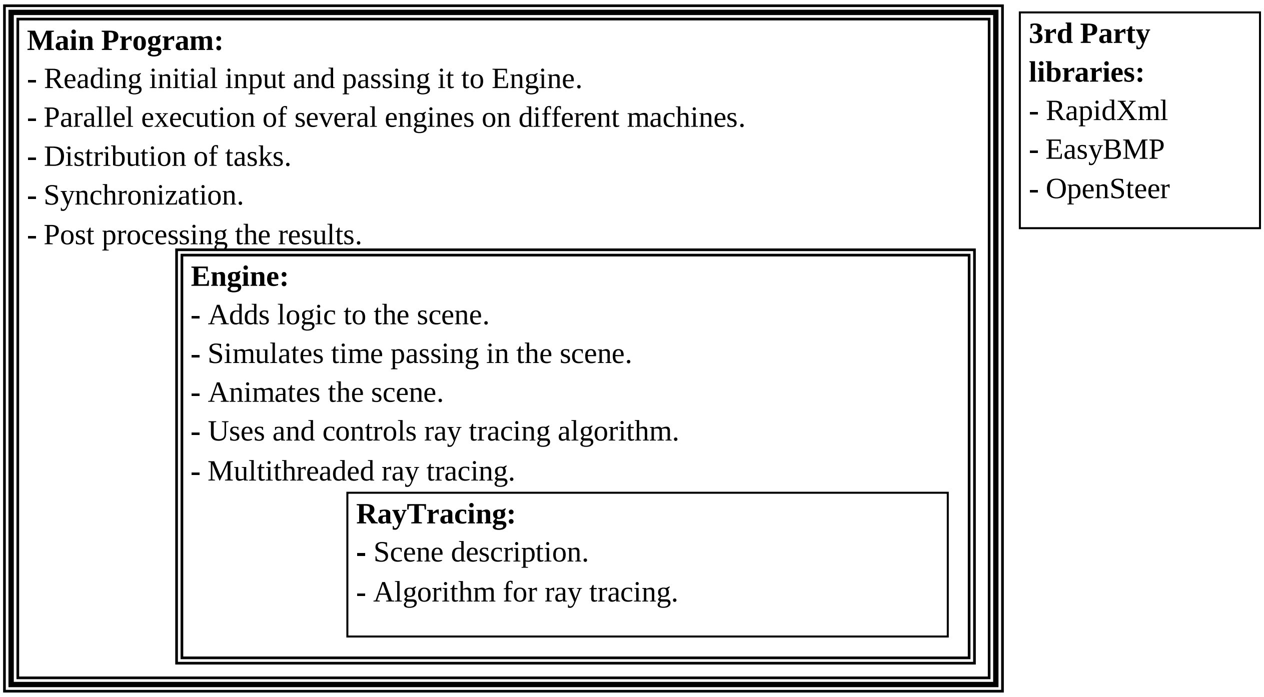
Additionally, there are number of third party libraries used by all of the modules like RapidXml, EasyBMP, OpenSteer.

2.2.1 RayTracing module

Scene description of the world includes objects, cameras and lights. It uses tree representation, where each node state is relative to the parent. This allows building of complicated objects, made out of many nodes with possibility to execute own logic on each node. Some examples are shown on Fig. 5 and Fig. 6. If the master parent node "Chassis" is moved or rotated in 3D space, all sub-nodes will be rotated and moved accordingly.

Cameras and light sources are also represented as tree nodes. This allows them to be attached to some object and move together with it accordingly. For example, light source can be attached as a head-light in front of the car node or camera can be attached over the driver's seat.

The ray tracing algorithm works with objects that are either defined as a primitive sphere or a mesh. The system can import wavefront .obj files. The rendering algorithm uses the backward ray tracing method.



Main Program:
- Reading initial input and passing it to Engine.
- Parallel execution of several engines on different machines.
- Distribution of tasks.
- Synchronization.
- Post processing the results.

Engine:
- Adds logic to the scene.
- Simulates time passing in the scene.
- Animates the scene.
- Uses and controls ray tracing algorithm.
- Multithreaded ray tracing.

RayTracing:
- Scene description.
- Algorithm for ray tracing.

3rd Party libraries:
- RapidXml
- EasyBMP
- OpenSteer

Fig. 4. System modules: higher levels control lower levels

.

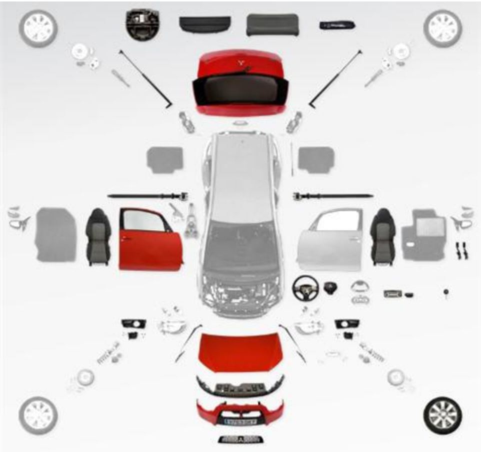


Fig. 5. Car decomposes to many different parts

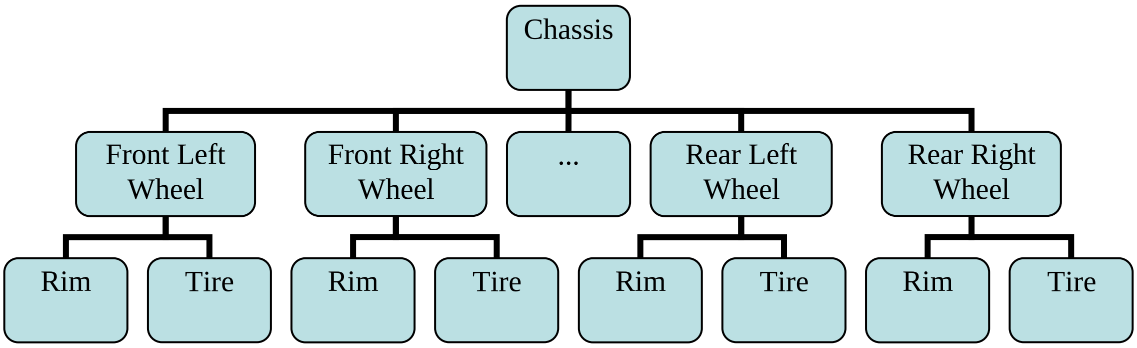


Fig. 6. Shows tree like representation of car parts

2.2.2 Engine module

As shown above, the RayTracing module contains just a static state of the scene. This is enough as a requirement for image rendering, but if the goal is to make a video sequence then some additional logic have to be added to the world. The purpose of the Engine module is to progress the world with time passing. It contains number of logical object entries that are attached to a given scene node. This object modifies the state of the controlled node, according to time elapsed. This provides easy way to generate moving objects in the scene, to detect collision, to define user controlled objects, etc.

3 Parallel computational models for image rendering

The parallel computational model for image rendering is based on two paradigms: manager – worker and divide-and-conquer. Screen pixels are divided into work chunks thus creating several independent tasks that are distributed for processing by the available hardware resource. Each worker (thread or process) is responsible for all the calculation required for tracing the rays through certain part of the image, i.e. several pixels (Fig. 7).

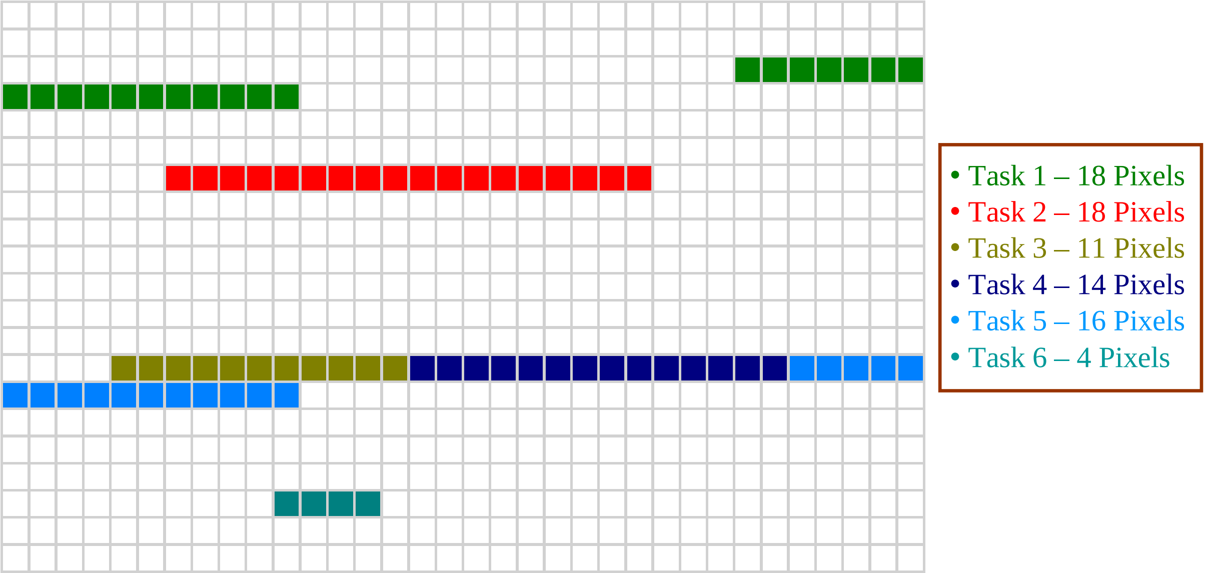


Fig. 7. Distributing screen pixels into work chunks that correspond to the tasks to be executed

3.1 Parallel ray tracing using multithreading

For the multithreading parallel programming model the OpenMP standard is used. It is implemented in the RayTracing module, where the algorithm divides the given work chunk into sub-chunks and executes them in different threads. A simple fork – join job is used for the parallel ray tracing model – image rendering is completed when all sub-tasks are finished and return the results (Fig. 8).

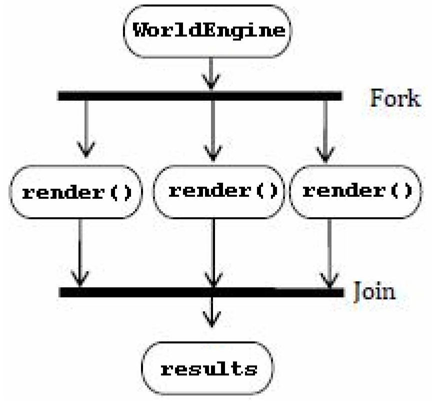


Fig. 8. Fork – join paradigm working on different chunk of pixels from the screen.

3.2 Parallel ray tracing using message passing

For the message passing programming model of the ray tracing the standard MPI is utilized. The parallel computations are carried out by distributing the work between several nodes in a cluster.

Master-slave paradigm is used and the required MPI function calls for communication and synchronization between the manager process and the worker processes are implemented in the "Main Program" module, whose job is to load the scene on each computer. One of the processes is a manager process that is responsible to divide the work between the available computational nodes. The master process also collects the result chunks from the slaves and assembles the final image after each rendered frame.

As shown on Fig.9 the developed parallel ray tracing system uses hybrid programming model that is aimed to fully utilize the computational resources of computer cluster comprising multi core nodes – each of the processes uses multithreaded processing of its image workload.

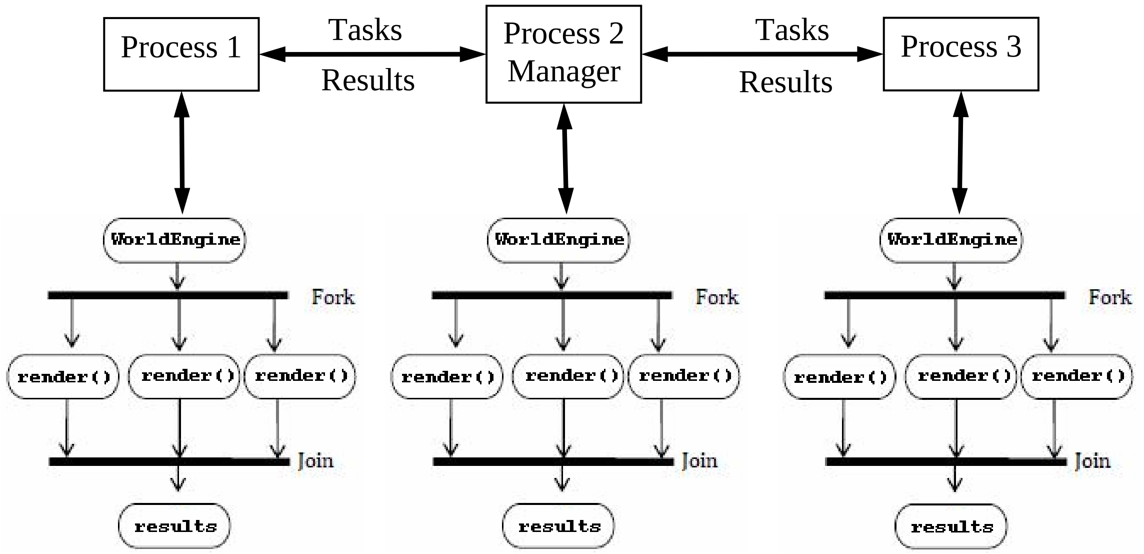


Fig. 9. Master – slave paradigm working on bigger chunks of pixels from the screen

4 Experimental results

The experimental platform used for evaluation of the performance benchmark of the developed parallel rendering system is a heterogeneous compact cluster with 10 nodes – eight AMD Opteron 64 DualCore Processors @ 2 GHz and two Intel Xeon 8 core Processors @ 2.2 GHz. Two different scenes with 74 718 and 53 868 triangles were used. Additional scene with only 27 triangles and mirror surfaces was used for experiments with 12 levels of recursion of the algorithm.

The results for the speedup for the multithreaded (OpenMP) parallel programming model for each of the three test scenes are shown on Fig. 10. The comparison of the multithreaded (OpenMP) vs. message passing (MPI) image rendering of Scene3 are given on Fig. 11.

The experimental results show very good scalability of the parallel image rendering both in view of the workload and the hardware platform. For the OpenMP model linear speedup is achieved for the Scene 2. Slightly lower values of speedup of processing of Scene 3 can be explained if the synchronization overhead is taken into account – the Scene 3 rendering time was measured for generation of 360 frames. Almost linear scalability was also achieved with respect to the number of processes/threads that was utilized. As seen on Fig.11 the processing is faster for the shared memory parallel programming model compared to the distributed memory message passing model due to the higher latency of the network communication.

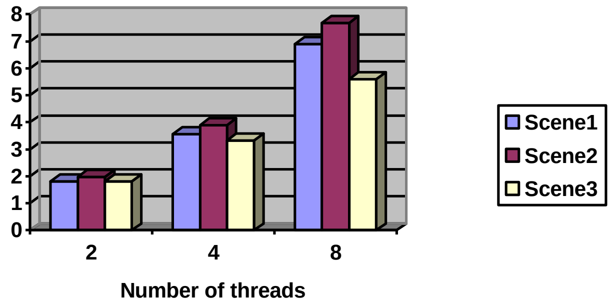


Fig. 10. Speedup of the multithreaded (OpenMP) parallel programming model

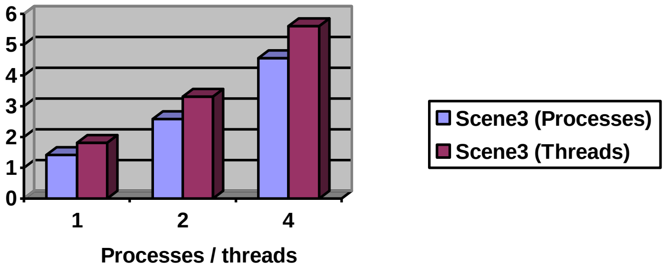


Fig.11. Speedup comparison of multithreaded vs. message passing image rendering of Scene3

5 Conclusion

A software system for image rendering based on backward ray tracing algorithm was designed and developed. The system comprises several modules that provide all the required functionality, and implement the ray tracing algorithm and scene description (RayTracing module), the implementation of logic and ability to create animations (Engine module). In order to speedup the image rendering multithreaded and message passing programming models are suggested and implemented. Thus the developed system provides possibility for speeding up the calculations of the ray tracing algorithm. The good scalability of the performance of the system makes it possible to use it for image and video rendering as well as for remote visualization.

Further development of the system will be aimed at adding a load balancing of the workload of the parallel computational model that will additionally improve the performance especially in the case of heterogeneous clusters. In order to provide larger scaling of the parallel rendering software to many core machines, parallel input and output should be also implemented that will minimize the communication overhead and increase the processing speed allowing real time image and video rendering.

References

[1] A.S.Glassner, *An Introduction to Ray Tracing*, Academic Press, 1989.

[2] Samuel R. Buss, *3-D Computer Graphics - A Mathematical Introduction With OpenGL*, Cambridge University Press, 2003.

[3] Radermacher, P. Ray Tracing: Graphics for the Masses, http://www.cs.unc.edu/~rauchnach/xroads-RT/RTarticle.html

[4] Kadir S., T. Khan, Parallel Ray Tracing using MPI and OpenMP, Project Report, Introduction to High Performance Computing, Royal Institute of Technology, Stockholm, Sweden, 2008.

[5] Bigler J., A. Stephens, S. Parker, Design for Parallel Interactive Ray Tracing Systems, Proc, of IEEE Symposium on Interactive Ray Tracing, Salt Lake City, UT, 18-20 Sept. 2006, pp.187–196.

[6] Wald I., C. Benthin, A. Dietrich, P. Slusallek, Interactive Ray Tracing on Commodity PC Clusters: State of the Art and Practical Applications, Proc. of Euro-Par Conf. for Parallel Processing, Lecture Notes in Computer Science, 2003, Vol. 2790, 2003, pp.499-508.

[7] Marsalek L., A. Hauber, P. Slusallek, High-speed volume ray casting with CUDA, Proc. of IEEE Symposium on Interactive Ray Tracing, Los Angeles, CA, 9-10 Aug. 2008, pp.185-185.

[8] Zlatuška M., V. Havran, Ray Tracing on a GPU with CUDA – Comparative Study of Three Algorithms, Proc. of WSCG Conf., Feb. 1-4, 2010, pp.69-76.

[9] Gunther J., S. Popov, H.-P. Seidel, P. Slusallek, Realtime Ray Tracing on GPU with BVH-based Packet Traversal, Proc. of IEEE Symposium on Interactive Ray Tracing, 10-12 Sept. 2007, pp.113-118.

[10] Woop S, J. Schmittler, P. Slusallek, RPU: a programmable ray processing unit for realtime ray tracing, Proc. of ACM SIGGRAPH: ACM Transactions on Graphics, Vol. 24, No.3, July 2005.

Filip Slavov
Technical University of Sofia
Computer Systems and Technologies
Sofia 1000, bul. “Kl. Ohridski” 8
Bulgaria
E-mail: NibbleByte@mail.bg

Article withdrawn at the author's request

Article withdrawn at the author's request

Article withdrawn at the author's request

Article withdrawn at the author's request

Article withdrawn at the author's request

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Cloud Collaboration - Platform for online collaboration in open source distributed cloud systems

George Suciu
Teacher Coordinator: Simona Halunga

Abstract

In this paper we present the way in which SlapOS is used to run an online collaboration platform in the cloud. This will include: chat, sharing document, voice, video and calendar scheduling. SlapOS is the first open source operating system for Distributed Cloud Computing. SlapOS is based on a grid computing daemon – called slapgrid – which is capable of installing any software on a PC and instantiate any number of processes of potentially infinite duration of any installed software. Slapgrid daemon receives requests from a central scheduler – the SlapOS Master – which collects back accounting information from each process. SlapOS Master follows an Enterprise Resource Planning (ERP) model to handle at the same time process allocation optimization and billing.

1 Introduction

Cloud Computing is traditionally divided in three market segments: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). To better understand cloud communications, it is useful to understand the different service models of cloud computing. The best known is Software as a Service (SaaS) where the customer purchases access to an application of a service hosted in the cloud. Platform as a Service (PaaS) refers to access to platforms that allows the customers to deploy their own applications in the cloud, and Infrastructure as a Service (IaaS) is at a lower level with access to the systems, storage, network connectivity, and OS management.

The traditional layered approach implicitly supposes that the IaaS layer of Public Clouds is implemented by very large server farms, which are supposed to provide optimal efficiency through economies of scale and automation. The IaaS layer of Private Clouds is implicitly supported by expensive Storage Area Networks (SAN) hardware. There are several efforts already under development, including the Distributed Management Task Force (DMTF) Open Cloud Standards Incubator, the Open Grid Forum's Open Cloud Computing Interface working group, and the Storage Network Industry Association Cloud Storage Technical Work Group. In France the Free Cloud Alliance promotes the first Open Source Cloud Computing Stack which covers applications like Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) with a consistent set of technologies targeted at high performance and mission critical applications. A great resource to see the spectrum of cloud standards activity can be found at the OMG's cloud-standards.org wiki [1].

On the SaaS side, cloud communications services support embedding communications capabilities into business applications such as Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) systems. For "on the move" business people, these services can be accessed through a smartphone, supporting increased productivity while away from the office. These services are over and above the support of service deployments of VoIP systems, IP contact centers, collaboration systems, and conferencing systems for both voice and video. These services can be accessed from any location and linked into current services to extend their capabilities, as well as stand alone as service offerings. In terms of social networking, using cloud-based communications provides click-to-call capabilities from social networking sites, and access to Instant Messaging systems and video communications, broadening the interlinking of people in the social circle.

With the increasing adoption of IPv6, 1 Gbps fiber to the home, multi-core CPUs and Solid State Disks, the traditional view on Public Cloud based on very large groups of servers, or the traditional view on Private Cloud based on corporate Storage Area Networks, is no longer as relevant as it used to be. A new form of Cloud massively distributed cloud can be implemented nowadays to provide Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) with new levels of cost efficiency, drastically reduced environmental impact and higher protection of citizen Freedom. It is now possible to implement Cloud Computing through a myriad of servers located in everyone's home or in every worker's desk. Internet transit marginal cost reduces close to zero, distributed storage software is used instead of SAN. Energy is saved by eliminating the need for air cooling and by reusing natural server dissipation for house or office heating. Citizen Freedom is better protected by preventing data to be centralized under control of single entity. Massively distributed cloud is already acknowledged and supported by the OW2 Consortium [2], by the Free Cloud Alliance [3] and by the Free Software Foundation [4].

We will introduce in this article SlapOS, the first open source operating system for Distributed Cloud Computing. SlapOS is based on a grid computing daemon called slapgrid which is capable of installing any software on a PC and instantiate any number of processes of potentially infinite duration of any installed software. Slapgrid daemon receives requests from a central scheduler the SlapOS Master which collects back accounting information from each process. SlapOS Master follows an Enterprise Resource Planning (ERP) model to handle at the same time process allocation optimization and billing. SLAP stands for "Simple Language for Accounting and Provisioning".

This structure has been implemented for cloud-based automation of ERP and CRM software for small businesses and aspects are under development under the framework of the European research project "Cloud Consulting" [5]. The goal of Cloud Consulting is to create new technologies which automate the configuration of Enterprise Resource Planning and Customer Relationship Management software for the benefit of SMBs. The technologies to be used are Knowledge Engineering, Machine Learning and Data Mining.

This article is a continuation of the material "Cloud Consulting: ERP and Communication Application Integration in Open Source Cloud Systems" [15]

2 Characteristics of cloud computing

2.1 SlapOS Architecture

SlapOS is an open source Cloud Operating system which was inspired by recent research in Grid Computing and in particular by BonjourGrid [6]–[8] a meta Desktop Grid middleware for the coordination of multiple instances of Desktop Grid middleware. It is based on the motto that "everything is a process". SlapOS is now an OW2 project. Fig. 1 shows the current architecture.

SlapOS defines two types of servers: SlapOS Nodes and SlapOS Master. SlapOS Nodes can be installed inside data centers or at home. Their role is to install software and run processes. SlapOS Master acts as a central directory of all SlapOS Nodes, knowing where each SlapOS Node is located and which software can be installed on each node. The role of SlapOS Master is to allocate processes to SlapOS Nodes.

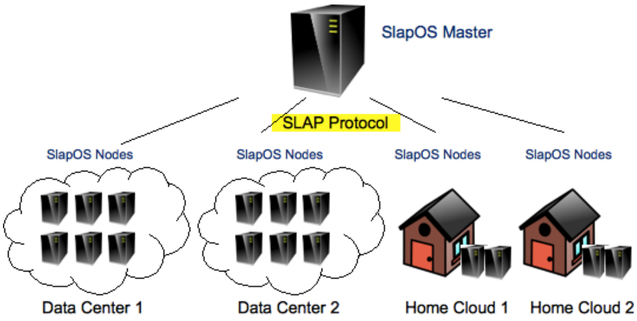


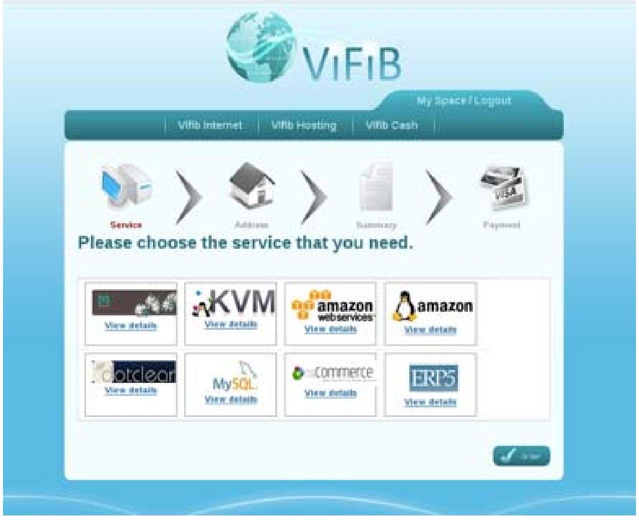
Fig. 1. The SlapOS Architecture

SlapOS Nodes and SlapOS Master exchange are interconnected through the HTTP and XML based SLAP protocol. SlapOS Master sends to each SlapOS Node a description of which software should be installed and executed. Each SlapOS Node sends to SlapOS Master a description of how much resources were used during a given period of time for accounting and billing purpose.
From a user point of view, SlapOS Node looks like an online shop for Cloud Computing resources. The user connects to SlapOS Master through a simplified front end, selects which software he or she needs. SlapOS Master then allocates the software onto a SlapOS Node and provides the connection information to the user. The allocated software can be of any type: virtual machine, database server, application server, web cache front end, etc

2.2 An example of SlapOS frontend

Title From a developer point of view, as shown in Fig. 2, SlapOS is a simple and universal API to create instances of any software daemon through a programmatic interface. A simple code allows a developer to request a new instance of a memcache server by invoking the request method of SlapOS API.

Memcache [9] is a widely adopted key-value store protocol which is used to cache values in large scale web infrastructure. It is usually installed and configured by system administrators using packaging systems such RPM or DEB. In this example, a single method call does in a few seconds what a human system administrator would have done in few minutes at best.



Please choose the service that you need.

Fig. 2. An example of SlapOS front-end

2.3 SlapOS kernel

SlapOS is implemented as an extension of widely adopted open source software: GNU/Linux, Buildout [10] and Supervisord [11] and as depicted on Fig. 3. The only new software introduced by SlapOS is Slapgrid, a daemon in charge of implementing the SLAP protocol on each SlapOS Node.

Each time slapgrid receives a request from SlapOS master to install a software, it downloads a description of that software in the form of so-called buildout profile. It then runs the buildout bootstrap process to install the software. Buildout is a Python-based build system for creating, assembling and deploying applications from multiple parts, some of which may be non-Python-based. Buildout can be used to build C, C++, ruby, java, perl, etc. software on Linux, MacOS, Windows, etc. Buildout can either build applications by downloading their source code from source repositories (subversion, git, mercurial, etc.) or by downloading binaries from package repositories (rpm, deb, eggs, gems, war, etc.). Buildout excels in particular at building applications in a way which is operating system agnostic and to automate application configuration process in a reproducible way.

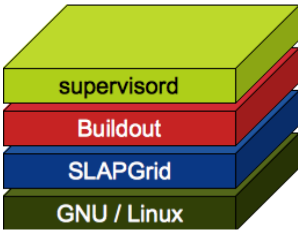


Fig. 3. The SlapOS kernel

Each time slapgrid receives a request from SlapOS master to run a software as a new process, it calls first buildout to create all configuration files for that process then delegates to supervisord the execution of the process. Supervisor is a client/server system that allows its users to monitor and control a number of processes on UNIX-like operating systems. It provides a higher abstraction and flexibility than traditional sysinit. After some time, a typical SlapOS Node will include multiple software applications, as shown in Fig. 4, and, for each software application, multiple instances, each of which running in a different process. For example, both Mediawiki and OS Commerce could be installed onto the same SlapOS Node, with six instances of each being run as processes. By running software instances as processes, rather than by creating a virtual machine for each software instance as one would do with Amazon EC2 [12], SlapOS is able to use hardware resources and RAM in particular more efficiently.

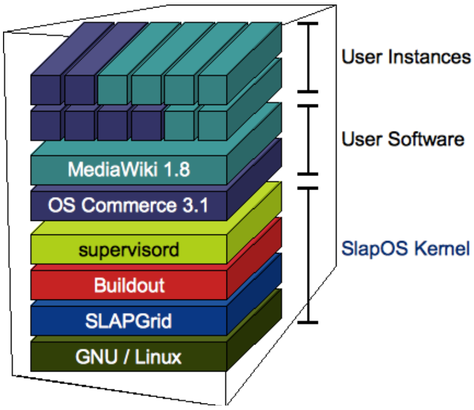


Fig. 4. SlapOS implementation

2.4 Details about implementation

SlapOS Master runs ERP5 Cloud Engine, a version of ERP5 open source ERP capable of allocating processes in relation with accounting and billing rules, as shown in Fig. 5. Initial versions of SlapOS Master were installed and configured by human. Newer versions of SlapOS Master are implemented themselves as SlapOS Nodes, in a completely reflexive ways. A SlapOS Master can thus allocate a SlapOS Master which in turn can allocate another SlapOS Master, etc.

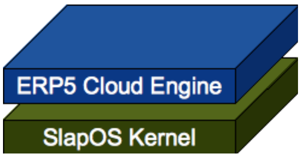


Fig. 5. SlapOS Master.

3 Research results and future design

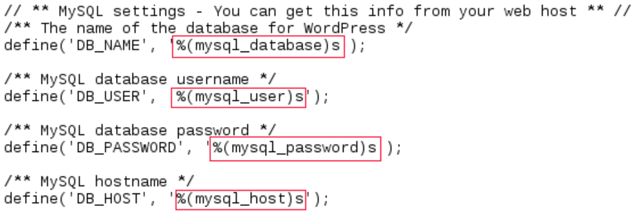
In this chapter we present the way in which SlapOS is used to request and run an online collaboration platform such as WordPress for chat, sharing document, voice, video and calendar scheduling. This will include: writing a configuration template, collecting a profile, installing and running a software instance [13], [14].

We will use the SlapOS platform implemented during the “Cloud Consulting” project [5] hosted on several servers running Ubuntu Linux – Apache – MySQL template with current software release.

3.1 Research Results

WordPress is a more advanced PHP application then other online collaboration software and is a good example of how to use fully featured SlapOS system – with asynchronous requesting of services.

For deploying the software first we need to create a configuration template. In order to simplify this task, one template can be published in a GIT repository and can be downloaded from an URL

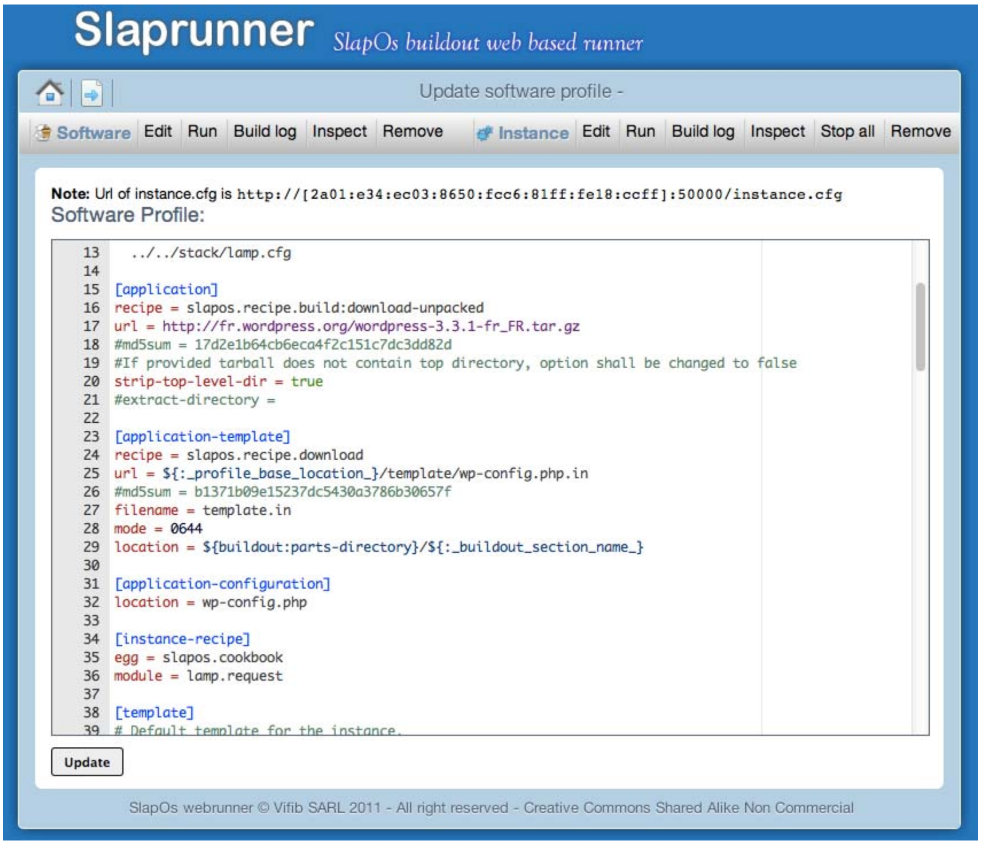


```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', '%(mysql_database)s');

/** MySQL database username */
define('DB_USER', '%(mysql_user)s');

/** MySQL database password */
define('DB_PASSWORD', '%(mysql_password)s');

/** MySQL hostname */
define('DB_HOST', '%(mysql_host)s');
```

Fig. 6. Configuration template

The template, shown in Fig.6, is modified according to the local database settings and can be published anywhere on web-based file storages such as cloudooo.com, Google Docs, etc. The URL of the template must allow access to raw unformatted data, without any HTML tags.

Secondly a SlapOS software profile is collected and configured from the GIT repository and pasted in the SlapOS Web Runner. The configuration requires setting the WordPress download URL, the previously prepared template URL and the path where WordPress will process the configuration file, as shown in Fig. 7


Slaprunner
SlapOs buildout web based runner
Update software profile -
Software
Edit
Run
Build log
Inspect
Remove
Instance
Edit
Run
Build log
Inspect
Stop all
Remove
Update
SlapOs webrunner © Vifib SARL 2011 - All right reserved - Creative Commons Shared Alike Non Commercial

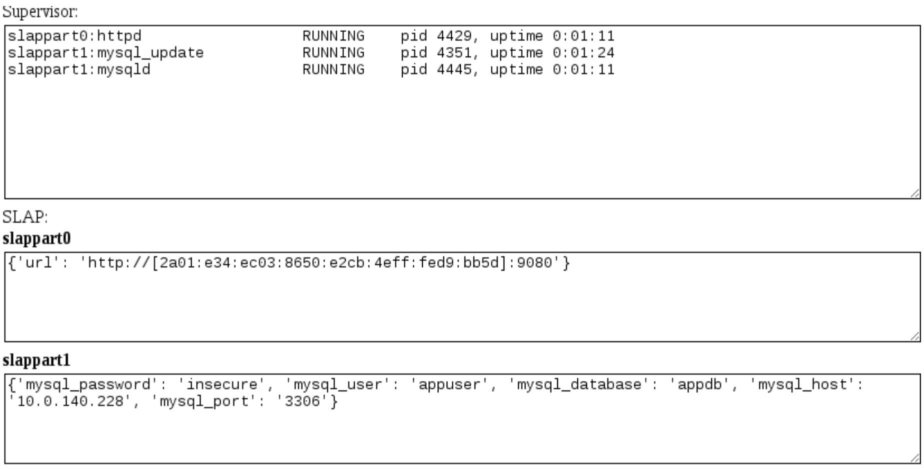
Note: Url of instance.cfg is `http://[2a01:e34:ec03:8650:fcc6:81ff:fe18:ccff]:50000/instance.cfg`

Software Profile:

```
  ../../stack/lamp.cfg

[application]
recipe = slapos.recipe.build:download-unpacked
url = http://fr.wordpress.org/wordpress-3.3.1-fr_FR.tar.gz
#md5sum = 17d2e1b64cb6eca4f2c151c7dc3dd82d
#If provided tarball does not contain top directory, option shall be changed to false
strip-top-level-dir = true
#extract-directory =

[application-template]
recipe = slapos.recipe.download
url = ${:_profile_base_location_}/template/wp-config.php.in
#md5sum = b1371b09e15237dc5430a3786b30657f
filename = template.in
mode = 0644
location = ${buildout:parts-directory}/${:_buildout_section_name_}

[application-configuration]
location = wp-config.php

[instance-recipe]
egg = slapos.cookbook
module = lamp.request

[template]
# Default template for the instance.
```

Figure 7. SlapOS software profile

Thirdly a SlapOS instance profile for static web site is opened from the GIT repository and used to build the software. As asynchronous request method is used it is required that the instantiation runs three times to simulate computer partition requesting instantiation of another partition. First run will send request to simulated SlapOS Master, the second one will instantiate the requested partition and the third one will finalize instantiation of requester partition, as shown in Fig. 8.



Supervisor:

```
slappart0:httpd                  RUNNING    pid 4429, uptime 0:01:11
slappart1:mysql_update           RUNNING    pid 4351, uptime 0:01:24
slappart1:mysqld                 RUNNING    pid 4445, uptime 0:01:11
```

SLAP:

slappart0

```
{'url': 'http://[2a01:e34:ec03:8650:e2cb:4eff:fed9:bb5d]:9080'}
```

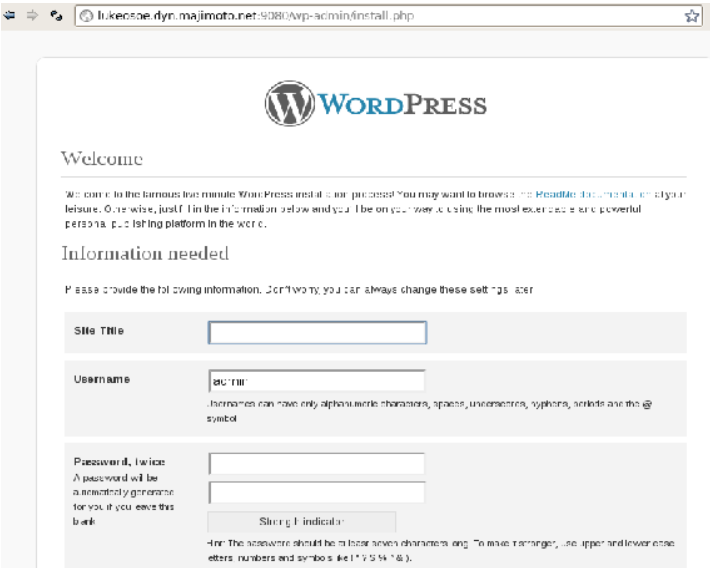
slappart1

```
{'mysql_password': 'insecure', 'mysql_user': 'appuser', 'mysql_database': 'appdb', 'mysql_host':
'10.0.140.228', 'mysql_port': '3306'}
```

Figure 8. SlapOS inspect of the instance

There are two partitions created, one holds the installation of MySQL and the other partition holds the WordPress application automatically configured to connect to MySQL together with Apache and PHP.

After successful instantiation it is possible to access WordPress and follow the install screen, as shown in Fig. 9. In order to finish the installation and free the resources it is required to stop the instance.



lukecsoe.dyn.majimoto.net:9080/wp-admin/install.php

Welcome

Welcome to the famous five minute WordPress installation process! You may want to browse the ReadMe documentation at your leisure. Otherwise, just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username: admin
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods and the @ symbol

Password, twice
A password will be automatically generated for you if you leave this blank

Strength indicator

Hint: The password should be at least seven characters long. To make it stronger, use upper and lower case letters, numbers and symbols like ! " ? $ % ^ &).

Figure 9. WordPress installation screen on cloud

3.1 Future design

SlapOS uses buildout and a single URL to describe how to build and install software. This approach could be extended in different ways. Ideally, it should be possible to install software using other build systems or even by using packages (DEB, RPM). This is not incompatible with SlapOS approach as long as the software which is installed provides itself a way to create multiple software instances through buildout or through any other type of templating system. It should also be possible to specify multiple URLs, embedded for example in a single file containing a list of URLs. This would be useful to specify different releases of the same software which are all considered as acceptable (ex. MySQL 5.0, MySQL 5.1, etc.).

Another evolution could be the encapsulation of best Cloud practices into either buildout recipes (for modularity) or SlapOS API (for simplicity). For example, self-controlled elasticity based on performance monitoring and feedback could be encapsulated into a class and reused by different Software Product. SlapOS will first consider the approach of buildout recipes which provides maximum flexibility before considering extending the current simple API.

SlapOS should also soon serve as a platform for open source software publisher to turn their software into multi-tenant SaaS. SlapOS has received a lot of attention from IT industry and telecommunication industry because it is capable of hosting as many as 200 ERP instances at a cost of less than 1 USD per month and per instance. It is also acknowledged that SlapOS can save huge investments in research and development for those companies which need to turn their existing software into SaaS. On the other hand know how is attracted continuously by companies such as Google, Facebook and Microsoft where it remains secret.
Another future design we envision is to deploy applications like Etherpad based on node.js to enable collaboration on documents in real-time

4 Conclusion

SlapOS is capable of allocating virtual machines, application servers, databases and even ERP applications beyond the borders of IaaS, PaaS and SaaS in Cloud Computing. Also SlapOS is in reality much more reliable than traditional approaches to Cloud because, by selecting independent sources, SlapOS can survive any case of force majeure: strike, earthquake, political decision, electricity breakout, etc. That said, SlapOS can also be used to operate traditional data centers more efficiently [15].

We have implemented the current software release of the popular WordPress content management systems for business type online collaboration applications and intend to develop configuration templates also for future versions.

With SlapOS it is also possible to create a rich library of open source applications hosted on the Cloud for research and education entities that want to have free online access to the know how of Cloud Computing. Application can be a complete business application (ex. ERP5, OS Commerce, Media Wiki, Drupal), a well defined network service (ex. memcached, cloudooo, MySQL, Apache) part of a Service Oriented Architecture (SOA), a virtual machine (ex. Kvm) or even an instance of an IaaS provider (ex. NiftyName, EC2, GoGrid, etc.).

Nevertheless development is still needed for Slaprunner, the SlapOS buildout web based runner so that software profiles and release versions can be better managed and many other web-based applications added to the software release directory.

SlapOS is the proposed testbed to make sure that Distributed Cloud Computing, Communications and Consulting knowledge remains shared and open.

References

[1] http://cloud-standards.org (Feb. 2012)

[2] http://www.ow2.org (Feb. 2012)

[3] http://www.freecloudalliance.org/ (Feb. 2012)

[4] http://www.fsf.org/ (Feb. 2012)

[5] http://www.beiaro.eu (Feb 2012)

[6] Heithem Abbes, Christophe C´erin, and Mohamed Jemni.*Bonjourgrid as a decentralised job scheduler. In APSCC 08.Proceedings of the 2008 IEEE Asia-Pacific Services Computing Conference*, Washington, DC, USA,2008. IEEE Computer Society, pages 89–94

[7] Heithem Abbes, Christophe C´erin, and Mohamed Jemni.Bonjourgrid: *Orchestration of multi-instances of grid middlewareson institutional desktop grids. In 3rd Workshop onDesktop Grids and Volunteer Computing Systems (PCGrid2009)*, en conjonction avec IPDPS 2009, Rome, Italie, 29 Mai 2009, pages 118-123.

[8] Heithem Abbes, Christophe C´erin, Mohamed Jemni: *A decentralized and fault-tolerant Desktop Grid system for distributed applications. Concurrency and Computation: Practice and Experience 22(3)* (2010): pages 261-277

[9] Memcached: a free and open source, high-performance, distributed memory object caching system. http://memcached.org/

[10] Buildout - software build system reloaded http://www.buildout.org/

[11] Supervisor: A Process Control System http://supervisord.org/

[12] Tze Ng, Guohui Wang "*The impact of virtualization on network performance of Amazon EC2 data center", IEEE INFOCOM 2010 - 029th IEEE International Conference on Computer Communications*, Vol. 29, no. 01, March 2010, pages 1 - 9

[13] Patricia Takako Endo, André Vitor de Almeida Palhares, Nadilma Nunes Pereira, Glauco Estácio Gonçalves, Djamel Sadok, Judith Kelner, Bob Melander, Jan-Erik Mångs "*Resource Allocation for Distributed Cloud: Concepts and Research Challenges*", IEEE Network, Vol. 25, no. 4, July 2011, pages 42 - 46

[14] Bengt Ahlgren, Pedro A. Aranda, Prosper Chemouil, Sara Oueslati, Luis M. Correia, Holger Karl, Michael Söllner, Annikki Welin "*Content, Connectivity, and Cloud: Ingredients for the Network of the Future*", IEEE Communications Magazine, Vol. 49, no. 7, July 2011, pages 62 - 70

[15] George Suciu, Octavian Fratu, Simona Halunga, Cristian George Cernat, Vlad Andrei Poenaru, Victor Suciu, *"Cloud Consulting: ERP and Communication Application Integration in Open Source Cloud Systems", 19th Telecommunications Forum - TELFOR 2011*, IEEE Communications Society, 2011, pages 578-581.

George Suciu
Spiru Haret University, Bucharest
ROMANIA
E-mail: george@beia.ro

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Research and analysis of the method for the total tardiness time minimization of jobs with due dates on a computing device

Maksym Sukhonos, Dmytro Lienko
Teacher Coordinator: Sergey Minukhin

Abstract

Methods for solving the problem of minimizing the total tardiness time of jobs with due dates to meet the following computing resources have been considered and analyzed. Heuristic algorithms for the solution of the ranking based approach have been examined. The computing experiments for validation of the algorithms effectiveness have been conducted. Calculating the relative error and time complexity for different realizations of the input queue of jobs are presented. The suitability of algorithms for usage in real-time systems has been examined.

Keywords: resource, Hamiltonian path, optimal schedule, tardiness time, due date, error, time complexity.

1 Introduction

In modern distributed systems, that use clusters, processing units, workstations etc. as a resource (e.g. in Grid-systems [1]), jobs entering some resource to be processed have different processing times and due dates. Deviation from this parameters leads to significant increase of computing cost (penalties) or violation of execution conditions. Thus it is relevant to develop mathematical representation and software implementation of the algorithm that should be provided to single calculating units (devices) of computing systems in order to minimize total tardiness of all the jobs that come continuously, at the same time, and according to their chosen execution strategy.

The formulation of the problem for composing an optimal sequence of jobs (schedule) stands as follows. We assume that a range of independent jobs $J = \{j_1, j_2, \ldots, j_n\}$ entered a processing unit and every job will be executed on it. For each job of the range its processing time l_j and due date d_j are given. The problem is to determine the order (sequence) of execution of all jobs that enter a calculation unit at the same time, which will minimize their total tardiness time.

In section 2, the formulation of the problem is described. In section 3, the results of the computing experiments are given.

2 Formulation of the problem of total tardiness time minimization

Mathematically, the problem of execution time minimization for the jobs with due dates, is formalized as follows: it is needed to build a jobs execution schedule that minimizes the function

$$f = \sum_{j=1}^{n} \max(0, C_j - D_j) \to \min, \tag{1}$$

where C_{j} denotes the real completion time of the job *j*, D_{j} denotes the job execution due date, and *n* denotes number of tasks in the range.

The review of methods for solving this problem is given in [2 – 4], according to which, this problem is NP-hard. However, the methods presented have a general disadvantage: The solving of the problem in real-time is complicated because of the exponential growth of alternatives being analyzed. In paper [5], a PDC-algorithm is proposed.

Hence, development of model and method for solving the problem and justification of its application upon condition of small time complexity and error in real-time in computing system with non-clustered resources is relevant in terms of evaluation of effectiveness of calculating resources used in different organizations. In particular, it has significant importance when forming different kinds of resources of such systems (their organization) with partially clustered resources (e.g., Grid-systems) and with changing input data intensity.

The approach using the spanned tree to solve the problem and algorithms for its implementation are proposed and researched in [6].

3. Computer experiments and analysis of results

The developed application implements exhaustive search algorithm, direction optimization algorithm and direction optimization algorithm with dominance rule.

3.1 Test data generation

For testing the application in order to increase sufficiency of gained results the library [7] and stochastic sequence generator were used. The developed generator produces sequences with different characteristics.

The instances are generated as follows: for each job *j* (*j* = 1..*N*), an integer processing time l_j was generated from the uniform distribution [1, 100]. In order to simulate different hardness the due dates are generated from different uniform distributions. For an average tardiness factor *TF* = {0.2, 0.4 ,0.6 0.8, 1.0} and a relative range of due dates *RDD* = {0.2, 0.4 ,0.6 0.8, 1.0}, from the uniform distribution

$$\left\lfloor L \cdot \left(1 - TF - \frac{RDD}{2}\right), L \cdot \left(1 - TF + \frac{RDD}{2}\right)\right\rfloor, \tag{4}$$

an integer due date d_j is generated for each job, where

$$L = \sum_{j=1}^{N} l_j \,. \tag{5}$$

Thus, for each observation 25 instances are generated.

To determine the sufficient number of instances in a sample, 1, 2, 3 and 4 samples with 25, 50, 75 and 100 instances respectively were generated with 40, 60, 80, 100, 120, 150 jobs. The schedules for the samples were built with direction optimization algorithm, whereupon mean and standard deviation for minimum, maximum and average tardiness time were calculated.

The results of this experiment justify practical implication. 50 to 75 instances in a sample are enough for it to be sufficient because mean and standard deviation for 75 instances (in some cases for 50 instances) don't vary significantly comparing to 100 instances.

For instances with number of jobs from 3 to 11, 3 observations including 75 instances were generated. The optimal schedules were built with direction optimization method and direction optimization method with dominance rule.

For the exhaustive search algorithm the execution time grows exponentially, so for a greater number of jobs heuristic algorithms were used.

In order to estimate the accuracy of heuristic algorithms approximation comparing to exhaustive search one, mean and standard deviation of relative error were calculated with

$$M_\varepsilon = \frac{\sum_{j=1}^{N} \frac{t_j^{opt} - t_j^{exh}}{t_j^{exh}}}{N}, \tag{6}$$

$$\sigma_\varepsilon = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\frac{t_j^{opt} - t_j^{exh}}{t_j^{exh}} - M_\varepsilon\right)^2}, \tag{7}$$

where t_j^{opt} denotes tardiness time of the job obtained from direction optimization algorithm, t_j^{exh} denotes tardiness time of the job obtained from exhaustive search algorithm.

3.2 Experimental results analysis

To estimate time complexity of reviewed algorithms (exhaustive search, direction optimization, and direction optimization with dominance rule) computing experiments were performed.

In Table 1, results of the experiment with number of jobs from 3 to 11 are presented. The schedules were built with exhaustive search, direction optimization, and direction optimization with dominance rule algorithms.

Table 1
Results of the experiment with 3 to 11 jobs using all the described algorithms

Number of tasks	Exhaustive search algorithm execution time, ms	Direction optimization algorithm execution time, ms	Direction optimization algorithm with dominance rule execution time, ms	Mean of relative tardiness time error for optimization algorithm, %	Mean of relative tardiness time error for optimization with dominance rule algorithm, %
3	11	1	1	26,69	26,69
4	3	1	1	3,35	2,35
5	12	2	3	11,13	11,79
6	80	4	5	21,51	35
7	525	6	7	25,65	25,5
8	4448	9	10	10,11	8,21
9	40342	13	14	12,51	10,93
10	414706	18	20	27,66	36,77
11	4644922	25	28	3,73	5,23

The approximation functions for execution times were built using cubic polynomials in order to estimate time complexity: for exhaustive search algorithm (8), for direction optimization algorithm (9), and for direction optimization algorithm with dominance rules (10).

$$T(N)_{exh.search} = 51820.05n^3 - 944767.87n^2 + 5329583.96n - 9176552.53, \tag{8}$$

$$T(N)_{opt.} = 0.03n^3 - 0.22n^2 + 1.18n - 2.02, \tag{9}$$

$$T(N)_{opt.D} = 0.05n^3 - 0.55n^2 + 3.22n - 5.72. \tag{10}$$

Thus, performed experiments proved obtained in [6] theoretical time complexity of reviewed algorithms (***$O(n^3)$***) – dependence of execution time on jobs number.

Fig. 3, 4 present execution time dependencies on number of jobs for direction optimization algorithm and direction optimization algorithm with dominance rule. Fig. 5 presents heuristic algorithm relative error comparing to exhaustive search algorithm.

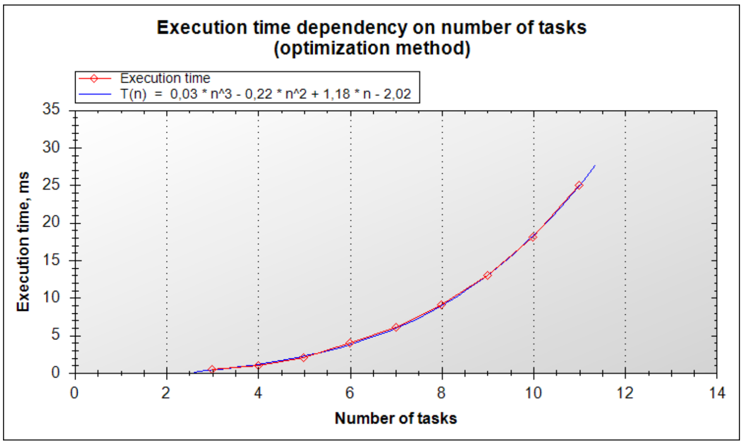


Figure 3. Execution time dependency on number of jobs for heuristic algorithm of optimization

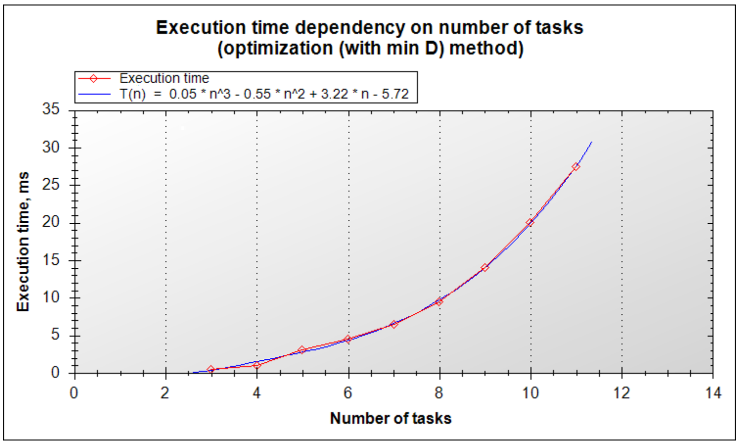


Figure 4. Execution time dependency on number of jobs for heuristic algorithm of direction optimization with dominance rule

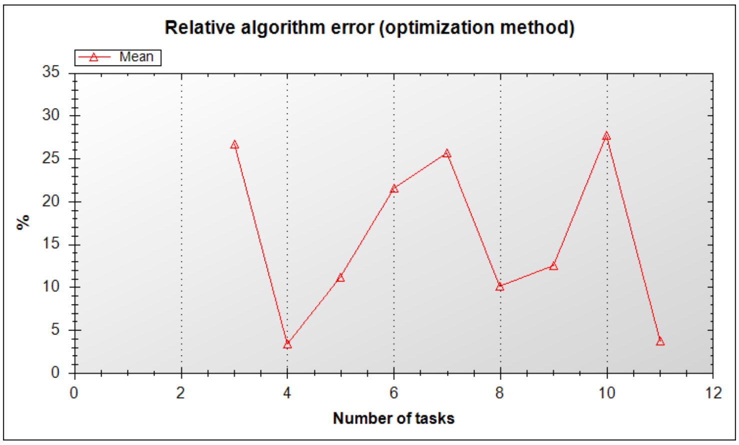


Figure 5. Relative error of heuristic algorithm of direction optimization

The experiments were conducted on a personal computer with dual-core processor with a frequency of 3.2 GHz and 4 GB of RAM under Windows 7. To ensure the accuracy of time measurement experiments were conducted on a single processor core.

4 Conclusion

Theoretically derived time complexity ***$O(n^3)$*** of solving the problem based on direction optimization algorithms has been experimentally substantiated.

The program for generating stochastic testing samples has been developed. It allows conducting computing experiment and a qualitative analysis of test results for different representative test samples.

Program implementation of the exhaustive search algorithm and two direction optimization algorithms allows evaluating the results and analyzing the relative error of the developed methods.

The time complexity of the method, resulting from a computer experiment justifies the use of the examined algorithms for real-time computing systems.

5 References

[1] Ponomarenko V.S. *Metody i modeli planirovaniya resursov v Grid-sistemakh: monografiya* / V.S. Ponomarenko, S.V. Listrovoy, S.V. Minukhin, S.V. Znakhur. – Kharkov: ID «INZHEK», 2008. – 408 s.

[2] Lawler, E.L. *A fully polynomial approximation scheme for the total tardiness problem*. //Operations Research Letters. – 1982 – №1. – P. 207–208.

[3] Koulamas C. *A faster fully polynomial approximation scheme for the single machine total tardiness problem* //European Journal of Operational Research. – 2009. – №193. – P. 637–638.

[4] Lazarev A.A., Gafarov E.R. *Special case of the single machine total tardiness problem is NP¡hard*. // 12th IFAC Symposium on Information Control Problems in Manufacturing. INCOM’2006.Preprints, Vol. III, Operational Research, May 17–19, 2006 – Saint-Etienne, France. – P. 155–157.

[5] Pavlov A.A., Misyura E.B. *Effektivniy tochniy PDS-algoritm resheniya zadachi o summarnom zapazdyvanii dlya odnogo pribora* // Systemni doslidzhennnya ta informatsijni tekhnolohiyi. – 2004. – №4. – S. 30–59.

[6] Minukhin S.V. *Metod minimizatsiyi chasu vykonannya zavdanj z dyrektyvnymy strokamy na neklasteryzovanomu resursi obchyslyuvalnoyi systemy* // Informatsijno-keruyuchi systemy na zaliznuchnomu transporti. – 2009. – №3. – S. 47 – 53.

[7] OR-Library: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/wtinfo.html.

Maksym Sukhonos
Kharkiv National University of Economics
Department of Information Systems
Kharkiv, 9a Lenin Ave.
UKRAINE
E-mail: maksim.sukhonos@gmail.com

Dmytro Lienko
Kharkiv National University of Economics
Department of Information Systems
Kharkiv, 9a Lenin Ave.
UKRAINE
E-mail: dmytrolenko@gmail.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Sharing interactive SVG documents through XMPP

Stefan Alexandru
Teacher Coordinator: Alexandru Radovici

Abstract

The goal of the project is to create a drawing board android application that allows users using different devices to interact while creating or discussing a presentation. For communication the application will implement an XMPP protocol and use the Google Talk server.

1 Introduction

Technology is evolving fast nowadays and it greatly influences the socio-economic aspects of our society. One of the ways it does that is by making communication easier and faster. The latest improvement regarding the technology of information and communication is the invention of the Smartphone, a mobile device which performs the tasks of a regular phone, those of a computer and others such as GPS and Camera functions.

The potential of this technology is very high and through my project I intend to exploit this potential in order to bring improvements in fields as education, research and management by helping users to express and share ideas.

The application will allow users to edit a drawing board adding and manipulating text and a range of shapes and symbols and to share with other users the changes they made to the board.

2 Technologies involved

2.1 Android

Android has taken big steps over the last few years. What began as a promising but little-implemented operating system has matured into a strong OS powering a slew of mobile devices, tablets and netbooks. Currently there are over 600 million devices in use and this number increases by 850000 every day.Android’s kernel is based on a Linux kernel that was modified by Google. It doesn’t support standard GNU libraries or X Window System. Another particularity is that in Android every application runs under a different user. [1]

2.2 XMPP

Standing for Extensible Messaging and Presence Protocol, XMPP is an open-source technology for real-time communication. It powers a wide range of applications, the one we are interested in being transferring SVG documents. [7]

2.3 SVG

Scalable Vector Graphics is an XML-based file format for vector 2D graphics. The advantage of vector graphics is that they use less memory and they don’t lose their clarity when displayed on a bigger resolution. SVG documents can be edited with any text editor and follow the structure of an XML documents. [9]

3 Implementation

The application consists in a vector drawing board. It is designed in a way that allows the information about the drawing to be efficiently memorised, displayed and sent to other users. Efficiency is of keen importance as the application has to work in real time and the mobile devices have a limited processing power.

The board is being memorised in a structure of drawings, a class extended by every element that can be drew. Each of those elements has a function which allows it to draw itself on the canvas that will be displayed in the application and on the SVG document that will be used for saving the project for sending it to other users. The application is able to create the drawings structure based both on user’s actions in the drawing environment and on the SVG documents it receives.

The drawing environment allows the user to edit the board and to configure the way his session will communicate with other users’ sessions. For editing the board he will be able to create new drawings and to select and modify, respectively delete the previously draw ones. He will be able to draw freely, chose from a range of predefined shapes, and insert text; also the user can chose the colour and the transparency.

The predefined objects the user can put on the board are rectangles, ellipses, polygons, some symbols, images he can load from his device and rectangles with text. The rectangles and the ellipses will have two colours, one for the outline and one for the fill.

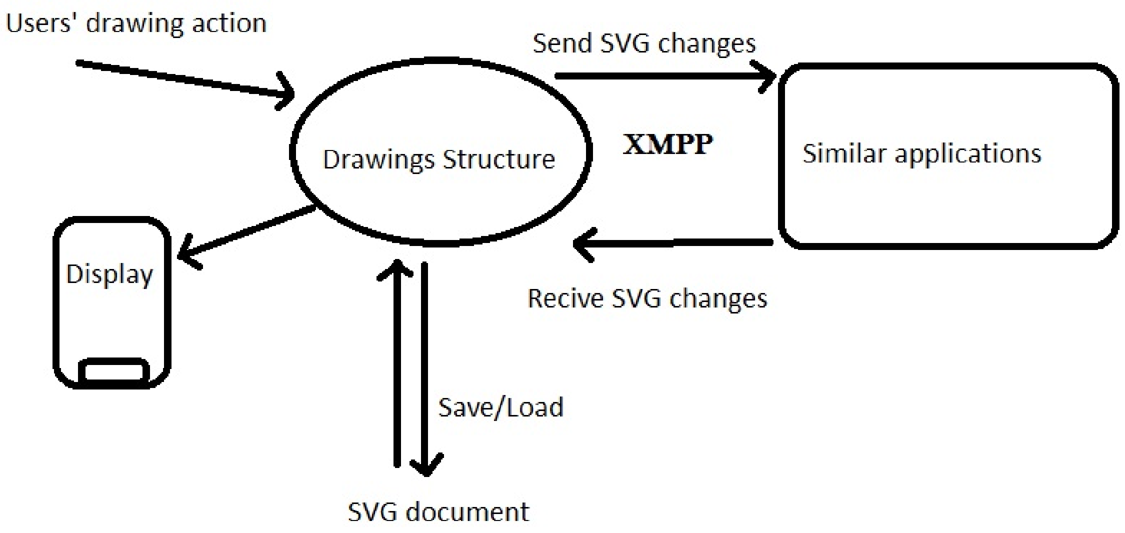
Besides the usual colours, the user will also be able to chose custom colours, by using a colour picker the application integrates. [5]

For allowing the user to create more complex projects, he will be able to create projects that have more than one page. For each page the application has to remember a drawing structure so I defined a data structure that can remember a page’s drawings structure and created an array of this type in order to retain the entire project.

For sharing the drawings with other users the application will implement XMPP, a distributed protocol running on a Google Talk server which would only require the user to have a free Google Talk account. The messages will describe the changes brought to the board using SVG coding. Also each project will have an administrator which will configure the rights the other users have on the board.

The user will also be allowed to save and load the project, each page as a SVG document. In order to do that the application will have to implement the functions of a SVG editor. It will be able to parse and understand the content of a SVG file in order to save the drawing in the structures used by the application and to generate SVG code based on the drawings structure.

The following scheme explains the way the application interacts with the user and with other applications.



4 Conclusions

The aim of this project is to take advantage of the recent technological improvements regarding the mobile devices in order to bring improvements in the fields of education, research and management in general, by implementing an android application allowing users to cooperate in creating common presentation.

The users will be able to draw freely or by using a range of predefined shapes and to insert text. He will also be able to select any object on the board and edit it (change it's position, size and colour or delete it). Every change the user does to the board will be shared with his peers.

The application will be able to save and load projects as SVG documents and will communicate with other sessions of the same application, or with similar applications through an XMPP protocol.

References

[1] Mark Murphy, *Begining Android*, Springer-Verlag, New York , Apress edition.
[2] http://en.wikipedia.org/wiki/Android_(operating_system)
[3] Rick Rogers, John Lombardo, Zigurd Mednieks, G. Blake Meike, *Android Application Development: Programming with the Google SDK,* May 20, 2009
[4] Sayed Hashimi,Satya Komatieni, Dave MacLean, *Pro Android 2*, Apress edition

[5] *http://developer.android.com/resources/samples/ApiDemos/src/com/example/android/apis/graphics/ColorPickerDialog.html*

[6] Johansson Leif, XMPP as MOM, *Greater Nordic Middleware Synposium,* Oslo:University of Stockholm, April 18 2005.

[7] William Wright, Dana Moore, *Jabber Developer's Handbook, July 18, 2003*

[8] Internet Engineering Task Force, *XML Media Types*, RFC 3023, January 2001. p. 24.

[9] J. Eisenberg, *SVG Essentials (O'Reilly XML),* February 12, 2002

STEFAN Alexandru
Universitatea Politehnica din Bucuresti
Facultatea de Inginerie in Limbi Straine
Splaiul Independentei nr. 313, sector 6
ROMANIA
E-mail: stefan_alexandru_2005@yahoo.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Bricks breaking game

Zoltan Tamasi, Zsigmond-Attila Szasz, Izabella Szornyi-Chis
Teacher Coordinator: Lector univ. Dr. Anca Vasilescu

Abstract

Bricks breaking game it's a 2D game developed using Java and C++ programming languages in which the player, controlling a bouncing ball with a movable paddle, must destroy all the bricks located at the top of the screen, without letting the ball to escape at the bottom of the screen. The player can control the paddle using the keyboard, mouse, or remotely, with the help of a web camera, using his own hand.

1 Introduction

Bricks Breaker is an arcade game developed by Atari, Inc and introduced in 1976.
The current application was made in the following programming languages: Java and C++.The graphical user interface and the logic part of the game were programmed in Java,the paddle's controlling mechanism by the player is done in C++. It processes the images captured by a webcam. Our goal is to highlight the advantage of motion tracking, which can provide remote control of the game.

1.1 Game rules

Bricks is an addictive game, with the following rules:
The *game starts* with moving your hand upwards, action which launches the ball off the paddle.
The ball travels across the screen, bouncing off the top and side walls of the screen. When a brick is hit, the ball bounces away and the brick is destroyed.

The ball moves according to the angle of meeting point with the paddle and trajectory of the ball.
This angle is influenced by the impact zone of the ball and paddle.The paddle is actually behaving like a semicircle when hitting the ball (otherwise the ball would move with the same angle all the time).
Hitting walls will not change the ball's trajectory,the ball bounces back with the angle of 180 minus entry angle.

For *controlling the paddle* we can use the keyboard, mouse, or remote control with motion tracking.The goal of this game is to destroy all the bricks above the paddle. After destroying all the bricks the player reaches the next level.
If the ball does not touch the paddle and reaches the gap at the bottom of the window, the player loses a life. The game ends when the player loses all three lives, or goes through all the levels.

The *difficulty* of the levels is determined by the number of bricks and the type of the bricks.

The models of bricks are specially designed, so the greater difficulty levels are containing more superior type bricks.
The *bricks* are of several types depending on their destructibility, namely:
- Type 1 bricks - gray: brick disappears after a single touch of the ball
- Type 2 bricks - blue: brick is destroyed after two strikes of the ball
- Type 3 bricks - red: brick is destroyed after three strikes the ball
- Type 4 bricks - green: brick is destroyed after four strikes of the ball

For each level is given a limit time for scoring.
The *score* is determined by the number of bricks destroyed multiplied by their type and remaining time after the level is finished.

2 Tehnology

2.1 A brief overview of the OpenCV library

Library OpenCV (Open Source Computer Vision) is a library developed by Intel and used for processing digital images and movies. It is open source under the BSD license; it can be used with C/C++ and Python (and Java soon) languages and runs on Windows, Linux, Android and Mac.

It is used in the following application areas: facial recognition, gesture recognition, robotics, motion tracking and last but not least human-computer interaction. At present is supported by Willow Garage and users can download both official versions and source code via SVN, but requires to be complicated by CMAK (a tool that is used for managing the build process of software using a compiler-independent method).

The role of library in our application is to retrieve images from a webcam and process them so that we can control the paddle of our game only with gestures (not using keyboard or mouse). Thus, we want to highlight a way of using user's interaction with the computer, other than those used so far, also the ease of digital image processing and optical flow using a programming language (not Photoshop for example).
Given the fact, that the library is written in C, and we chose to work in C++. Learning how to use this library was very fast and easy.

2.2 Lucas-Kanade Method

Lucas-Kanade method, developed by Bruce D. Lucas and Takeo Kanade is used in digital image processing:
This method assumes that the displacement of the image contents between two nearby instants (frames) is small and approximately constant within a neighborhood of the point p under consideration. Thus the optical flow equation can be assumed to hold for all pixels within a window centered at p. Namely, the local image flow (velocity) vector must satisfy:

$$\begin{cases} I_x(q_1)V_x + I_y(q_1)V_y = -I_t(q_1) \\ I_x(q_2)V_x + I_y(q_2)V_y = -I_t(q_2) \\ \qquad\qquad\vdots \\ I_x(q_n)V_x + I_y(q_n)V_y = -I_t(q_n) \end{cases}$$

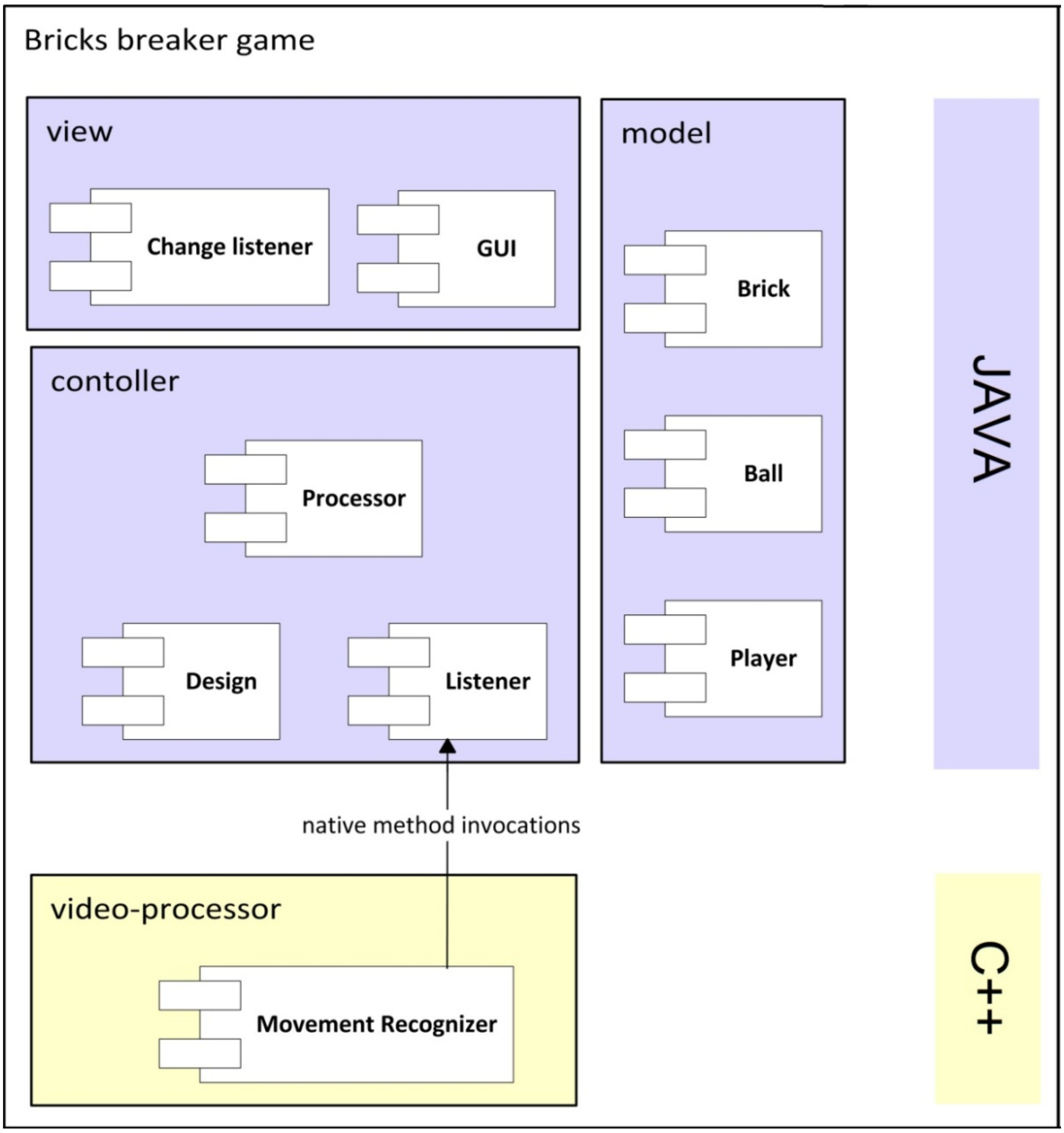
where q_1, q_2,..., q_n are pixels in the vicinity of the given point and $I_x(q_i)$, $I_y(q_i)$, $I_t(q_i)$ are partial derivatives of the I image relative to the coordinates (x, y) of the point intime t, evaluated to the point qiat the current time. The solution of these equations is given by the matrix A^TA, where A is the matrix of the system above.

The implemented algorithm in OpenCV library is used for detecting and tracking objects in a stream of images retrieved from a video source (such as a webcam for example). To have access to it, we need to include in the C++ project the following libraries:opencv_core, opencv_highgui, opencv_imgproc and opencv_video (actually the implementation of this algorithm is located in this library, but we need the rest of libraries for the extraction, processing and displaying the frames from the image-flow, and for accessing every pixel of the frames).

In our case, the algorithm is used to detect the object we want to be tracked while the application is running (such as our hand for example) and to determine actions based on the object movement in certain directions: for example if you move the previously selected object to the left, the paddle of the game will move also to the left and if you move to the right then moves to the right. The decision of which object to follow is made by o wide movement of it in a fixed direction.

3 The structure of the game

In the component-diagram we can observe the use of the model-view-controller architecture (MVC) which provides the insulation of the components.



3.1 Model component

The model component is a collection of objects of classes with settable fields that represent with their properties various components of the game, for example: brick, ball,player.

3.2 View component

View component represent the graphical user interfacein which the user interacts with the game. This component was developed with Java Swing and AWT. The game is a desktop application, which panel is a JFrame, which is the container of the graphical components.

3.3 Controller component

The controller component is the engine of the game, part which interacts with the graphical user interface, calculating and setting the properties of the models, and it's is responsible for handling events that come both from view and from the user. Here we find a "design"module which represents the placing of the bricks on the surface. This module loads the existing templates in the game but it's most important role is to publish an interface (API) through which we can create new patterns of placing the bricks, and which can be introduced in the game as plugins.In conclusion we can consider this project extensible(pluginable).

Components listed above are developed in Java programming language. Next will be described the component which is developed in C++ and has the role of the "user input" in parallel with the keyboard.

3.4 Video-processor component

Video processor component connects to a web camera (it can be a video camera or a digital camera that works while it is connected to the computer) given as input parameter.By default, if your computer has an integrated camera (such as a laptop for example) then this parameter is the number 0 or 1, or if there are multiple video devices connected to the system, then a number greater than 1. Once identified, the component begins to retrieve pictures of the device. Applying the Lucas-Kanade algorithm we can determine the most favorable points that can be traced, but it filters out the points so that remains only the ones we want to be tracked. The points are filtered by moving our object desired to be traced in a fixed direction.

Items left after filtering define the object of which horizontal movement involves the calling of a native method of the component controller. Interconnection of components developed in Java and C ++ is done through Java native interface (JNI).

4 Conclusion

The strength of this project is clearly the motion tracking part. Controlling the paddle with motion tracking is a major advantage, because wecan movethe paddle only with a gesture.

Using web cameras constitutes significant advantage, because they are financially accessible, a lot cheaper than sensors for example.

So our project aim is to highlight the benefits offered by motion tracking, motion detection facilitating the remote control of a particular game.

5 References

[1] Bruce D. Lucas and Bruce D. Lucas , *An Iterative Image Registration Technique with an Application to Stereo Vision*, Proceedings of Imaging Understanding Workshop, United States, pp 121-130, 1981.

[2] http://opencv.willowgarage.com/wiki/

[3] http://www.wikipedia.com/

[4] http://opencv.willowgarage.com/documentation/python/motion_analysis_and_object_tracking.html

[5] http://docs.oracle.com/javase/1.4.2/docs/guide/jni/spec/jniTOC.html

Tamasi Zoltan
Universitatea Transilvania Brasov
Facultatea de Matematica-Informatica
Specializarea: Informatica Aplicata
Iuliu Maniu 50, Brasov
ROMANIA
tamasi_zoltan22@yahoo.com

Szasz Zsigmond-Attila
Universitatea Transilvania Brasov
Facultatea de Matematica-Informatica
Specializarea: Informatica Aplicata
Iuliu Maniu 50, Brasov
ROMANIA
zsigmond.szasz@xu.unitbv.ro

Szornyi-Chis Izabella
Universitatea Transilvania Brasov
Facultatea de Matematica-Informatica
Specializarea: Informatica Aplicata
Iuliu Maniu 50, Brasov
ROMANIA
izabella.szornyi@xu.unitbv.ro

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Implementation of image analogies using neural networks

Jelena Tasic
Teacher Coordinator: Milan Tuba

Abstract

This work presents a new framework for image processing by example – image analogies. The framework comprises two phases: a learning phase and an application phase. In the learning phase it is learnt, on the basis of an original image –*model* and its corresponding filtered image –*master*, a *model-master* mapping. The second phase, the application phase, is the implementation of the learnt mapping – filter onto a newly selected image. Once learnt, the filter can be applied to newly selected images any number of times. For the learning of mapping neural networks have been chosen, a technique of supervised learning, considering that both the input data and the target (the filter we wish to be learnt) are given. The feed-forward architecture of neural networks is used, as well as the resilient propagation learning algorithm. This framework supports learning of different kinds of filters through selection of different kinds of input images (a *model* and *master* image). The first phase supports in addition an option of diverging from the model which we are trying to learn, so that user can evolve own images independently from the model.

1 Introduction

The aesthetic value of the result is the main motivation for using image processing in a great number of activities. The fact that software users, for instance graphic designers, most often do not have the programming expertise to develop it independently, leads to a seemingly insoluble problem: a software used in the aesthetically competitive context is programmed in an aesthetically indifferent context. In other words, the career of a graphic designer in a great measure depends on uniqueness of his or her graphic works which, paradoxically, they are forced to produce using the same software palette that their competition is using too. The softwares for image processing are for that reason in a way interactive. Although a great number of parameters allow the user a possibility for manual adjustment, the kind of the filter itself remains the same.

The proposed project investigates a possibility of learning a selected filter. The idea is that a user chooses two images, the original one and its corresponding filtered image which it gives to the software as input data on the basis of which is learnt the desired mapping – the selected filter. The result is the file that contains the neural network of the newly learnt filter. Once learnt filter can be applied to any other original image in order to produce the effect of the learnt filter. The next idea is to diverge from the model which we are trying to learn, so that a user can evolve own filters independently from the initial target model. This solution would enable design of a software dedicated to satisfaction of a certain aesthetic, directed by that same aesthetic.

The implementation of this general idea appears to be a difficult problem. However, it is only necessary to evolve the image processing program once. After that the program can be copied, distributed and used any number of time. As a result, the users whose careers depend on aesthetic originality can expect to be motivated to invest significant amount of time in evolving small programs, as their unlimited use will very quickly reward such an investment.

2 Related work

This project relies on a great number of other works. The general idea – learning filters on the basis of a given sample – is based on Hertzmann's article [1] where the term image analogies is introduced: „Given a pair of images *A* and *A'* (the unfiltered and filtered source images, respectively), along with some additional unfiltered target image *B*, synthesize a new filtered target image *B'* such that:

$$A : A' = B : B'.$$

In other words, we want to find an "analogous" image *B'* that relates to *B* in "the same way" as *A'* relates to *A*". The similarity metric is based on approximation of Markov random field model and uses pixel values for learning of a chosen filter. The statistics of the selected pixel neighbourhoods is used for calculating the relationship between the original and resulting pair of images. This algorithm uses a multi-layered representation of the images *A*, *A'* and *B* (Gaussian pyramid), together with a feature vector as well as the approximate nearest neighbour method for speeding up the mapping process. For the feature vector, in addition to RGB component a component for luminance of each pixel for calculating the metrics of distance is also used, considering that the human eye is more sensitive to lighting than to colours. Herzmann uses the same framework for curve analogies [2], learning of curve style on the basis of a given example.

As opposed to Herzmann's algorithm [1] which enables learning of different kinds of filters such as traditional image filters (for example, blurring and embossing), texture synthesis, super-resolution and artistic filters, other algorithms are specialise in narrower spectrum of filters in view of achieving better results. Freeman proposed an algorithm [3] which offers a way of guessing a range of high frequencies that are missing in the image. Algorithm takes training images as references with the aim of learning image sharpening. For the training data the algorithm uses a set of images of low resolution and the corresponding images of high resolution. The algorithm [4] solves the problem of improving the quality of personal photographs by using the favourite set of personal photographs as a training set. The algorithm consists of five phases: the automatic detection of the faces from the target photograph and the training images, segmentation of faces of the target and training images, decomposing images into colour, texture and lighting layers; performing global image corrections and performing of face-specific enhancements. The algorithm [5] is using Hertzmann's idea of image analogies where a pair of images is used for training. This algorithm uses semi-supervised learning technique, Markov's random model in order to secure global coexistence, and image quilting technique in order to secure a local coexistence. In the article [6] a method that improves a visual quality of satellite images by using the examples of super-resolution techniques is described.

Greyscale colouring of the images is a difficult problem. Levin [7] proposes a method based on a simple assumption: neighbouring pixels in the space that have a similar intensity should also be of similar colours. This algorithm does not require the precise segmentation of the image. It is necessary that the user–artist marks the colour of each region of the image he or she wishes to colour. As opposed to this approach Irony's [8] algorithm finds several points on the image which the algorithm had coloured correctly. Then Levin's approach is applied, as though the points are those that the user marked.

In medical industry a substantial amount of data is being collected, while only a small number can be correctly analysed. There is no reliable way of quickly segmenting a large amount of data. In the article [9] image analogies are used for resolving the problem of image segmentation. The method of supervised learning is employed as a technique for learning image segmentation. Medical images reviewed by the experts are used as training data.

In the article [10] a method is proposed for the automatic improvement of brightness and contrast by using supervised learning.

3 Framework and tools

The program is written in JRuby programming language. JRuby is a Java implementation of the Ruby programming language, which enables usage of all the Java libraries by writing the Ruby code. It is a dynamic and object orientated programming language. 1.9 version was used.

The proposed software uses Encog3[11] neural networks. Encog3 is an advanced framework for neural networks and machine learning. It contains classes for creating different types of networks, and also supports classes for normalising and processing data for a specific type of networks. It propagates multithreading programming. It works in many integrated developing frameworks, such as Eclipse or Netbeans. It is distributed as a JAR file. The software is developed in Netbeans programing framework. Encog3 is used for developing the recurrent and feed-forward neural networks. BasicNetwork and BasicLayer are used for creating neural networks. In addition to these two classes, activation functions are also used. The assumed activation function is hyperbolic tangent. Beside this activation function Encog 3 also supports the binary, linear, sigmoid, logarithmic and other activation functions, where the very nature of the problem suggests the usage of the specific one. A very effective form of training feed-forward and simple recurrent neural networks is propagation training. Some Encog3 forms of propagation training are: backpropagation, quick propagation, Manhattan update rule, resilient propagation.

4 The Proposed Solution vs. Traditional Approach to Image Processing

The traditional approach to digital image processing is applied in two domains: the spatial and frequency domain . In the spatial domain the image is represented as a two-dimensional function $f(x, y)$, where x and y are spatial coordinates. The value of the function f for any pair of coordinates is the intensity of the grey in that point. The term spatial domain relates to the image itself and the approaches in this domain are based on direct manipulation with the image pixels. The processes in the spatial domain are denoted by the expression:

$$g(x,y) = T[f(x,y)]$$

where $f(x,y)$ is the input image, $g(x,y)$ processed image, and T is the operator of the function f, defined by the neighbouring pixels (x,y).

Digital image processing in the frequency domain is carried out by using the Fourier's transformation to convert the image into the space of frequency where the processing is being done, and then converting the image back into the spatial domain through the inverse Fourier's transformation. In the frequency domain it is simpler to notice some important characteristics of the image, such as edge detection. Irregularities can also be detected. In the case of a noisy image, these irregularities can be easily removed by observing the image in the frequency domain and filtering out the high frequencies. Often one and the same thing can be done in both the frequency domain and the spatial domain.

As opposed to the traditional image processing this paper proposes a method that uses analogies as well as the user-led filter creation. The idea is to enable a non-specialist user to develop his or her own filter palette and be independent of filter producers. If it is necessary to process a large number of images and to get from them processed images that include application of a large number of filters, using just any software for standard image processing would be extremely difficult because each image would need to be processed separately. The advantage of using analogies is that once learnt filter (which can be a combination of many applied filters) can be applied on any number of newly selected images. Images produced by traditional approach to image processing were used as the training data to test the quality and demonstrate the potential of the proposed solution.

5 Implementation and Methodology of the Proposed Solution

Analogy is the basic reasoning process. People often use analogies without usually even being conscious about it. They use them for predicting or resolving different kinds of problems. Generalisation based on a set of familiar examples is the central problem of machine learning. For this reason, the goal of artificial intelligence from the inception of this field is to develop a system that could reason by using analogies.

The basic premise of this project comes from the essential idea of analogy presented by Hertzmann [1]. The idea of this work is to use neural networks for learning analogies. The reason for using neural networks in this project is the very fact that they have proved to be a good tool for learning and solving different kinds of problems.

5.1 Learning Technique and Training Data

As a program input two images are taken: the original image *A* (*model*) and the filtered image *A'* (*master*). Neural networks learn *model – master* mapping, and after a given number of generations, the achieved result is displayed in the *apprentice* window. Displaying of generations is enabled so that the learning process can be visually followed. As a learning technique the method of supervised learning is used, since both the input data and the expected results are known.

In view of resolving this problem the program offers an option of choosing a training data set. Two options are enabled: a choice of a number of a random set of pixels from the input images (*model* and *master*) or the manual selection of image surfaces to be used as training data.

The colours on and around any given pixel *p* of the image *model* correspond to the colours on and around that same pixel *p* of the image *master*, the filter that needs to be (Figure 1):

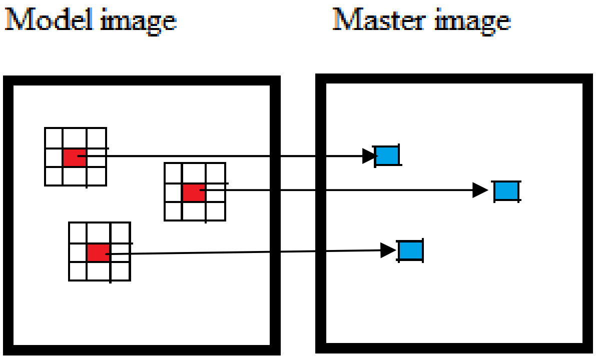


Figure 1

Red colour on the image *model* marks image pixels chosen as input data, while the blue colour on the image *master* marks the expected results. Figure 1 shows 3 pixels as input data.

A problem with neural networks during the learning of a given filter can be a too small or wrongly selected training data set. That could be the training data that does not contain specific information for achieving the desired result, thus it is necessary to select the training data carefully.

The learning procedure takes the achieved and the desired results and compares them, after which the network connection weights are corrected. The same set of data is processed several times during the network training until the connection weights are improved. The procedure is repeated until the desired similarity- *fitness* condition is satisfied. The program can be also be stopped the moment the user is satisfied with the visual similarity. After the end of the run, the program remembers the neural network that can be applied on a newly selected image *B* so that the learnt effect can be applied to produce an image *B'*.

5.2 Architecture

The proposed software uses Encog3 neural networks. During the implementation of the software it is necessary to make certain decisions in the sense of selecting the architecture, activational function and the algorithm for training the neural networks.

Feed-forward neural networks have been chosen (Figure 2), which consist of the input, the hidden and the output layer of neurons, where the upper layers do not supply feedback to the lower ones but the information flows in one direction only – from the input to the output.

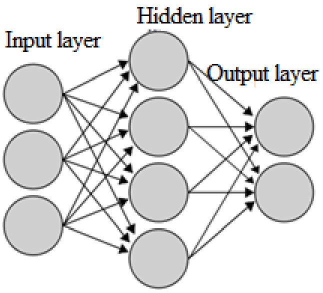


Figure 2

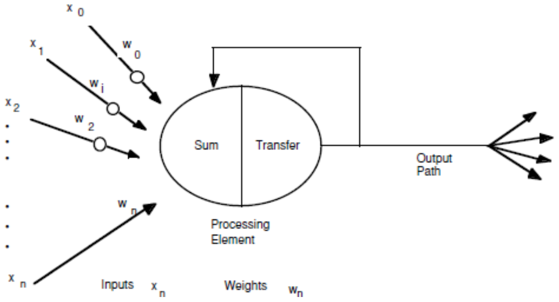


Figure 3

Grouping of the neurons into layers, the connections between the layers, sum functions and activation function define the functioning of the neural networks. The network is structured in such a way that each neuron of the hidden layer receives the signals from all the neurons from the previous layer.

In addition to the pixels chosen as training data, the size of the retina is also given as a parameter which influences the number of neurons in the input and the hidden layers. This parameter has value 3 on the image *model* shown on Figure 1, so the size of the retina is 3x3. For the retina 3x3 the number of neurons in the input layer for a colour image is 3*9, because a colour image has 3 bands, with one component for red, one for green and one for blue. In that case, the number of the output neurons is 3. For a *grey-scale* image and retina 3x3, the number of the input neurons would be1*9, while the number of the output neurons would be 1. An RGB colour model has been used.

Both the number of layers and the number of elements in each layer have impact on the efficacy of the neural network. The program offers an option to choose the number of hidden layers, as well as the parameter of value for each hidden layer which shows how many times is the number of neurons in the hidden layer larger than the number of neurons in the input layer.

Each neuron (Figure 3) consists of four basic functions: receipt of input, input processing, converting of the processed input into the output and synapse connections with other neurons. Network inputs are represented with the mathematical symbol *x(n)*, and each input information is multiplied by its weight *w(n)*. The results are added up (sum function):

$$Sum = \Sigma w_i x_i$$

and forwarded to the activation function which generates the result and then forwards it as the output.

5.3 Activation function

The activation function is used to scale the output data from the layers. The reason for using the activation function is the possibility for the neural network to learn non-linear function. If this function did not exist the neurons from the hidden layers would not allow for more possibilities than the ordinary network that only consists of the input and the output – the perceptron. For that reason a nonlinear activation function is used on the layers of neurons that are about to output.

For the activation function a sigmoid function has been chosen. The reason for using this function is the expected positive value as the output. The output of the sigmoid function is the value from 0 to 1.

$$f(x) = 1/(1+e^{-x})$$

This value is at the end of an iteration converted into the values from 0 to 255 so that a program error is represented in the levels of grey colour and so that the parameter of tolerated error is more intuitive to the software user.

5.4 Training algorithm

Resilient propagation has been chosen for the learning algorithm, as suggested by the authors of the Encog3 neural networks. The advantage of this algorithm is that it does not require setting of the parameters such as the learning rate or the momentum value. That is good because it is sometimes very hard to determine these parameters in search of an optimal solution. The algorithm can be easily swapped in the program, but the examples tested and presented here use the resilient propagation algorithm.

Training is the way in which neural networks learn how to improve the weights of synaptic connections in order to achieve the desired result. A training set of input data and the ideal output for each input is forwarded to the training algorithm. The resilient propagation training algorithm goes through series of iterations and after each one it attempts to lower the error rate for a certain degree. The error rate represents a per cent of difference between the real output and the output produced by the training algorithm. A gradient of error is calculated for each connection in the neural network. There is no global update parameter; rather for each value of the weight matrix there is a separate delta value. These values are at first randomly initiated at very low values. They are updated in accordance with delta values after each iteration. The magnitude of the gradient is used to determine how delta values should change. In this way each weight matrix can be individually trained. (algorithm [12])

6 Main functions and interface

The interface is very simple. The communication between the user and the program takes place in the interactive window *"Talk to me"* in which Ruby code can be written. By recalling the functions which will be described in this chapter the software enables the learning of the chosen filter, the creation of own filter and the application of the learnt filter on the newly selected image. Once learnt, filter is saved as a neural network in the .eg format. The software supports different image formats such as .jpg, .gif, .png.

6.1 The Process of Learning the Chosen Filter

With the running of the program the original image *model* is loaded first, after which is loaded the chosen filter image – *master*. Each image appears in the separate window. After each 500 epochs a new image is displayed in the *apprentice* window, i.e. the attempts of the neural network to learn the *master* set on the basis of the *model* set. In case the program is interrupted or the set similarity –fitness is satisfied, the result is displayed in the *apprentice* window. As already mentioned, the *"Talk to me"* window is an interactive window where the Ruby code can be written. By opening the *model* and *master* images all the parameters whose values can be changed (and were initially set at assumed values) are displayed in this window.

$tolerrable_err is a global variable whose value is initially set at 0.1. This variable represents the permitted error and is the condition for stopping the program. This value can be changed (e.g. to 0.01) with the command: *$tolerrable_err = 0.01*. The value of the error is in the interval je between 0 and 255 in the levels of grey. The run can also be stopped before the error becomes smaller of equal to the permitted one by pressing on the command *Ctrl+Alt+q*.

$taining_set_size is the global variant representing the number of the training data. It is initially set at 5000, which is the number of randomly chosen image pixels. This is one method of choosing data for the purposes of training. The other method for choosing data is not arbitrary. By clicking on a pixel from the *model* or *master* image, there appears a square of the size set through the global variable *$domain_width*. This value is initially set at 38, which represents a square the size of 38x38

pixels. In case this method of setting the training data is chosen, the size of the variable *$training_set_size* is ignored.

The global variable *$hidden_layers* represents a sequence of hidden layer quotients. The quotient number determines the number of hidden layers, while the number of neurons in this layer is equal to the number of neurons in the input layer multiplied by the quotient itself. One hidden layer was initially set at value 1.5. Recalling the *ma_retina* function we can change the size of the retina, i.e. the number of neurons in the input layer.

If the image and the filter are in colour, by recalling the *ma_grey* function the image is converted into a *grey-scale* image. Conversely, if the image is *grey-scale*, by recalling the *ma_col* function, the image is treated as though it is in colour.

The *ma_run* function starts the learning process by clicking the *OK* button. After the achieved similarity, the result can be saved by recalling the *ma_save* function. What is saved is the neural network of the learnt filter which can be applied to any other selected image any number of times.

By recalling the *ma_init* function *model* and *master* images are opened again for new learning of the filter, or the image is open on which we wish to apply the already learnt and memorised filter. *ma_info* function is used to collect information about the parameters whose values can be changed.

6.2 The Process of Creating Own Filter

It is important to bear in mind that graphic designers, artists and other people whose job depends on production of original images nevertheless use the same image processing programs. The proposed software yields interesting results regarding the production of filters that the user creates him/herself. On the one hand, the user would have to spend some time creating his or her own filter, which seems discouraging if it is not taken into account that the once learned filter can be reproduced an infinite number of times on other images and in real time. In this way the user could create his or her own personal filter palette and also work with the same while making a product that would distinguish him or her from the competition. The difference in respect to the option of learning the chosen filter is in the fact that the number of the *apprentice* windows is greater, i.e. the number of the neural networks that are trained in view of learning the given filter. The way in which evolution is simulated here is the selected small number of training data. Using this method we can simulate variations, i.e. the diversity of images produced after each displayed generation. After each displayed generation the user can swap the currently chosen master image with one of the images produced during the previous generation and create in that way a filter directed by his or her own aesthetics.

Initially, the number of *apprentice* windows is 6, but that number can also be lowered by switching off some of the windows. By recalling the *ma_make* function, another *apprentice* window is opened. The program stops by recalling the *ma_stop* function. We stop the program the moment we like one of the images in the *apprentice* windows more than the current *master* image. By clicking on the chosen image that image becomes the *master* and in this way the process can be continued by recalling the *ma_run* function, after which is produced a new generation of images. This process is repeated until we get the image, i.e. the filter that we would like to save. With each selection of a new image the neural network of the learnt filter can be saved.

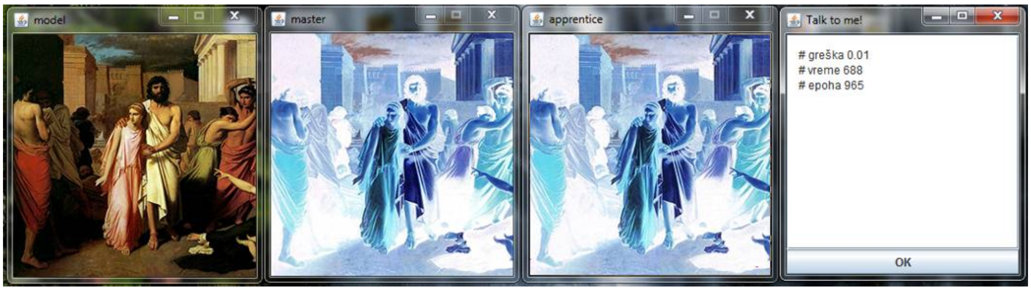
6.3 The Application of the Learnt Filter to the Newly Selected Image

Once the learnt filter can be applied any number of times to any other image. It is saved in .eg format. The opening of the file is carried out by recalling the *ma_open* function, while by recalling the *ma_use* function the learnt neural network is applied to the newly selected image.

7 Experiments

In this section the results achieved by using the proposed software are presented. For each selected filter the result of learning it will be shown first, followed by the results of application of the learnt filter to newly selected images. One example will serve to illustrate a model of the user-led creation of a filter.

The simplest filters are the operations on one pixel. They are defined as a function carried out on each pixel of the image, independently from the other pixels of that same image. Some of the point processes are: converting a colour image into a grey-scale one, enlargement and reduction of contrast or brightness, the image negative and the threshold. As an example of the point process two filters are selected: the negative and the threshold. The negative filter is particularly suited for improving the white and grey details surrounded by dark regions of the image, especially when black surfaces predominate. Figure 4 shows the result of learning the Photoshop's invert filter:



greška 0.01
vreme 688
epoha 965

Figure 4: oedipus_col.png is taken as the *model* image, and oedipus_col_invert.png as the *master* image

The received result is shown in the *apprentice* window. The error of the received result is 0.01, and the number of iterations is 965. The time needed for learning of the desired filter was 688 seconds. For the size of the retina 3x3 retina was set.

Figure 5 illustrates the application of the learnt filter *odipus_col_invert.eg* on newly selected images:

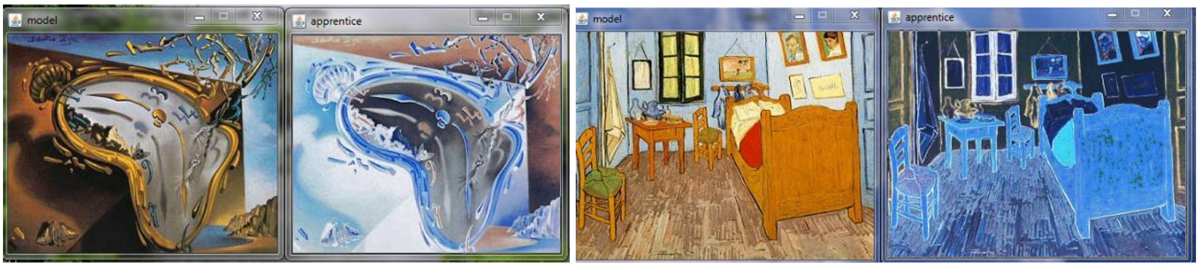
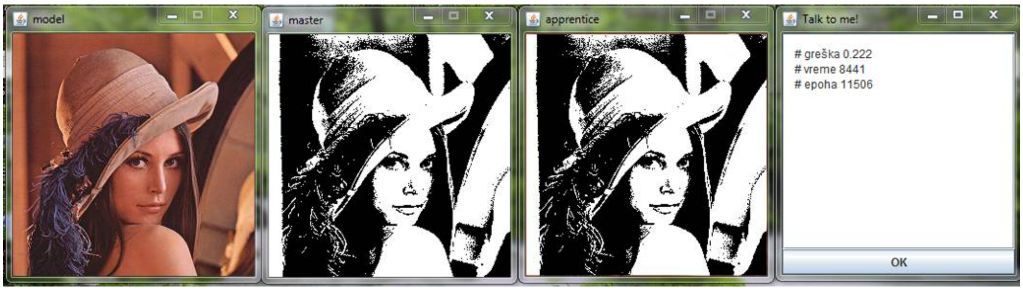


Figure 5: (a) watch.jpg (b) room_Arl.jpg

The segmentation of the image relates to the procedure of dividing the image into regions with similar attributes. Of the attributes brightness is used most often in the case of greyscale images, whereas the colour is used for the colour images. During the process of segmentation different characteristics can be used, such as texture measures and the edges. During the image analysis segmentation is one of the first and most important steps. Theoretically, there is no parameter for quantitative estimate of how well performed is a certain segmentation process. One of the methods for segmentation is determining of the brightness threshold. Figure 6 shows the result of learning of the Photoshop's threshold filter. For the size of the retina 3x3 retina was set:



greška 0.222
vreme 8441
epoha 11506

Figure 6: lena_col.png is taken as the *model* image, and lena_threshold90.png as the *master* image

The received result is shown in the *apprentice* window. The error of the received result is 0.222, and the number of iterations is 11506. The time needed for learning of the desired filter was 8441 seconds. Figure 7 shows the application of the filter *lena_col_threshold90.eg* to newly selected images:

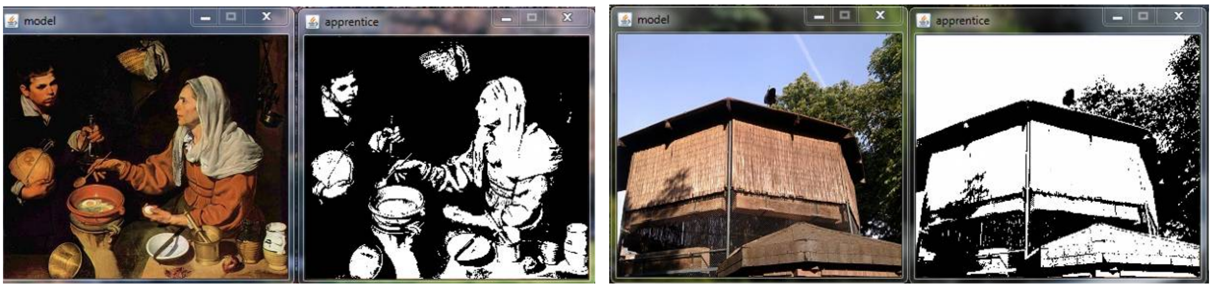
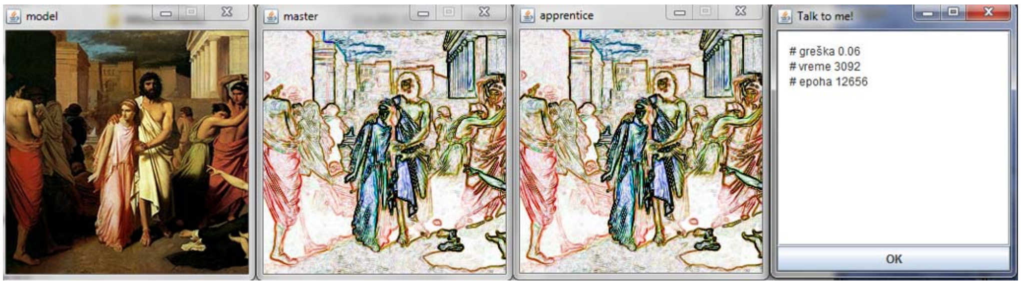


Figure 7: (a) velaskez.jpg (b)peacock.jpg

The main goal of image detection is deriving the important information from the image which can enable us to perform the computer interpretation and image analysis. The edges correspond to significant changes of image intensity. Intuitively, the edge is a set of connected pixels located on the border between two regions. There are many filters for edge detection and some of them are: Roberts cross, Sobel, Prewitt, Kirsh, Canny and the Laplacian filter. Figure 8 shows learning of the Photoshop's edge detection filter:



```
# greška 0.06
# vreme 3092
# epoha 12656
```

Figure 8: oedipus_col.png is taken as the *model* image, and oedipus_edge.png as the *master* image

The received result is shown in the *apprentice* window. The error of the received result is 0.06, and the number of iterations is 12656. The time needed for learning of the desired filter was 3092 seconds. The retina used is 5x5. Figure 9 shows the application of the *oedipus_col_edge_detection.eg* to newly selected images.

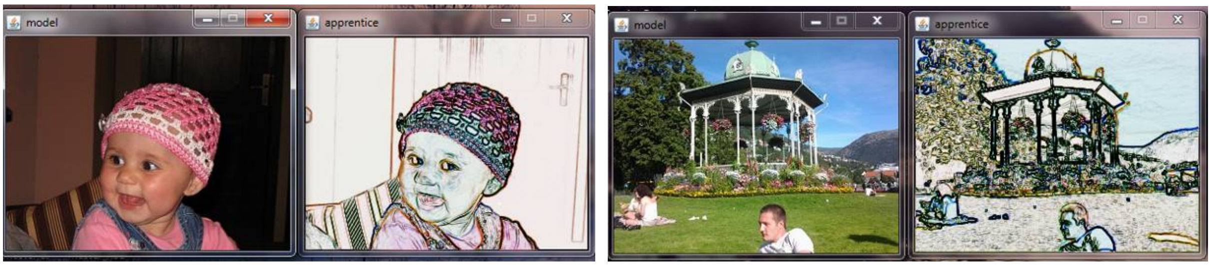
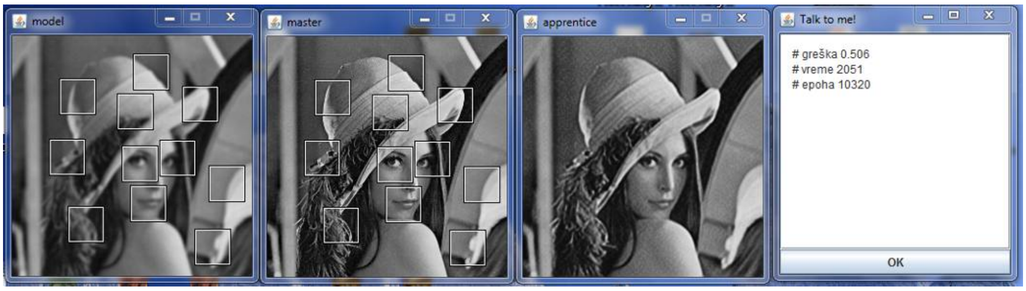


Figure 9: (a) nena.jpg (b) park.jpg

The interpretation of the image depends on its sharpness, the possibility of isolating the information contained in the image. The sharper images show more detail. However, it is not easy to define what the sharp image is. It could be defined as an image that looks like a natural scene, but what appears natural to the human eye is difficult to calculate. Usually, a blurry image is defined as one that has lost the high frequency information. The images produced by using a photo camera are often blurry. Instead of the individual processing of each photograph it would be much easier to use a filter that would improve the quality of the poorly produced/damaged photograph. Figure 10 shows learning of the deblur filter. Retina 5x5 was used. The training data has not been arbitrarily chosen; rather, the image surfaces were selected for the purpose:



greška 0.506
vreme 2051
epoha 10320

Figure 10: lena_blur.png is taken as the *model* image, and lena.png as the *master* image

The received result is shown in the *apprentice* window. The error of the received result is 0.506, and the number of iterations is 10320. The time needed for learning of the desired filter was 2051 seconds. Figure 11 illustrates the application of the learnt filter *lena_mix.eg* to newly selected images.

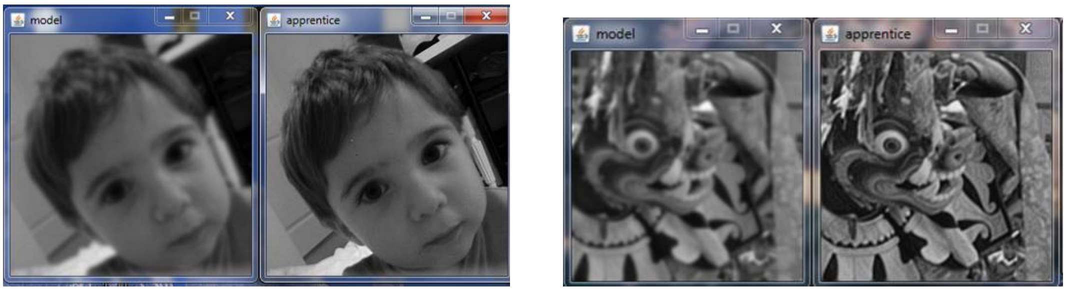
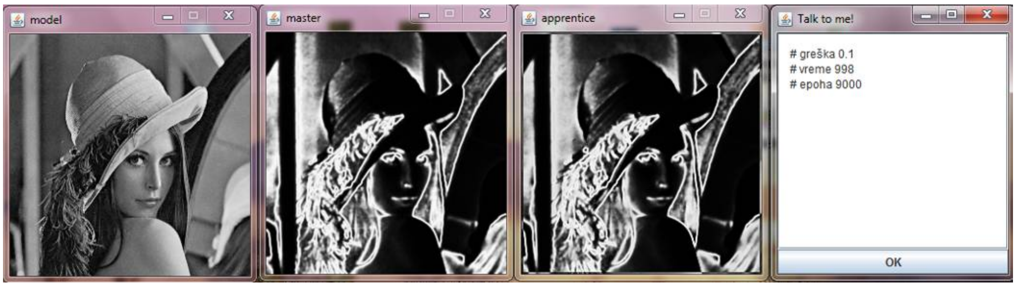


Figure 11: (a) mile.jpg (b) dragon.jpg

The advantage of using the proposed software is the possibility to learn a filter which is a combined application on the image of not only one but a larger number of filters. lena.png is taken as an example on which many filters have been applied in Photoshop: solarize, brightness, contrast i blur filter Figure 12:



greška 0.1
vreme 998
epoha 9000

Figure 12: lena.png is taken as the *model* image, and lena_mix.png as the *master* image

The received result is shown in the *apprentice* window. The error of the received result is 0.1, and the number of iterations is 9000. The time needed for learning of the desired filter was 998 seconds. Figure 13 shows application of the learnt filter *lena_mix.eg* to newly selected images:

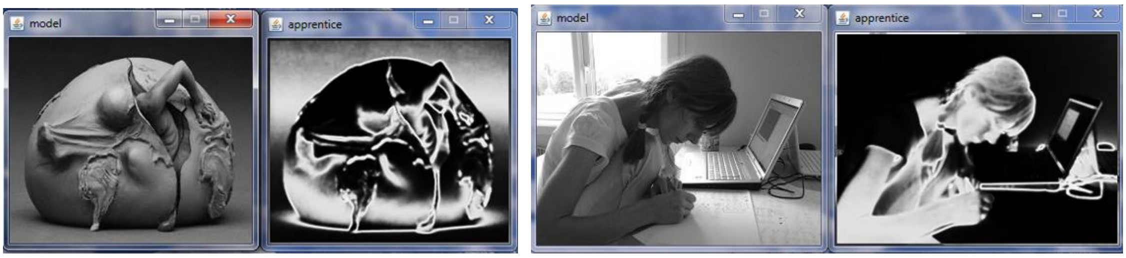
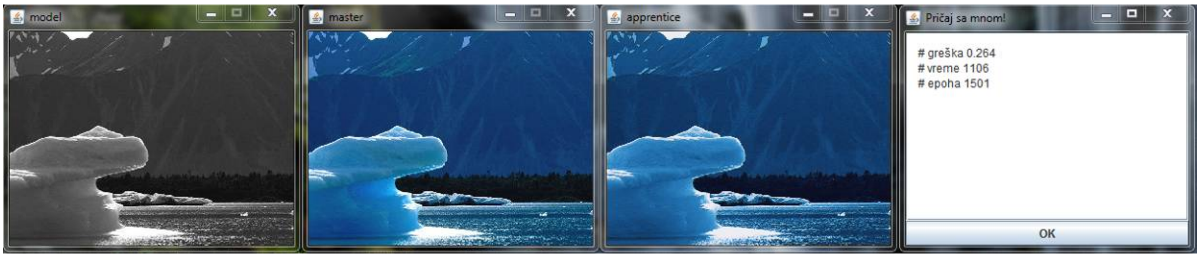


Figure13: (a)birth.jpg (b)me.jpg

Grey-scale colouring is a difficult problem. It is very challenging to colour an image to if some information about the image is not already available. The same objects can be coloured differently, for instance a blue and green ball which have the same shape and structure and are made from the same material. Colouring natural objects is also problematic. The leaves are during the spring green and brown in the autumn. Many solutions for this problem consist of deriving the information about the colour from the user him/herself. Figure 14 shows learning of this filter:



greška 0.264
vreme 1106
epoha 1501

Figure 14: ice_grey.jpg is taken as the *model* image, and ice_color.jpg as the *master* image

Figure 15 (a) shows the application of the learnt filter on a newly selected image while figure 15 (b) shows the original image. This appears to be a good solution; however, the problem occurs when the expected image does not contain the colour nuances of the trained pair.

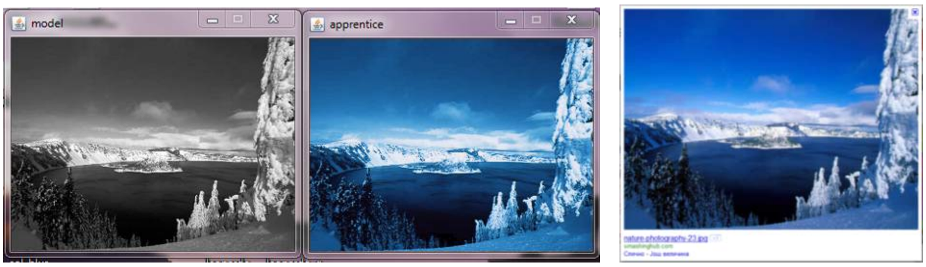


Figure 15: (a) winter_grey.jpg (b) original one

For the creation of own, personally led filter, the Photoshop's *poster edges* is taken as the starting point. Figure 16 only shows the images that are chosen as master images after a discontinued generation. The last image was produced after the tenth discontinued generation. The process can be continued. The neural network of each shown image is saved as a separate .eg file, so that each of these learnt filters can be applied to a newly selected image. The training set size is 150.

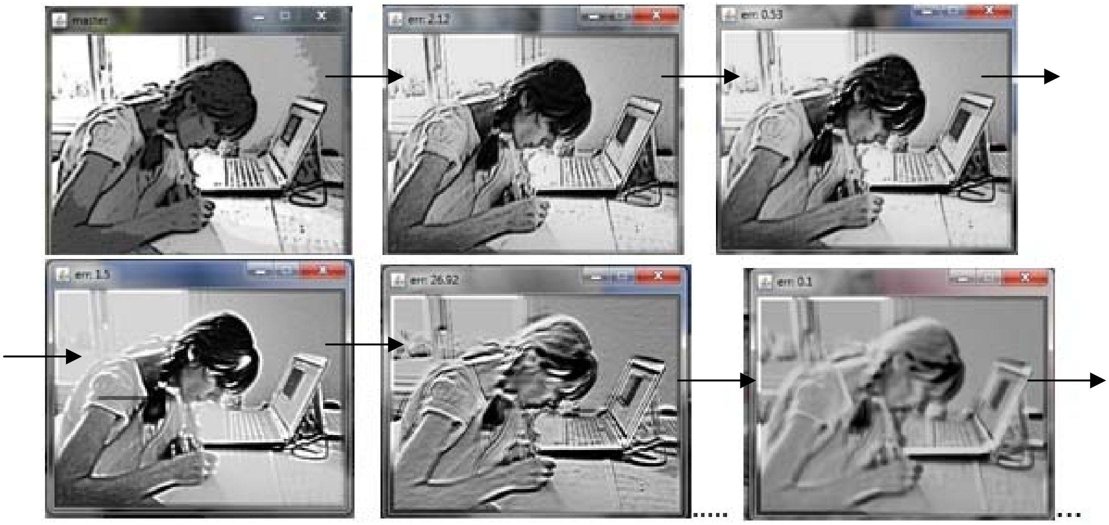


Figure 16: me.jpg

8 Conclusion

It is obvious that we cannot expect that the software which solves the problem of image analogy performs perfectly in the sense of learning all possible filters, especially because only one pair of images is used for training. It is still surprising that many kinds of filters can be learnt using this principle. The proposed solution for image analogy using neural networks has proved to be a good method for learning different kinds of filters, such as: threshold, conversion of a colour image into a grey-scale one, image inversion, enlargement and reduction of contrast and brightness, detecting, blurring and sharpening of edges. The software also supports learning of the filters that combine several applied image transformations. This is very important if processing of a large number of images is required. The software in addition solves the problem of getting an image of a higher resolution, which is particularly useful if it is required that a part of the image is enlarged, or if the image is blurred, which often happens when a photo camera is used. The software enables natural image transformations instead of the selection of different filters and their adjustments. The user can simply choose the desired effect and reproduce the same on a new image. This taken into account, the filters do not have to be individually invented or programed separately. Ideally, the same mechanism could be used to produce a wide spectrum of effects. In addition to learning a chosen filter, the software enables a generation of own filters. The proposed software also has its limitations: the filters that include distortion and those that depend on the context of the image itself cannot be learnt by using the proposed solution.

References

[1] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless and D. H. Salesin. Image Analogies. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'01).*, 327-340, 2001.

[2] A. Hertzmann, N. Oliver, B. Curless and S. M. Seitz. Curve Amalogies. *In Proceedings of the 13th Eurographics Workshop on Rendering (EGRW '02).*, 26-28, 2002.

[3] W. T. Freeman, T.R. Jones, and E.C. Pasztor. Example-based superresolution. *IEEE Computer Graphics and Application.,* 22(2): 56-65, 2002.

[4] N. Joshi, W. Matusik, E. H. Adelson, and D. J. Kriegman. Personal photo enhancement using example images. *ACM Transactions Graph*ics., 29(2): 1-15, 2010.

[5] L. Cheng, S. V. N. Vistwanathan, X. Zhang. Consistent Image Analogies using Semi-supervised Learning, IEEE *Computer Vision and Pattern Recognition.*, 1-8, 2008.

[6] M. Li, F. Wang, X. Liu and H. Jin. Super-Resolution Rendering for Digital Earth Applications, *Proc. SPIE 7840, 784005.*,2009

[7] A. Levin, D. Lischinski and Y. Weiss. Colorization using optimization. ACM *Trasactions On Graphics*., 23(3), 2004.

[8] R. Irony, C. Cohen-Or and D. Lischinski. Colorization by Example. *In Proceedings Of Eurographics Symposium on Rendering.,* 201-210, 2005.

[9] J. B. Lackey and Michael D. Colagrosso, Supervised segmentation of visible human data with image analogies, 2004

[10] V. Bychkovsky, S. Paris, E. Chan, and F. Durand. Learning Photographic Global Tonal Adjustment with a Database of Input / Output Image Pairs. *IEEE Computer Vision and Pattern Recognition.,* 2011.

[11] J. Heaton. *Getting Started whith Encog3 in Java.* Heaton Research, Chesterfield, MO, 2011.

[12] C. Igel, M Toussaint and W. Weishui. Rprop Using the Natural Gradient. *Trands and applications in constructive approximation.*, 151: 259-272, 2005.

Jelena Tasic
University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: jelenatasicsd@yahoo.com

prof. Milan Tuba
University Megatrend of Belgrade
Faculty of Computer Science
Bulevar Umetnosti 29, N. Beograd
SERBIA
E-mail: tuba2@ieee.org

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Routemaster – A Windows Phone 7 Travel Application

Tecsa Simona, Stancu Mihai, Pirvu Doru
Teacher Coordinator: Hunyadi Daniel

Abstract

This paper explores the potential of a user friendly trip planning smartphone app as a means of increasing public transit usage. We present the problems we wish to solve and the benefits the application has to offer, as well as technical details regarding its design and implementation. In the end we present our results and plans for future development.

1 Introduction

Transportation represents a very important part of our life. It has been responsible for the development of civilizations and it is a non-separable part of any society.

1.1 Problem statement

Unfortunately transportation accounted for more than 23% of total worldwide CO_2 emissions in 2005, of which roughly 73% were generated by road transport [1].

In the European Union, in thirty years from 1970 to 2000 the modal share of the car has increased with 4.5% from 73.8% to 78.3% while the public transport modal share has decreased with 8.7% from 24.6% to 15.9% [2].

In 2008, Americans took 10.7 billion trips on public transportation [8]. According to the American Public Transportation Association (APTA), public transportation in the United States saves approximately 1.4 billion gallons of gasoline and about 1.5 million tons of carbon dioxide annually [3]. Yet only 14 million Americans use public transportation daily while 88 percent of all trips in the United States are made by car—and many of those cars carry only one person.

The relative reduction of public transport ridership is the result of major sociological and politico-economic changes. It corresponds to changes in lifestyles, mainly characterised by tighter time constraints.

The management of mobility has never been as difficult and complex as today. The challenge is therefore to implement flexible, innovative solutions to meet the needs of public transport.

In 2009, 51,000,000 automobiles have been sold worldwide [4]. The reason for this is that marketing efforts by public transportation systems are vastly outspent by those of the major automobile companies, which spend billions each year to attract new customers. Worldwide advertising and marketing efforts among the automobile sector as a whole total $21 billion [10]. All these investments aimed at attracting new customers help increase car sales, but also boost congestion, carbon dioxide emissions, and air pollution, while working against broader public transportation use and more sustainable urban transportation systems.

1.2 Possible solutions

To fight these trends, public transportation systems must not forget about branding, marketing, advertising, user education and using smart, creative, cost-efficient campaigns targeted at increasing and maintaining ridership. Transit agencies must focus their efforts on how to attract new users that currently use private transport, such as cars and motorcycles; retain existing public transport users who might feel compelled to buy a private vehicle; and secure political and financial support from government officials.

The focus should be on brand and identity; user education, information systems, and feedback tools, including online engagement; marketing campaigns; public relations; and internal and external communications. While public transport users determine whether to use a system based on its reliability, frequent service, safety and cleanliness, service hours, and costs and structures, public transport systems still need to do branding, marketing, and communications to increase and maintain ridership.

To create a successful brand, then, a public transport system should start by defining its core values. Most public transport systems strive for a brand that clearly presents their services as modern, efficient, rapid, reliable, convenient, comfortable and safe.

User education is critical: Citizens who consider riding public transport, particularly those who have the option to drive, can be deterred by the unfamiliarity of a system —where it goes, the fare collection, the boarding process — basically every aspect of using it. Agencies can overcome this hurdle with extensive user education, especially in the form of information kiosks, online and mobile content.

1.3 Our solution

In 2010 1.6 billion mobile phones have been sold [15] and, in profit share, worldwide smartphones now far exceed the share of non-smartphones. A report published in september 2010 by the UN indicates that 67% of the world's population, or two-thirds total, are mobile subscribers [5].

From 2G to 4G, mobile communication technology has achieved rapid progress in just a few years, paving the way for continued innovation and developments. With their rich features and capabilities, smartphones have become fertile ground for the growth of mobile apps. While all categories of apps are more popular on smartphones than on feature phones, the difference is more pronounced in categories such as Maps/Navigation, where more computing power, larger screens and touch interfaces deliver a more satisfying experience.

The first thing potential riders want to know before deciding to use public transport is whether it can get them to their desired destination. A web app that features cleanly designed maps and schedules can provide this information to passengers before they even leave their homes. It can also tell them expected travel times, potential delays or any other information that would affect their commute. All these things make public transport more user-friendly, reliable and accessible, increasing the chance that discretionary riders become loyal customers.

The solution we propose consists of a mobile application that targets people of all ages who use public transportation systems, specifically road and rail transport. Our app will help people find the route that suits them best, and much like a GPS, it will provide instructions on how to get to their destination. The app will also compute money and carbon emissions savings to show how its usage impacts the environment and to increase environmental awareness.

Even though many transit agencies still don't have comprehensive online strategies, and have not opened their data, we are developing an innovative way to share public transportation information, using data collected through social media. Currently this is a work in progress.

We also recognize the effectiveness of educating children about public transport to boost ridership. The basic idea is that children will be likely to share what they've learned with their friends and family. This is why appealing to younger audiences was a major priority to us.

Finally, the app will be free, open-source, open to user contribution, and most importantly, it will be very simple, intuitive and fun to use.

2 Design

Our primary goal, as stated above, is to make our application extremely easy to use. User experience is our main focus, and we take a rigorous and systematic approach to the way we display information to the user and design interaction.

2.1 User Experience

Design rules have governed the development of Routemaster and helped produce an efficient, usable, intuitive, and high-quality user interface.

One of the guidelines that we followed was that at any point, our application must allow the user to answer five fundamental questions about the current means of transport:

- What is it?
- Where does it go?
- When does it start?
- How do I get from here to there?
- Why should I ride it?

When designing the UI, we permanently asked ourselves the following questions:

- What information will the user need at key decision points in his/her journey?
- Can children and elderly see and understand the information?
- Is the most basic information conveyed in a way that reduces or eliminates language and literacy barriers?

2.2 Use Cases

We tried to create an efficient and elegant interface, and in order to do this, at every step of the development process, from paper sketch, to computer mockup, graphics editing, even coding and testing, we asked the question: Is this feature really necessary? If at any point the answer to that question was “No”, then we would discard it.

Interaction flows smoothly and the user always knows exactly the context of the information displayed. The diagram below shows the general use case:

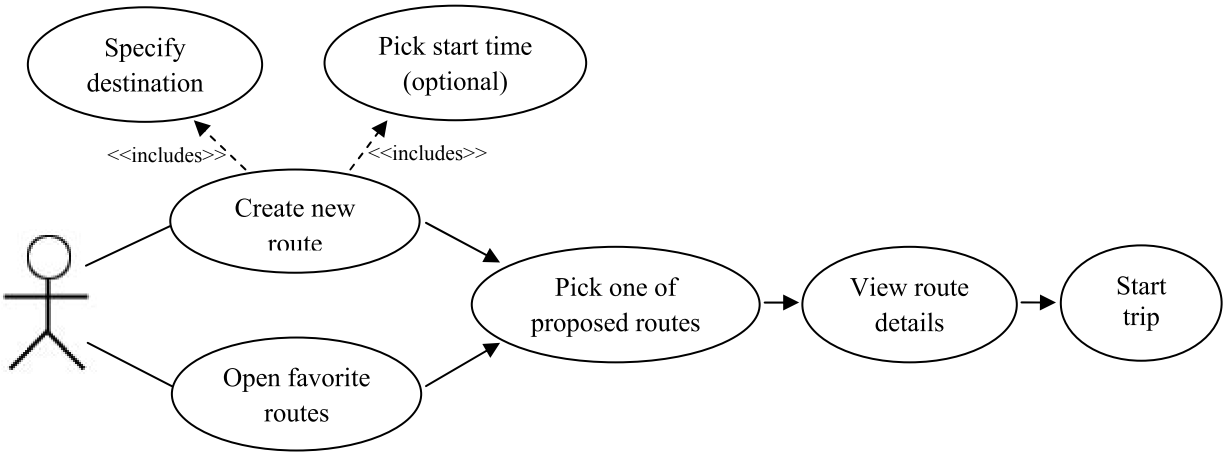


Figure 1 General use case of Routemaster

The application looks and feels integrated into the Windows Phone platform and it adheres to the design principles and strategies proposed by Microsoft [17]. We strived to keep each screen as focused as possible and eliminate any extra clutter. We present to the user only what is absolutely necessary. The app is extremely intuitive, such that it requires no training and no user guide is necessary to be used.

3 Implementation

The application is written in C#, using the Windows Phone 7 SDK. We used Visual Studio 2010 for coding and Microsoft Expression Blend for creating mock-ups of our screens.

Below we present the main technologies that we used to develop our application.

3.1 Silverlight

Silverlight is a powerful development tool for creating engaging, interactive user experiences for Web and mobile applications. Silverlight is a free plug-in, powered by the .NET framework and compatible with multiple browsers, devices and operating systems.

Currently Silverlight for Windows Phone is the only solution to build native Windows Phone apps. Fortunately, it provides a very large number of controls that we can use, including the Bing Maps Silverlight Control, which is 100% compatible with Bing Maps, our primary source of geographic information.

3.2 Bing Maps

Microsoft® Bing™ Maps Platform provides the power behind the online mapping service Bing Maps that enables users to search, discover, explore, plan, and share information about specific locations. The Bing Maps Platform, which includes an AJAX map control, a Silverlight map control, as well as Bing Maps REST Services, Bing Spatial Data Services and Bing Maps SOAP Services provides unique opportunities for developers to incorporate both location and local search features into their apps [18].

Key features of the Bing Maps Platform include [19]:

- Photo-based images with features such as Streetside and 45 degree oblique "bird's eye" views (nominally including 4 views at 90 degree viewpoint increments) that present data in context while simplifying orientation and navigation.
- The ability to overlay standard or custom data points and layers with different themes.
- Rich developer support options available.
- Large set of APIs available upon which developers can build applications.
- Bing Transit, which provides alternative routes to a specified destination, based on information provided by transit agencies in the GTFS Format

3.3 GTFS Format

Developed by Google, the General Transit Feed Specification (GTFS) defines a common format for public transportation schedules and associated geographic information. GTFS "feeds" allow public transit agencies to publish their transit data and developers to write applications that consume that data in an interoperable way [16].

The example below shows comma-delimited data samples for the "agency.txt" GTFS file:

```
agency_id, agency_name, agency_url, agency_timezone, agency_phone, agency_lang
TurSib, TurSib, http://www.tursib.ro/, 301, 0269-426100, ro
```

Among the benefits of using this format we mention: feeds are easy to create and edit; feeds are easy to parse; feeds are easily extendable through the addition of extra columns and files that are ignored by official parsers and validators; Changes to the spec are backwards-compatible

4 Future developments

We put a lot of effort in developing our application, but there are a series of enhancements that we plan to include in our next release. Among these we mention:

- Including speech recognition capabilities for the Romanian language
- Adding car-pooling capabilities
- Porting the application to other devices

5 Conclusion

Public transportation provides personal mobility, reduces congestion, provides economic opportunities, saves fuel and money, reduces carbon footprint and improves air quality, yet according to CNN, only 5% of the world population uses public transport regularly.

We suggest that a key challenge to expanding transit ridership is to encourage people who have not used public transportation to develop familiarity with it, and to increase current riders' confidence in public transit. Various factors contribute to the hesitancy of potential new users to try-out public transit options, including uncertainty about routes, schedules, and other basic utilization information.

In these conditions, automated information systems provide the best means of accessing such data. In particular, the development of next generation smartphones offer the attraction of obtaining information about the transit system through more familiar and easy to use interfaces.

We showed how our application can help make a difference and increase environmental awareness for people of all ages and we provided technical details to better support our claims.

It is vital that we figure out new ways to make the transition to public transport worthwhile in the eyes of society. If we further improve technologies such as Routemaster that make lives more convenient, and increase societal benefits, we will be able to accelerate the modal shift toward public transportation and put an end to our car-dependent society.

References

[1] http://www.jama-english.jp/publications/2008_CO2_RoadTransport.pdf
[2] http://old.enea.it/com/ingl/New_ingl/publications/editions/pdf/7_Demand_Transport_Services.pdf
[3] http://environment.about.com/od/greenlivingdesign/a/public_transit.htm
[4] http://worldmeter.com/cars
[5] http://www.mobilemarketingwatch.com/67-of-the-worlds-population-are-mobile-subscribers-5541/
[6] http://www.email-marketing-reports.com/wireless-mobile/smartphone-statistics.htm
[7] http://policy.rutgers.edu/faculty/pucher/Sustainable.pdf
[8] http://www.thewavetransit.com/files/Facts%20on%20Public%20Transportation.pdf
[9] http://www.mtagogreen.com/pubtrans.html
[10] http://www.embarq.org/sites/default/files/EMB2011_From_Here_to_There_web.pdf
[11] http://www.squidoo.com/mobilewebdevelopment
[12] http://policy.rutgers.edu/faculty/pucher/Sustainable.pdf
[13] http://www.jama-english.jp/publications/2008_CO2_RoadTransport.pdf
[14] http://blog.nielsen.com/nielsenwire/wp-content/uploads/2010/09/NielsenMobileAppsWhitepaper.pdf
[15] http://visagemobile.com/2011/02/09/gartner-1-6-billion-mobile-device-sales-in-2010
[16] https://developers.google.com/transit/gtfs/
[17] http://msdn.microsoft.com/en-us/library/hh202895(v=vs.92).aspx
[18] http://msdn.microsoft.com/en-us/library/bb545001.aspx
[19] http://en.wikipedia.org/wiki/Bing_Maps_Platform
[20] http://personalmoneynetwork.com/moneyblog/2012/03/12/public-transportation-usage/
[21] *Charles Petzold*, Programming Windows Phone 7, Microsoft Press, 2010

TECSA Simona
"Lucian Blaga" University of Sibiu
Computer Science
Dr. I. Rațiu Street No.7-9, Sibiu, 550012
ROMANIA
E-mail: tecsa_simona@yahoo.com

STANCU Mihai
"Lucian Blaga" University of Sibiu
Computer Science
Dr. I. Rațiu Street No.7-9, Sibiu, 550012
ROMANIA
E-mail: stancu_meehigh@yahoo.com

PIRVU Doru
"Lucian Blaga" University of Sibiu
Computer Science
Dr. I. Rațiu Street No.7-9, Sibiu, 550012
ROMANIA
E-mail: pirvu.doru@ymail.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Multilevel image thresholding for image segmentation

Ana Tuba
Teacher Coordinator: Milan Tuba

Abstract

One of the often used image processing techniques is image thresholding. It can be used for printing or other applications where reduced number of intensity levels is required. Image segmentation as a step in image analysis can also successfully be facilitated by appropriate image thresholding. This paper examines different approaches to binary or multilevel image thresholding. Different criteria for selecting threshold points are compared including maximum entropy thresholding (MET) as defined by Kapur et al. Since computational complexity exponentially increases with the increase of number of threshold levels alternative methods to exhaustive search are proposed.

Keywords: Image processing, Image thresholding, Multilevel thresholding, Image segmentation, Maximum entropy thresholding

1 Introduction

Digital images are becoming more popular and there is a growing number of situations where they are used. It is often required that a digital image is processed in some way to become more usable. Although the field of digital image processing is built on a foundation of mathematical formulations, human intuition and analysis play a central role in the choice of one technique versus another and this choice is often made based on subjective, visual judgments.

Digital images are for the purpose of image processing represented as a matrix of pixels where the light intesity for each pixel (gray or some color model with usualy three components) is recorded as an integer is some interval, usually 0 to 255, minimum value representing black and maximum value representing white.

Image thresholding is a techique of (significantly) reducing the number of intesity levels. The original number of possible intensity levels, which is usually 256, is reduced to only two (binary thresholding) or some other small number (multilevel thresholding). Even for the small number of desired remaining levels like 5 or 6, the problem of finding optimal in some sense points for level separation by exhaustive search becomes computationaly very expensive.

This paper examines different ways of determining thresholding levels and their characteristics. The rest of the paper is organized as follows: in Section 2 basics of digital images are presented, Section 3 introduces image histograms and their characteristics, Section 4 deals with the main topic of this paper: image thresholding and its various forms: binary and multi-level thresholding, different ways of determining threshold points including maximum entropy thresholding, while

Section 5 presents some applications of the image thresholding like printing and image segmentation.

2 Digital image

Digital image is electronically recorded image. There are two basic types of digital images: vector and raster. Vector images are decoded using mathematical formulas. Raster images are composed of pixels. Pixels should be small enough so that human eye would not identify them as separate elements, but perceive as a continuous image. The number of pixels horizontally and vertically represents the resolution of the image. Raster image is archived by recording for each pixel its brightness and possibly color information [1].

Raster images can be stored in many file formats [2]. The BMP file format is very simple and under Windows widely used. It is capable of storing 2D digital images of arbitrary width, height (resolution), both monochrome and color, in various color depths, and optionally with some primitive data compression. BMP images are stored in a format known as device-independent bitmap (DIB). Only when these pictures are to be displayed at the output device, the device driver translates DIB color to the color the output device can display.

BMP image consists of 3 or 4 parts. The first tw parts are file header followed by an informational part. If the image uses color palette then follows (optional) palette-table and at the end are the pixel data. The position of pixel data from the beginning of the file is given in the file header. Information relating to the width and height of the picture, compression and color are in the information header. The file header is 14 bytes long and consists of the following fields:

```
typedef struct {
        unsigned short int type;                   /* magic identifier */
        unsigned int size;                         /* size of image in bytes */
        unsigned short int reserved1, reserved2;
        unsigned int offset;                       /* Offset to pixel. data byte */
} HEADER;
```

The field *type* (should be 'BM') can be used as a simple test that it is a legitimate BMP file and the field *offset* gives the number of bytes where real data (pixels) is from the beginning of the file.

Image data are in the information header which is 40 bytes long. The most interesting fields are the *width* and *height*, number of pixels per bit (can be 1, 4, 8 or 24), the number of color planes and type of compression (if used).

```
typedef struct {
        unsigned int size;               /* Header size in bytes */
        int width,height;                /* Width and height of image */
        unsigned short int planes;       /* Number of color planes */
        unsigned short int bits;          /* Bits per pixel */
        unsigned int compression;        /* Compression type */
        unsigned int imagesize;          /* Image size in bytes */
        int xresolution,yresolution;     /* Pixels per meter */
        unsigned int ncolors;            /* Number of colors */
        unsigned int importantcolors;     /* Important colors */
} INFOHEADER;
```

The simplest format of BMP files for image processing experiments is 24-bit (true color) format without compression or color palette. Pixel data consists of 3 bytes per pixel in the sequence b, g, r and each row is padded to the nearest number divisible by 4.

3 Digital image histogram

Histograms are very important for the understanding of digital images. They are used to depict image statistics in an easily interpreted visual format [3], [4], [5]. With a histogram it is easy to determine certain types of problems in an image, for example, it is simple to conclude if an image is properly exposed by visual inspection of its histogram. In fact, histograms are so useful that modern digital cameras often provide a real-time histogram overlay on the viewfinder (Fig. 1) to help prevent taking poorly exposed pictures. It is important to catch errors like this at the image capture stage because poor exposure results in a permanent loss of information which it is not possible to recover later using image-processing techniques.

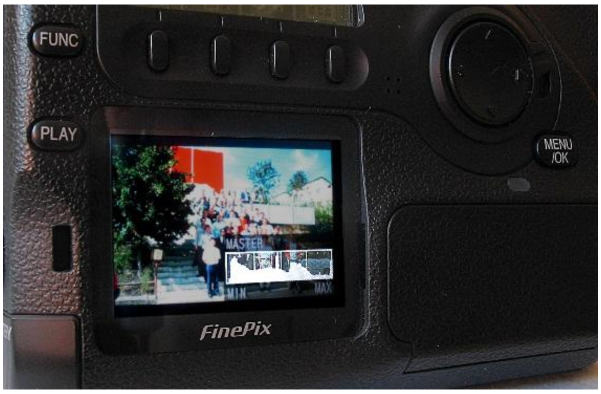


Fig. 1: Digital camera back display showing a histogram overlay

Histograms of images describe the frequency of the intensity values that occur in an image. This concept can be easily explained by considering a grayscale image like the one shown in Fig. 2. A histogram *h* for a grayscale image *I* with intensity values in the range

$$I(u, v) \in [0, K-1] \tag{1}$$

would contain exactly *K* entries, where for a typical 8 bit grayscale image $K = 2^8 = 256$. Each individual histogram entry is defined as *h(i)* = the number of pixels in image *I* with the intensity value *i*, for all $0 \leq i < K$. Therefore *h(0)* is the number of black pixels with the value 0, *h(1)* the number of pixels with the value 1, and so forth. Finally *h(255)* is the number of all white pixels with the maximum intensity value 255 = K−1. The result of the histogram computation is a one-dimensional vector *h* of length *K* (Fig.3).

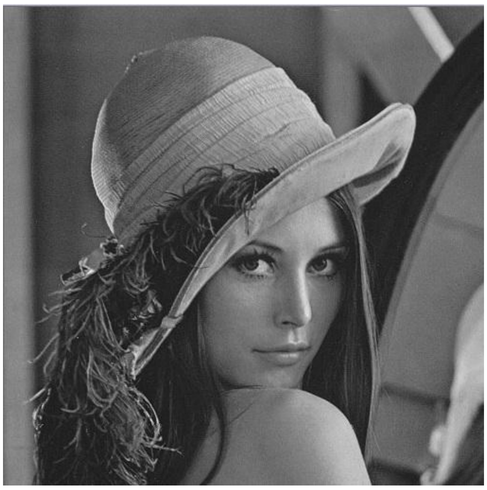


Fig. 2: Image Lena.bmp

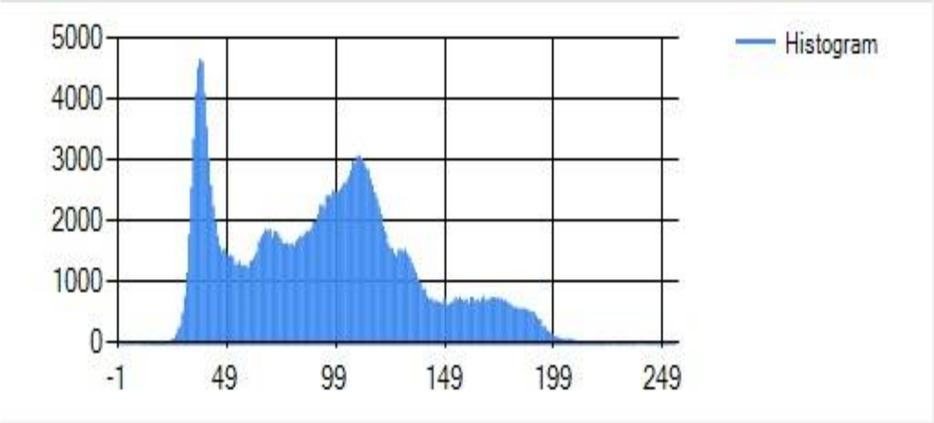


Fig. 3: Histogram of image Lena.bmp

Since a histogram encodes no information about where each of its individual entries originated in the image, histograms contain no information about the spatial arrangement of pixels in the image. This is intentional since the main function of a histogram is to provide statistical information (the distribution of intensity values) in a compact form. Given the loss of spatial information, in all but the most trivial cases, it is not possible to reconstruct an image using only its histogram. As an example, consider the wide variety of images that can be constructed using the same number of pixels of a specific value. These images would appear different but have exactly the same histogram (Fig. 4).

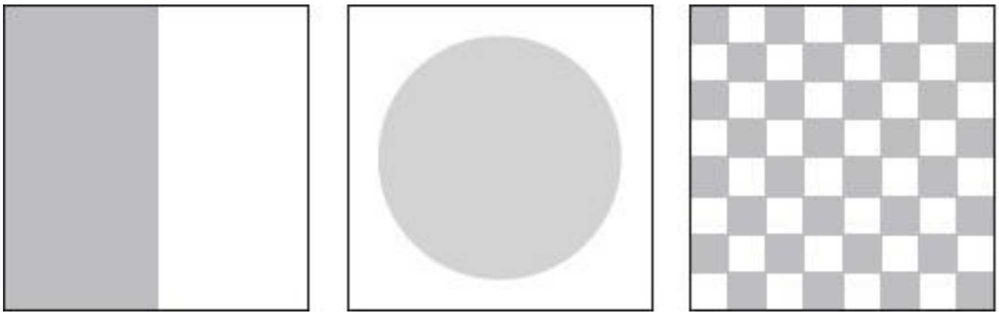


Fig. 4: Three very different images with identical histograms

4 Thresholding

Thresholding is one of the simplest techniques for performing image segmentation and it is very useful in separating objects from image background, or discriminating objects from other objects that have distinct gray-levels. Thresholding involves bi-level thresholding and multilevel thresholding.

Bi-level thresholding is used to create binary images [6]. For bi-level thresholding the main objective is to determine one threshold which separates pixels into two groups, one including those pixels with gray levels above that threshold, the other including the rest.

Thresholding is still used for the preparation of black and white pictures for printing. If the image intensity exceeds some threshold value the pixel is set to white, otherwise it is set to black. White corresponds to the maximum display intensity. The threshold is usually set at approximately half the maximum display intensity (128 out of 255). Figure 5 (a), (b) and (c) shows results for the image "Lena.bmp" with thresholds 125, 80 and 60. It illustrates how a simple thresholding technique results in the considerable loss of fine detail. The fine detail is lost because of the relatively large errors in displayed intensity for each pixel.

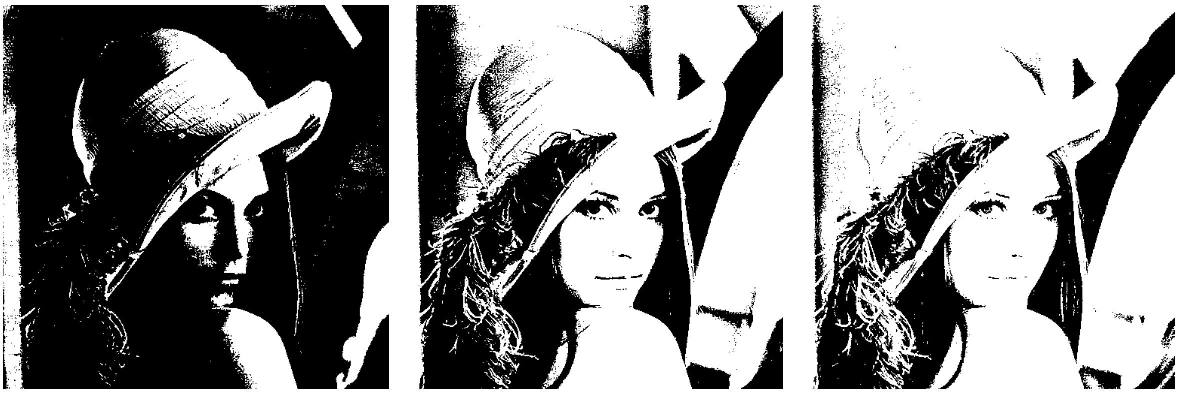


Fig. 5: (a) Lena threshold 125 (b) threshold 80 (c) threshold 60

The threshold can be selected in various ways. One option is to calculate the average value. Then the image "Lena" looks as shown in Figure 6 (a). Another possibility is to calculate the median as shown in Figure 6 (b). It may also be appropriate to use the maximum value (Figure 6 (c)).

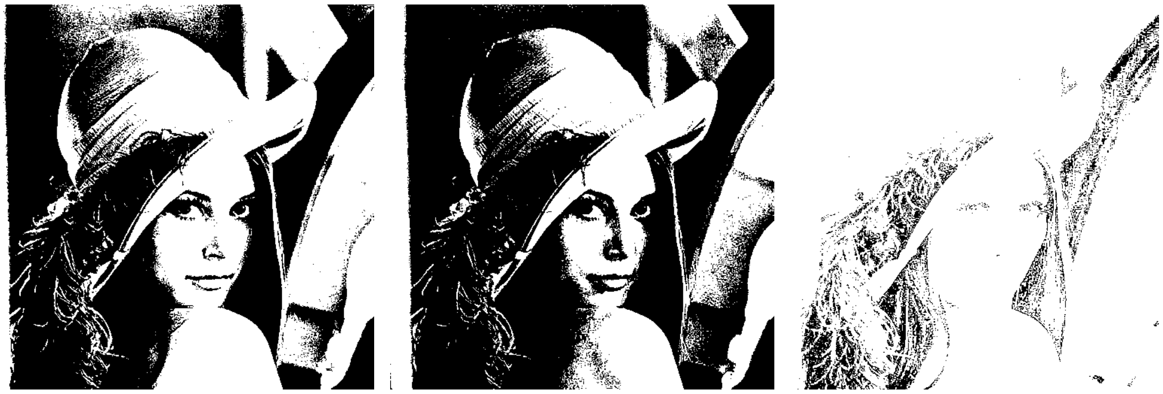


Fig. 6: (a) Average threshold 94 (b) Median threshold 102 (c) Mod threshold 37

4.2 Multi-level thresholding

Multilevel thresholding is one of the most popular image segmentation techniques [7], [8], [9]. For multilevel thresholding the main objective is to determine multiple thresholds which divide pixels into several groups. The pixels which belong to the same class have gray levels within a specific range defined by two neighbour thresholds.

Suppose that the intensity histogram in Figure 7 corresponds to an image, *I(x, y)*, composed of a light object on dark background, in such a way that object and background pixels have intensity values grouped into two dominant modes. One obvious way to extract the object from background is to select a threshold, *T*, that separates these modes. Then, any point *(x, y)* in the image at which *I(x, y)>T* is called an object point; otherwise, the point is called a background point. In other words, the segmented image *I(x, y)* is given by:

$$\mathrm{I(x,\ y)} = \begin{cases} 1, I(x,y) > T \\ 0, I(x,y) \le T \end{cases} \qquad (2)$$

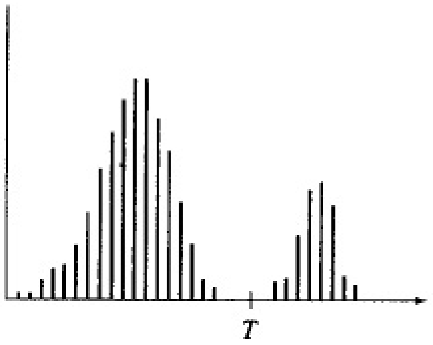


Fig. 7: Histogram which can be partitioned by a single threshold

When *T* is a constant applicable over an entire image, the process detrmined by Equation (2) is referred to as global thresholding. When the value of *T* changes over an image, the term variable thresholding is used. The term local or regional thresholding is used sometimes to denote variable thresholding in which the value of *T* at any point *(x, y)* in an image depends on properties of a neighborhood of *(x, y)*. Figure 8 shows a more difficult thresholding problem involving a histogram with three dominant modes corresponding, for example, to two types of lights objects on dark background. Here, multi-level thresholding classifies a point *(x, y)* as belonging to the background if $I(x, y) \le T_1$, to one object class if $T_1 < I(x, y) \le T_2$, and to the other object class if $I(x, y) > T_2$. That segmented imege is given by:

$$\mathrm{I(x,\ y)}=\begin{cases} a, I(x,y) > T_2 \\ b, T_1 < I(x,y) \le T_2 \\ c, I(x,y) < T_1 \end{cases} \tag{3}$$

where *a*, *b* and *c* are any three distinct intensity values.

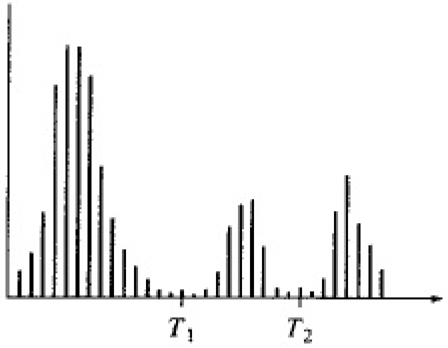


Fig. 8: Histogram which can be partitioned by a dual threshold

Segmentation problems requiring more then two thresholds are difficult (often impossible) to solve, and better results usually are obtained using other methods such as variable thresholding. We may infer intuitively that the success of intensity thresholding is directly related to the width and depth of the valley separating the histogram modes. In turn, the key factors affecting the properties of the valley are:

1. the separation between peaks (the further apart the peaks are, the better the chances of separating the modes)
2. the noise content in the image (modes broaden as noise increases)

3. the relative sizes of objects and background
4. the uniformity of the illumination source
5. the uniformity of the reflectance properties of the image.

In Figure 9 an image is segmentation with two threshold points is demonstrated. The threshold points are determined as deepest valleys between three acumulated levels.

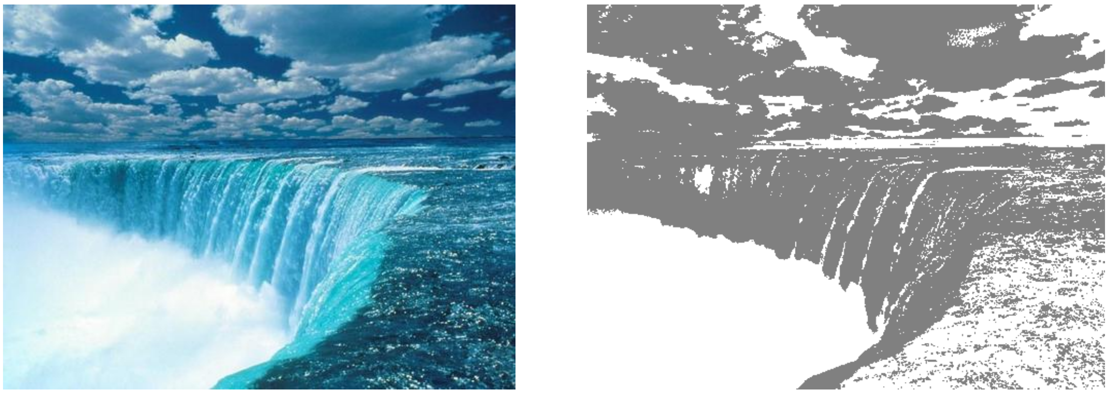


Fig. 9: Image segmentation with three level, with thresholding 85 and 208

4.3 Maximum entropy thresholding

The maximum entropy thresholding (MET) had been widely used for determining the optimal thresholding in image segmentation. The maximum entropy criterion for image thresholding was first proposed by Pun, and later it was corrected and improved by Kapur [10]. The maximum entropy method is based on the histogram of the image. It is a means of thresholding an image that selects an optimum threshold value by choosing the thresholding points from the image's histogram that exhibits the maximum entropy over the entire image. Let there be L gray levels in a given image I having M pixels and these gray levels are in the range $\{1, 2,..., L-1\}$. The multilevel thresholding problem can be configured as a k-dimensional optimization problem, for determination of k optimal thresholds $[t_1, t_2,..., t_k]$ which optimizes an objective function. Generally, these functions are determined from the histogram of the image, denoted by $h(i)$, $i=0, 1,..., L-1$, where $h(i)$ represents the number of pixels having the gray level i. The normalized probability at level i is defined by the ratio $P_i=h(i)/M$.. The goal is to maximize the objective function:

$$f([t_0, t_1,..., t_{k-1}])=H_0+H_1+...+H_k \tag{4}$$

where

$$H_0 = -\sum_{i=0}^{t_1-1} \frac{P_i}{w_0} \ln \frac{P_i}{w_0}, \; w_0 = \sum_{i=0}^{t_1-1} P_i \, ,$$

$$H_1 = -\sum_{i=t_1}^{t_2-1} \frac{P_i}{w_1} \ln \frac{P_i}{w_1}, \; w_1 = \sum_{i=t_1}^{t_2-1} P_i \, ,...$$

$$H_k = -\sum_{i=t_k}^{L-1} \frac{P_i}{w_k} \ln \frac{P_i}{w_k}, \; w_K = \sum_{i=t_k}^{L-1} P_i \tag{5}$$

If the histogram exibits clearly distinct valleys by observing the histogram the best thresholding points can be determined. Otherwise, the easiest way to determine the best thresholding is to apply the maximum entropy thresholding. At the histogram of the image "Lena" three valleys can be observed. Thresholding at these three valleys are shown in Figures 5a, 5b and 5c. But if the maximum entropy thresholding is applied for one level, the thresholding point 134 is obtained. This picture is shown in Figure 10 (a). In Figure 10 (b) is shown picture with thresholding points 86 and 141, because that thresholding points is obtained when the maximum entropy thresholding is applied for two level.

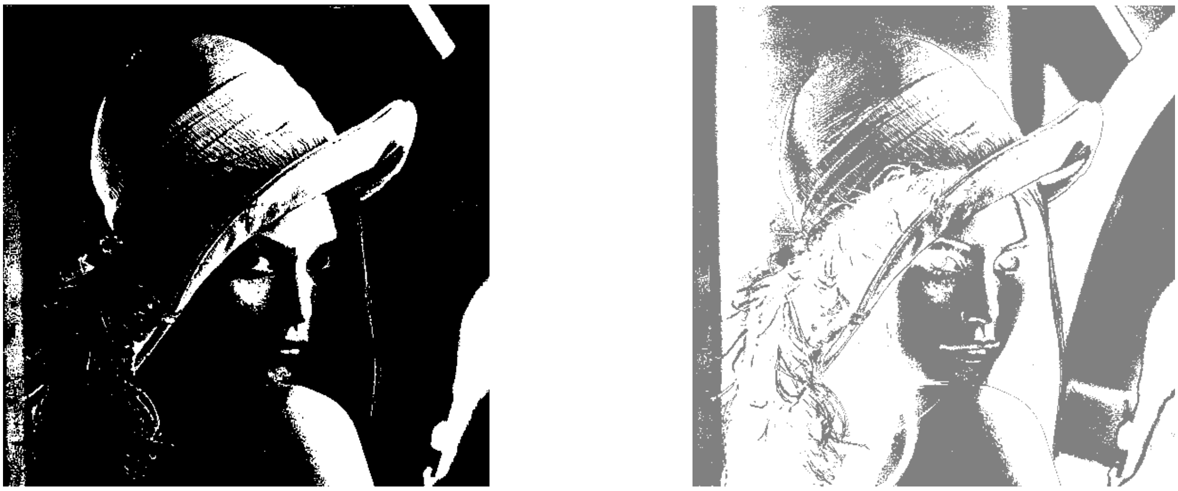


Fig. 10: (a) „Lena” with one level thresholding, 134 (b) with two level thresholding, 86 and 141

5 Applications of thresholding

4.1 Printing

Binary thresholding is the simplest technique for printing gray images on black and white devices. It is, however, rather crude method and more advanced techniques were developed. One of the most popular ones is a technique developed by Floyd and Steinberg that distributes error made when a gray pixel is substituted by a white or black one to surrounding pixels. Further, the algorithm is cleverly constructed so that the error is always distributed downward and to the right. If the image is computed in scan line order, no backtracking is necessary. In particular, the Floyd-Steinberg algorithm distributes the error 3/8 to the right, 3/8 downward, 1/4 down-right diagonally. With the thresholding midway between the maximum and minimum display intensities, *T=(black+white)/2,* the algorithm is:

```
Image(x, y) - intensity of the image at pixel (x, y)
Xmin, Xmax, Ymin, Ymax are the raster limits for each scan line-top to bottom
{
    T = (black + white) / 2;
    for (i = Ymin; i <= Ymax; i++)
      for (j = Xmin; j <= Xmax; j++)
        if (Image(x, y) < T)
          {
            Pixel(x, y) = black;
            Error = Image(x, y) - black;
          }
        else {
                Pixel(x, y) = white;
                Error = Image(x, y) - white;
              }
    Image(x+1, y) = Image(x+1, y) + 3/8*Error;
    Image(x, y+1) = Image(x, y+1) +3/8*Error;
    Image(x+1, y+1) = Image(x+1, y+1) + Error/4;
}
```

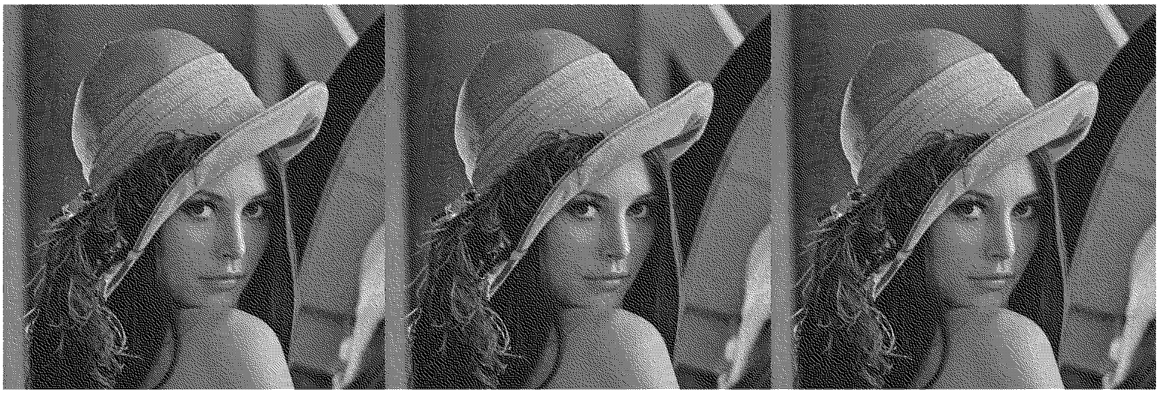


Fig. 11: Floyd-Steinberg algorithm applied on image "Lena.bmp"
(a) threshold 125 (b) threshold 80 (c) threshold 60

Distributing the error to neighboring pixels improves the detail in the image because it preserves the information inherent in the image. As we can see on Fig. 11 (a), 11 (b) and 11 (c), image is not sensitive to selecting threshold point.

4.1 Image segmentation

Image segmentation is one of the most important operations in image analysis and computer vision [11], [12]. Segmentation subdivides an image into its constituent regions or objects. The level of detail to which the subdivision is carried depends on the problem being solved. That is, segmentation should stop when the object or regions of interest in an application have been detected. Thresholding, region growing, and region splitting and merging are examples of methods in this category. For most of these approaches improvements in segmentation performance can be achieved by combining methods from distinct categories, such as techniques in which edge detection is combined with thresholding. Also image segmentation based on morphology is particularly attractive because it combines several of the positive attributes of segmentation.

Let R represent the entire spatial region occupied by an image. We may view image segmentation as a process that partitions R into n subregions, $R_1, R_2,..., R_n$, such that:

1. $\bigcup_{i=1}^{n} R_i = R$
2. R_i is a connected set, $i=1, 2,..., n$
3. $R_i \cap R_j=0$
4. $Q(R_i)=TRUE$ for $i=1, 2,..., n$
5. $Q(R_i \cup R_j)=FALSE$ for any adjacent regions R_i and R_j.

Here, $Q(R_k)$ is logical predicate defined over the points in set R_k, and 0 is the null set. The first condition indicates that the segmentation must be complete. That is every pixel must be in a region. The second condition requires that points in a region be connected in some predefined sense. The third condition indicates that the regions must be disjoint. The fourth condition deals with the properties that must be satisfied by the pixels in a segmented region, for example, $Q(R_i)=TRUE$ if all pixels in R_i have the same intensity level. The fifth condition indicates that two adjacent regions R_i and R_j must be different in the sense of predicate Q.

Thus, we see that the fundamental problem in segmentation is to partition an image into regions that satisfy the preceding conditions. Segmentation algorithms for gray images generally are based on one or two basic categories dealing with properties of intensity values: discontinuity and similarity. In the first category, the assumption is that boundaries of regions are sufficiently

different from each other and from the background to allow boundary detection based on local discontinuities in intensity. Edge-based segmentation is the principal approach used in this category. Region-based segmentation approaches in the second category are based on partitioning an image into regions that are similar according to a set of predefined criteria.
Figure 12 shows image segmentation using multilevel thresholding. Threshold points are determined by observing histogram (b) and maximum entropy threshold (c).

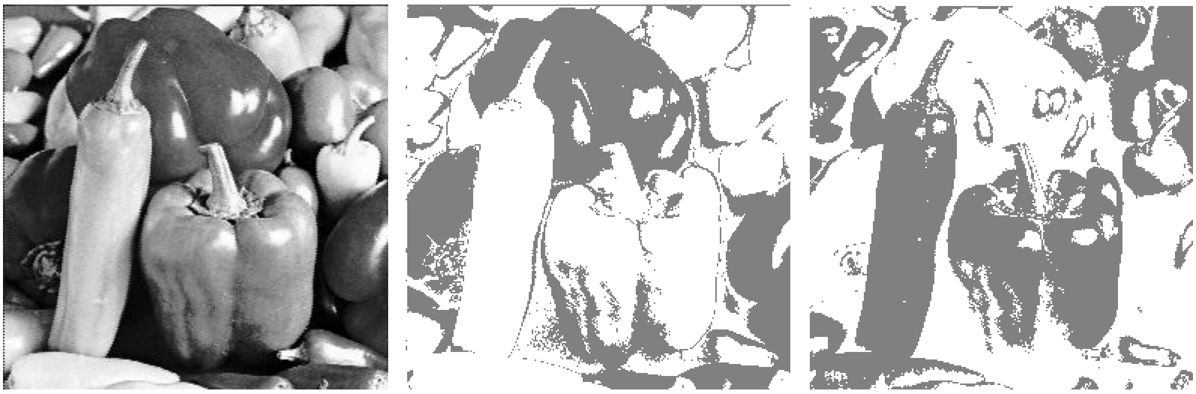


Fig. 12: Image segmentation
(a): original image (b): with thresholding 16 and 95 (c): with thresholding 93 and 178

On Figure 12 is shown original image (a), then image with thresholding which we see by observing the histogram (b), and image with thresholding which we calculate by maximum entropy thresholding algorithm.

6 Conclusion

In this paper, we examined tresholding as a very useful technique for image processing. Several ways to determine where the threshold points would be are demonstrated. One is the minimum valley of the histogram, the second is the median of the histogram. If the histogram has more valleys, it is best to have as many thresholding points as there are histogram valleys. With a number of thresholding points exponentially increases computational time required to determine that points if exhaustive search is used.

References

[1] Rafael C. Gonzales, Richard E. Woods. *Digital Image Processing*, 3rd Edition, Pearson Prentice Hall, p. 954 2008.

[2] W. Burger and M. Burge. *Principles of Digital Image Processing*. Springer-Verlag London, p. 261, 2009.

[3] E. Jeffrey and P. Van Dyke. *Histogram generation apparatus and method for operating the same*. IEEE Transactions on Consumer Electronies, Vol. 44, No. 4, pp. 1389-1395, 1998.

[4] H. Yang. *Design of Digital Image Histogram Equalization Based on CCS*. Journals of Modern Electronics Technique, 2010.

[5] P. L. Rosin. *Evaluation of global image thresholding for change detection*. Elsevier Pattern Recognition Letters, Vol. 24, Issue 14, pp. 2345-2356, 2003.

[6] P. Shanmugavadivu and K. Balasubramanian. *Image Inversion and Bi-Level Histogram Equalization for Contrast Enhancement*. International Journal of Computer Applications, Vol. 1, No. 15, pp. 61-65, 2010.

[7] P. S. Liao, T. S. Chen and P. C. Chung. *A fast Algorithm for Multilevel Thresholding*. Journal of Information Science and Engineering, Vol. 17, pp. 713-727, 2001.

[8] K. Hammouche, M. Diaf and P. Siarry. *A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation*. Elsevier Computer Vision and Image Understanding, Vol. 109, Issue 2, pp. 163-175, 2008.

[9] P. Gupta, V. Malik and M. Gandhi. *Implementation of Multilevel Threshold Method for Digital Images Used In Medical Image Processing*. International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, Issue 2, 2012.

[10] J. N. Kapur, P. K. Sahoo and A. K. C. Wong. *A new method for grey-level picture thresholding using the entropy of the histogram*. Computer Vision Graphics Image Processing, pp. 271-285, 1985.

[11] S. Sarkar, G. R. Patra and S. Das. *A Differential Evolution Based Approach for Multilevel Image Segmentation Using Minimum Cross Entropy Thresholding*. Lecture Notes in Computer Science, Swarm, Evolutionary, and Memetic Computing, Vol. 7076, pp. 51-58, 2011

[12] S. Arora, J. Acharya, A. Verma and P. K. Panigrahic. *Multilevel thresholding for image segmentation through a fast statistical recursive algorithm*. Elsevier Pattern Recognition Letters, Vol. 29, Issue 2, pp. 119-125, 2008.

Ana Tuba
University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: anatuba89@gmail.com

Prof. Milan Tuba
University Megatrend Belgrade
Faculty of Computer Science
Bulevar umetnosti 29, N. Belgrade
SERBIA
E-mail: tuba@ieee.org

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Ant Colony Optimization in a parallel framework

George Viorel Vişniuc, Mihai Manea, Radu Rotari, Alexandru Ioan Florea

Teacher Coordinator: Lecturer PhD. Cerasela Crişan

Abstract

This paper describes the process of redesigning a software application for parallel processing in order to provide better and faster solutions for graph related problems. The work consists in designing and implementing an application that uses an Ant Algorithm in a parallel programming environment configured to solve several instances of the Traveling Salesman Problem. The results show promising behavior, enabling further developments.

1. Introduction

The purpose of this project is to use parallel computing and inter-process communication in order to generate faster and better results for TCP instances, using Ant algorithms.

Ant algorithms are based on the observation that ant colonies use pheromone trails to indicate efficient already-found paths. These problems are addressed by the Ant Colony Optimization metaheuristic template. Real-life graph-represented problems similarly present themselves, as the seeking of a Hamiltonian path in a weighted graph. We focus here on the Traveling Salesman Problem (TSP).

The software application that was modified in this project was created by Thomas Stützle, professor in the Computer Science Department at Darmstadt University of Technology. Our project consists of several Ant algorithm-based sequential modules that solve TSP instances that have the TSPLIB format. The algorithms we used are the *MAX-MIN Ant System* and the *Elitist Ant System*.

The performance was enhanced by developing and using a tailor-made communication design intended to optimize information transfer between processes under a framework developed with the help of MPI libraries. Multi-processor systems can increase the overall performance of this application if it is executed on more processes.

2. Theoretical aspects

This section provides definitions and explanations of terminology throughout the document.

2.1. Parallel Computing

Parallel computing is the Computer Science discipline that deals with the system architecture and software issues related to the concurrent execution of applications. It has been an area of active research interest and application for decades, mainly focusing on high performance computing, but is now emerging as the prevalent computing paradigm due to the semiconductor industry's shift to multi-core processors [1].

Parallel computing is an evolution of serial computing that attempts to emulate what has always been the state of affairs in the natural world: many complex, interrelated events happens at the same time, yet within a sequence. We have chosen to design and implement a parallel application in order to be more efficient and to explore some new hardware features [2].

2.2. Message Passing Interface

MPI is a library, not a language. It specifies the names, calling sequences, and results of subroutines to be called from FORTRAN programs, the functions to be called from C programs, and the classes and methods that make up the MPI C++ library. The programs that users write in FORTRAN, C, or C++ are compiled with ordinary compilers and linked with the MPI library.

MPI is not a revolutionary new way of programming parallel computers. Rather, it is an attempt to collect the best features of many message-passing systems that have been developed over the years, improve them where appropriate, and standardize them. The structure of MPI makes it straightforward to port existing codes and to write new ones without learning a new set of fundamental concepts. Nevertheless, the attempts to remove the shortcomings of existing systems have made even the basic operations a little different [3].

2.3. The Traveling Salesman Problem

The goal of the Traveling Salesman Problem (TSP) is to find a minimum-length tour of a given number of cities, visiting each city exactly once and returning to the starting city.

Many routing problems generalize the TSP. The problem of efficient computer wiring can also be modeled as a TSP: it considers several modules, each with a number of pins. The efficient scheduling of jobs on a single machine given the time needed by each job and the time needed to prepare the machine for each job is also TSP.

Ants from the artificial colony are able to successively generate shorter feasible tours by using information accumulated in the form of a pheromone trail deposited on the edges of the TSP graph. Computer simulations demonstrate that the artificial ant colony is able to generate good solutions to both symmetric and asymmetric instances of the TSP. The Ant-based method is an example, like simulated annealing, neural networks, and evolutionary computation, of the successful use of a natural metaphor to design an optimization algorithm [4]·

TSPLIB is a library of sample instances for the TSP [5] (and related problems) from various sources and of various types. There are several types of instances of TSP:

– Symmetric traveling salesman problem (TSP): Given a set of n nodes and distances for each pair of nodes, find a roundtrip of minimal total length visiting each node exactly once. The distance from node *i* to node *j* is the same as that from node *j* to node *i*.

– Hamiltonian cycle problem (HCP): Given a graph, test if the graph contains a Hamiltonian cycle or not.

– Asymmetric traveling salesman problem (ATSP): Given a set of n nodes and distances for each pair of nodes, find a roundtrip of minimal total length visiting each node exactly once. In this case, the distance from node *i* to node *j* and the distance from node *j* to node *i* may be different.

2.4. Heuristic solving method for a problem

A heuristic solving method balances the quality of the result with the resources used in finding it. When one applies a heuristic algorithm, he expects to quickly obtain a “good-enough” result.

Ants switch information by depositing pheromone tracks. A moving ant leaves, in varying quantities, some pheromone on the ground to mark its way. While an isolated ant moves essentially at random, an ant encountering a previously laid trail is able to detect it and decide with high probability to follow it, thus reinforcing the track with its own pheromone. This elementary ant's behavior inspired the development of Ant Colony Optimization [7] (ACO) by Marco Dorigo in 1992, a meta-heuristic stochastic combinatorial methodology belonging to a family of related meta-heuristic methods such as simulated annealing, Tabu search and genetic algorithms.

2.5. Ant Algorithms

Ant algorithms (AAs) form a new nature-inspired optimization technique used in solving Combinatorial Optimization Problems (COPs). They describe an iterative process in which a population of simple agents (artificial ants) repeatedly creates candidate solutions of the given problem.

There are two data sets that probabilistically guide the solution creation process of AA. These are: the heuristic information on the given problem and the memory containing experience gathered by ants in the previous iterations (the pheromone trails). The communication between ants is mediated by the selective deposit of pheromone on the elements of good solutions. The quantity of pheromone deposited on each element reflects the quality of the solution it belongs. Thus, the elements with a higher quantity of pheromone become more attractive to the other ants. At the end, the best solution is chosen [6].

3. The application

3.1 The communication design

The communication between processes has tremendous impact on the quality of a parallel application. When carefully designed, it enables the application to perform better, resource- and time-wise, hence justifying the project scope. The communication design has thus to address the exchanged information (the content), the events that trigger the exchange (the timing), the pairing of exchanging processes (the connectivity), the characteristics of communications (the mode), the purpose of the received information (the exploitation), and the further processing of the information (the scope).

3.2 Overall description

The communication design is based on a "master-slave" model where information is centralized and distributed by the master. The master process defined as process 0 is responsible for the following procedures:

- Detect incoming data transmissions, which are founded and delivered as solutions by the slave processes
- Perform integrity checks in order to confirm the validity of the solution
- Store a data structure containing the pool of solutions
- Perform a sorting procedure on the data structure to recover the best current solution
- Communicate the best stored solution back to the slave processes.

No communication is done between slave processes. The transferred data consist of a Hamiltonian path and its length.

The slave processes which individually work in order to return a solution have the following tasks:

- Perform a given number of iterations of the algorithm; once done, communicate the best solution to the master
- Perform the re-entry procedure that consists of updating the pheromone trails and restarting with the parameters specified by the solution just received.

The algorithm was applied to Traveling Salesman Problem (TSP) instances from TSPLIB [5].

4. Results, Conclusion & Further Development

A series of testing scenarios were designed to compare results given by the original sequential application with those generated by the parallel version.

The original application successively starts a number of tries. The modified application starts them in parallel. In any test, the number of processes used must be equal to the number of tries. If the number of processes is less than or equal to the number of processors, the performance enhancement is proportional.

The tests were run using:

- (#1) the original software;
- (#2) the modified software, with the ant algorithms not using the other tries' solutions;
- (#3) the modified software, with the ant algorithms using the other tries' solutions.

The focus of the project is twofold:

- to increase the performance of the application on multi-core machines;
- to faster generate better solutions through inner-process communication of the best solutions found.

We measured the performance increase by comparing, in each test, the number of iterations that were run in the span of 300 seconds. The results obtained are presented below. We used MAX-MIN Ant System and Elitist Ant System on the TSP instances pr2392 and att532. Each test lasted for 300 seconds, used 50 ants per iteration and the parallel version had 2 slaves. The tests were run on a quad core 3300 GHz Intel i5 2500k processor.

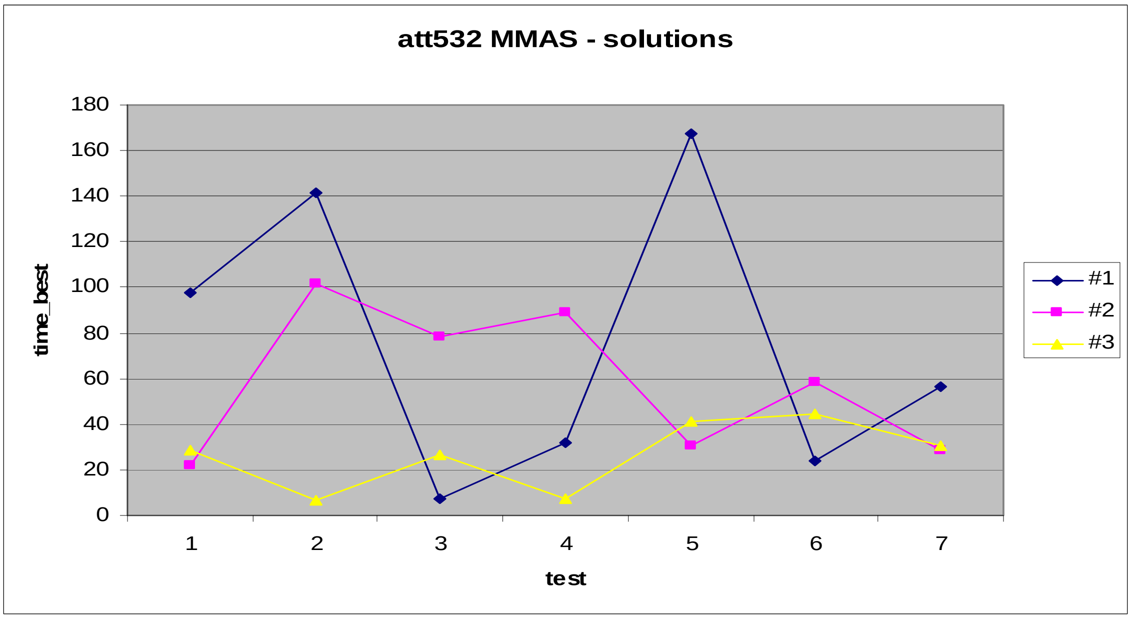
This is an example of a report generated by the modified application:

```
=== TOP 10 ===
lung_top10[0] = 27686  -  in 28.595 seconds
lung_top10[1] = 27693  -  in 27.238 seconds
lung_top10[2] = 27705  -  in 7.051 seconds
lung_top10[3] = 27708  -  in 8.44 seconds
lung_top10[4] = 27709  -  in 7.051 seconds
lung_top10[5] = 27712  -  in 8.44 seconds
lung_top10[6] = 27722  -  in 5.46 seconds
lung_top10[7] = 27723  -  in 5.46 seconds
lung_top10[8] = 27742  -  in 7.051 seconds
lung_top10[9] = 27780  -  in 3.713 seconds
#0 - time taken = 300.004000
#0 - iterations used = 9705
```

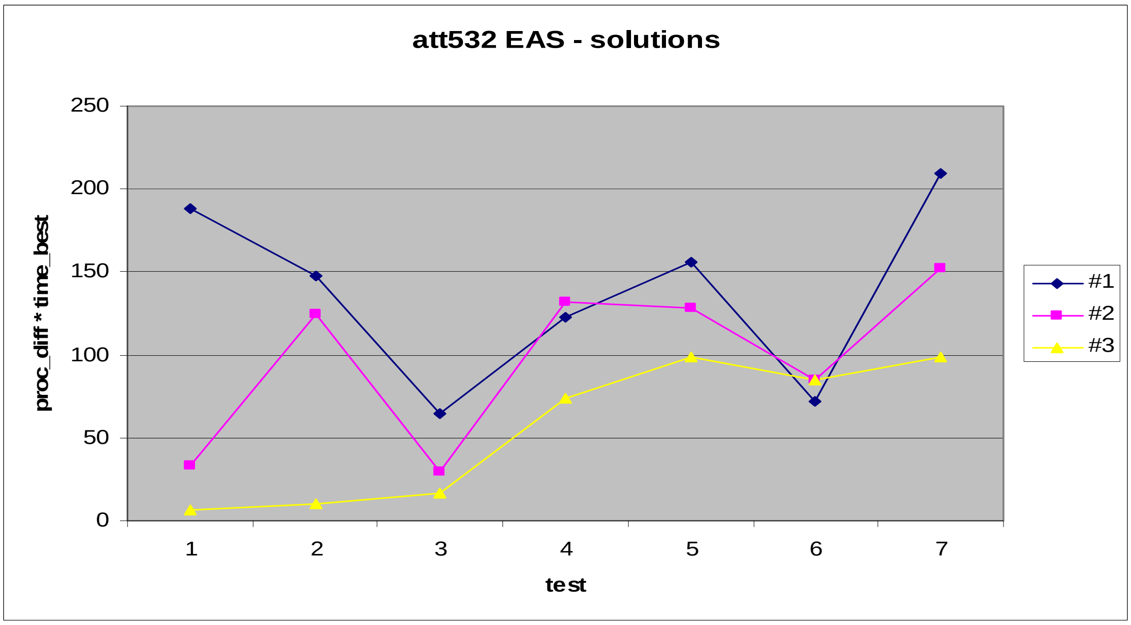
The time in which the best solution was found was taken into account in the following way:

- in series of tests where the application did not always generate the optimum result in 300 seconds, the time it took to find the best solution was multiplied by the percentage difference between the result and the optimum;
- in series of tests where the application always generated the optimum result in 300 seconds, only the best solution's time was plotted.

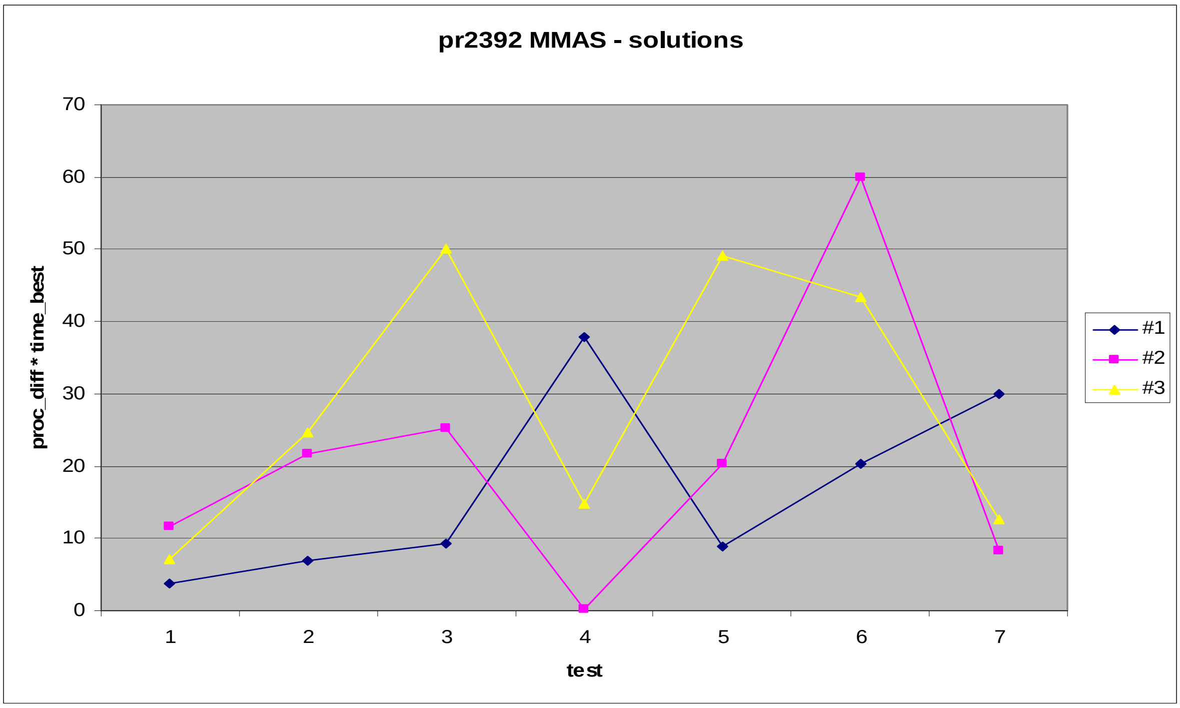
The att532 problem was optimally solved in all tests using the Max-Min Ant System. The graph below shows that when using solutions from other processes, the optimum solution tends to be found faster.

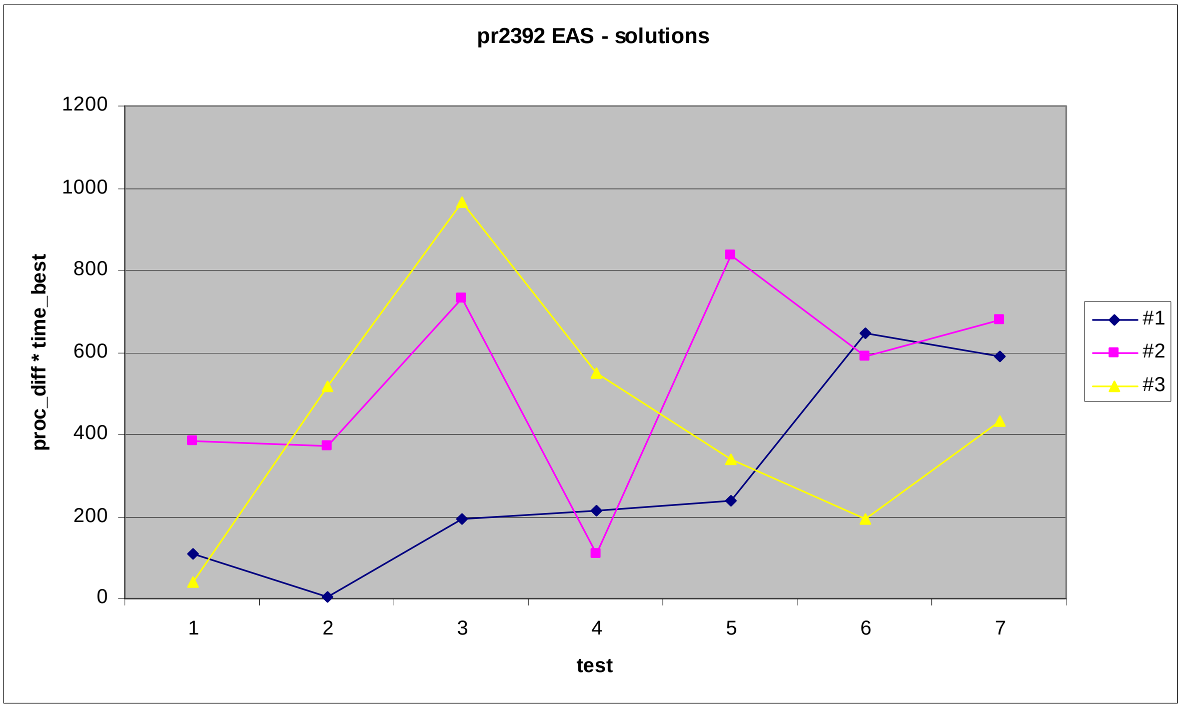


The rest of the tests did not find the optimal solution in the 300 seconds time limit. The results are quantified in the graphs below by multiplying the best tour length found with the time it took to find it. Again, when solving att532 with the Elitist Ant System, there is a clear improvement when the processes share and use each other's solutions.

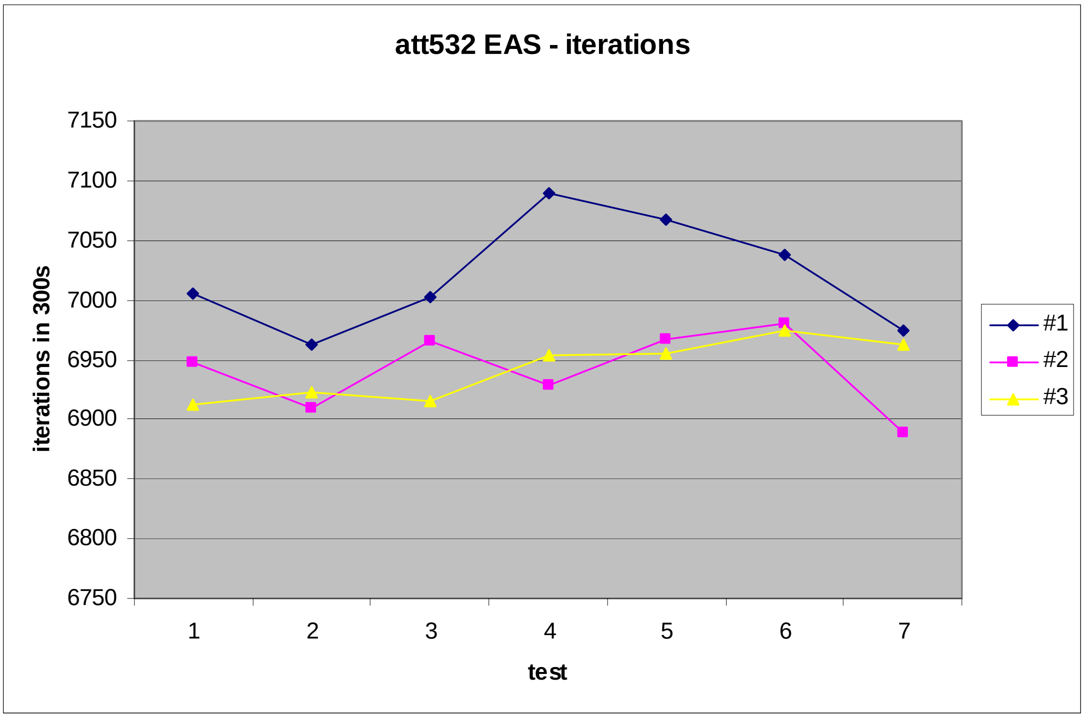


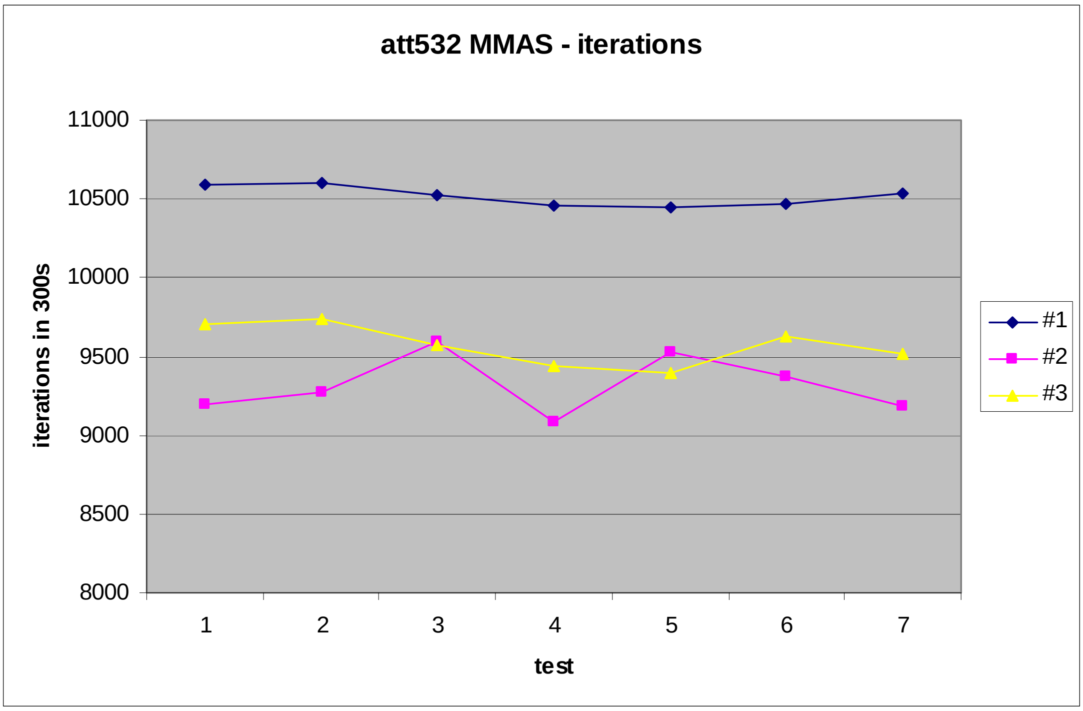
The results were ambiguous in our other two tests, attempting to solve pr2392:

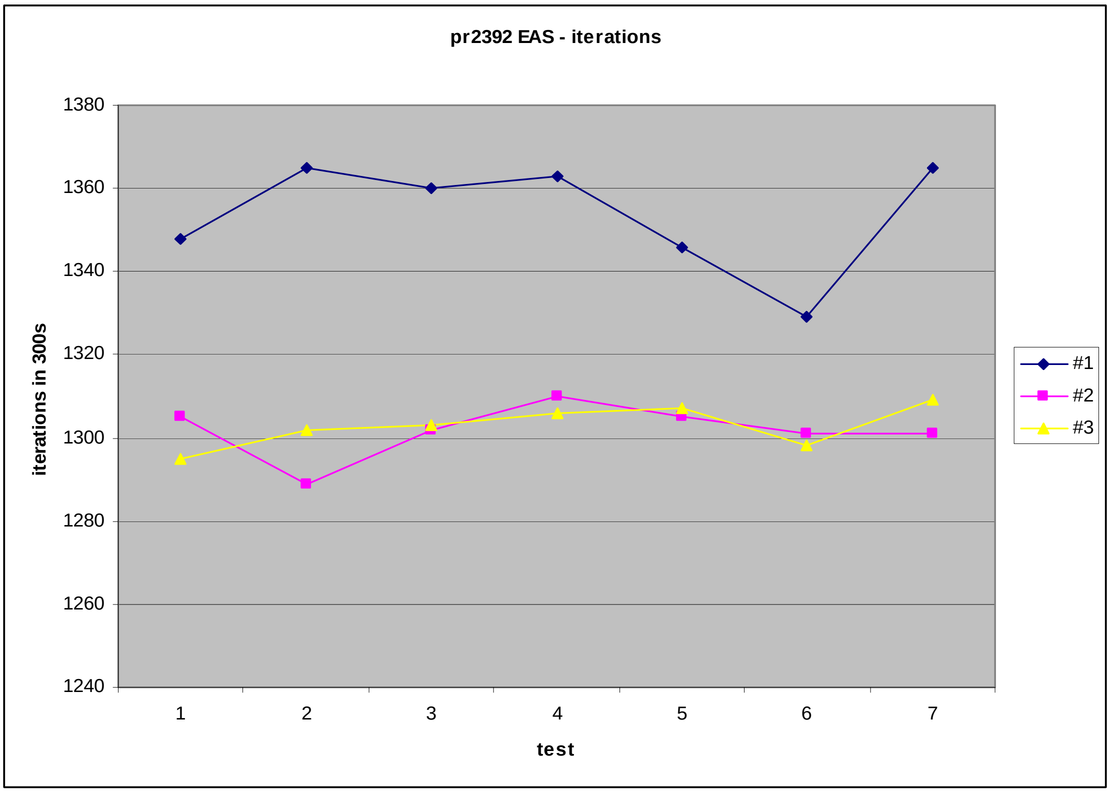


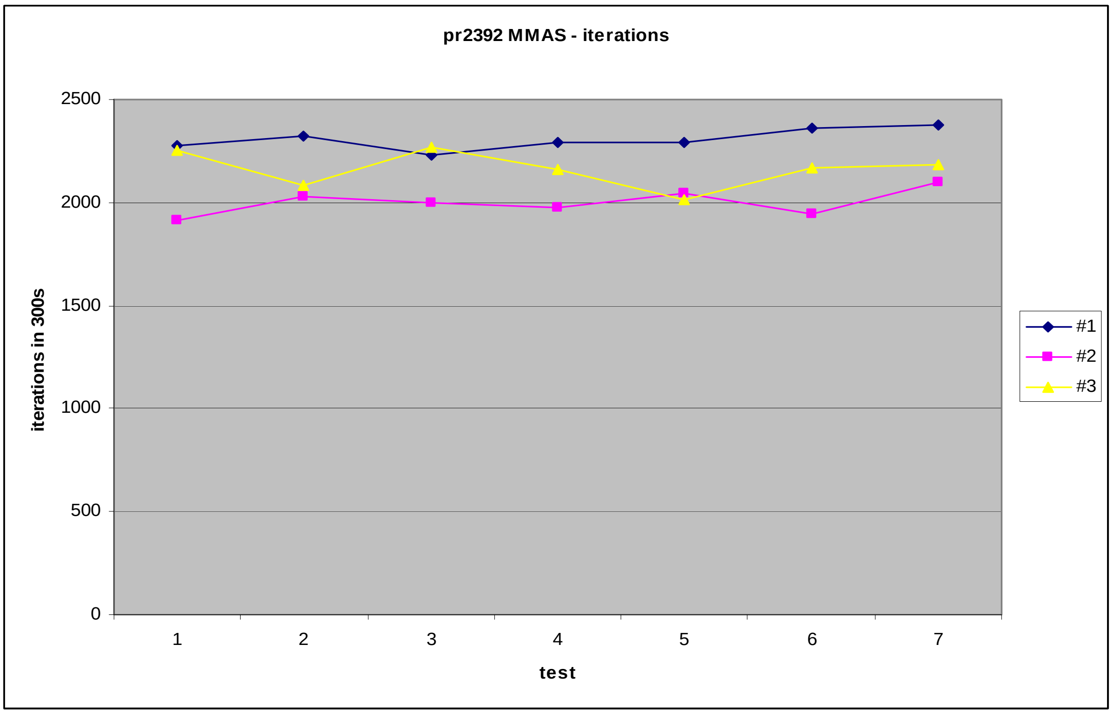


The following graphs show the decrease in performance per process as compared to the original application. The overall performance increase can be measured by multiplying by 3 the number of iterations in cases #2 and #3.






pr2392 EAS - iterations
iterations in 300s
1380
1360
1340
1320
1300
1280
1260
1240
1
2
3
4
5
6
7
test
#1
#2
#3


pr2392 MMAS - iterations
iterations in 300s
2500
2000
1500
1000
500
0
1
2
3
4
5
6
7
test
#1
#2
#3

We intend to further develop our application by:

- optimizing the code by using non-blocking communication between processes;
- factoring the number of common edges into the top10 ranking process, thereby increasing the odds that the ants will avoid local optimums by taking more diverse paths and increasing the chance that shorter paths will be generated faster.

References

[1] *THE MANYCORE SHIFT: Microsoft Parallel Computing Initiative Ushers Computing into the Next Era,* White paper, 2007

[2] Blaise Barney, *Introduction to Parallel Computing, Lawrence Livermore National Laboratory.*

[3] Gropp William, Lusk Ewing, Skjellum Anthony*, Using MPI : Portable Parallel Programming With the Message-passing Interface Scientific and Engineering Computation,* MIT Press, 1999.

[4] Kylie Bryant, *Genetic Algorithms and the Traveling Salesman Problem,* Department of Mathematics Harvey Mudd College, 2000.

[5] Gerhard Reinelt, *TSPLIB95,* University of Heidelberg.

[6] Huseyin Atakan Varol, Zafer Bingul *A New PID Tuning Technique Using Ant Algorithm,* American Control Conference Boston, Massachusetts, 2004

[7] M. Dorigo & T. Stützle, *Ant Colony Optimization,* MIT Press, 2004

George Viorel Visniuc
"Vasile Alecsandri"
University of Bacau
Department of Informatics
Calea Marasesti, 157, 600115,
ROMANIA
visniuc_george@yahoo.com

Mihai Manea
"Vasile Alecsandri"
University of Bacau
Department of Informatics
Calea Marasesti, 157, 600115,
ROMANIA
manea_a_mihai@yahoo.com

Radu Rotari
"Vasile Alecsandri"
University of Bacau
Department of Informatics
Calea Marasesti, 157, 600115,
ROMANIA
rotari.radu@gmail.com

Alexandru Ioan Florea
"Vasile Alecsandri"
University of Bacau
Department of Informatics
Calea Marasesti, 157, 600115,
ROMANIA
alecz_fia@yahoo.com

Second International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2012, April 26-28
Sibiu, Romania

Paradigms of Artificial Intelligence Programming: Ruby Case Studies

Volosincu Bogdan, Rusu Andrei
Teacher Coordinator: Ralf Fabian

Abstract

The present paper presents some fundamental styles of computer programming in the big field of artificial intelligence. The intended meaning of the representations and demonstrations presented here is to create new paths for innovative and creative ideas that will be implemented and finally get materialized in a better-crafted software and also with steadily added value. The case studies of the present paper are from the point of view of Ruby Programming Language and Object-oriented programming (OOP).

1 Introduction

Artificial Intelligence (AI) is a cross-disciplinary field focusing on creating and research machines and systems that can engage on behaviors and operate in ways that humans consider intelligent. Given the complexity and goals of the artificial intelligence solutions the field extends from computer science to domains as neuroscience, biomedical, nature-inspired computation etc.

Peter Norvig [Director of Research, Google] provide a perspective that defines Artificial Intelligence in four categories:

1) systems that think like humans,
2) systems that act like humans,
3) systems that think rationally,
4) systems that act rationally [43] .

Artificial Intelligence is therefore concerned with investigating mechanisms that underlie intelligence and intelligence behavior. Although when we refer to artificial intelligence sounds very interesting there is still a lot of doubt and controversy about this filed, about the methods, solutions and in specially the meaning of the word intelligence.

The progress in this field is incredible and in continuous growth but it can’t compare with the most complex and attractive object in the universe witch is the brain witch at present, we poorly comprehend. Everybody knows that brain is responsible for control of our learning, memory, motivations, consciousness and that the artificial intelligence is trying to recreate all this by implementing different methods to achieve these goals. For example the fuzzy logic, neural networks, evolutionary computation, swarm intelligence.

2 AI Programming Paradigms in Ruby

2.1 Fuzzy Logic

Was defined almost half century ago and is the mathematical basis for ”computing of perceptions”[1]. The concept of Fuzzy Logic was conceived by Lotfi Zadeh at the University of California at Berkley and is a way of processing data by allowing partial set membership rather than crisp set membership or non-membership witch has proved to be incredible effective in processing of information with high level of uncertainty.

Fuzzy Logic is extremely developed and used in AI evolutionary computing including DNA computing, chaos theory and parts of learning theory and has become more like a separated science so that is not surprising that in the world are several organizations which are specialized on research of the fuzzy logic:

1) International Fuzzy Systems Association (IFSA)
2) Berkeley Initiative in Soft Computing (BISC)
3) North American Fuzzy Information Processing Society (NAFIPS)
4) The European Society for Fuzzy Logic and Technology (EUSFLAT)

2.2 Neural Networks

Another type of artificial intelligence programming paradigm is the neural network which is a form of multiprocessor computer system that tries to model the neurons networks of the brain and are very different - they are composed of many rather feeble processing units which are connected into a network. Their computational power depends on working together on any task - this is sometimes termed *parallel processing*.

2.1.1 Self-organizing map

Is a type of neural network that is trained using unsupervised learning to produce a low-dimensional, discredited representation of the input space of the training samples, called a map.

2.3 Evolutionary Computation

A paradigm that is concerned with the investigation of systems inspired by the neo-Darwinian theory of evolution by means of natural selection (natural selection theory and an understanding of genetics). Popular evolutionary algorithms include the Genetic Algorithm, Evolution Strategy, Genetic and Evolutionary Programming, and Differential Evolution [4, 5].

2.4 Genetic Algorithms

Is a programming technique that mimics biological evolution as a problem-solving strategy. The Genetic Algorithm module implements the Genetic Search and Chromosome classes. The Genetic Search is a generic class, and can be used to solved any kind of problems that performs a stochastic search of the solution of a given problem.

The Chromosome is "problem specific". Ai4r built-in Chromosome class was designed to model the Travelling salesman problem. If you want to solve other type of problem, you will have to modify the Chromosome class, by overwriting its fitness, reproduce, and mutate functions, to model your specific problem.

2.5 ID3 Decision Trees

The ID3 algorithm is used to build a decision tree, given a set of non-categorical attributes C1, C2, .., Cn, the categorical attribute C, and a training set T of records.

```
function ID3 (R: a set of non-categorical attributes,
                    C: the categorical attribute,
                    S: a training set) returns a decision tree;
begin
```

```
          If S is empty, return a single node with value Failure;
          If S consists of records all with the same value for
            the categorical attribute,
            return a single node with that value;
          If R is empty, then return a single node with as value
            the most frequent of the values of the categorical attribute
            that are found in records of S; [note that then there
            will be errors, that is, records that will be improperly
            classified];
          Let D be the attribute with largest Gain(D,S)
            among attributes in R;
          Let {dj| j=1,2, .., m} be the values of attribute D;
          Let {Sj| j=1,2, .., m} be the subsets of S consisting
            respectively of records with value dj for attribute D;
          Return a tree with root labeled D and arcs labeled
            d1, d2, .., dm going respectively to the trees

             ID3(R-{D}, C, S1), ID3(R-{D}, C, S2), .., ID3(R-{D}, C,Sm);
  end ID3;
```

Ruby source code:

Marketing target strategy example using ID3 Decision Trees in Ruby

Let's suppose that you are writting an application that must identify people as relevant marketing targets or not. The only information that you have is a collection of examples, provided by a marketing survey:

```
  DATA_LABELS = [ 'city', 'age_range', 'gender', 'marketing_target'  ]

  DATA_SET = [
              ['New York',  '<30',      'M',  'Y'],
        ['Chicago',   '<30',      'M',  'Y'],
        ['Chicago',   '<30',      'F',  'Y'],
        ['New York',  '<30',      'M',  'Y'],
        ['New York',  '<30',      'M',  'Y'],
        ['Chicago',   '[30-50)',  'M',  'Y'],
        ['New York',  '[30-50)',  'F',  'N'],
        ['Chicago',   '[30-50)',  'F',  'Y'],
        ['New York',  '[30-50)',  'F',  'N'],
        ['Chicago',   '[50-80]',  'M',  'N'],
        ['New York',  '[50-80]',  'F',  'N'],
        ['New York',  '[50-80]',  'M',  'N'],
        ['Chicago',   '[50-80]',  'M',  'N'],
        ['New York',  '[50-80]',  'F',  'N'],
        ['Chicago',   '>80',      'F',  'Y']
       ]
```

We can create an ID3 Decision tree:

```
id3 = ID3.new(DATA_SET, DATA_LABELS)
```

The Decision tree will automatically create the "rules" to parse new data, and identify new posible marketing targets:

```
  id3.get_rules
   # =>  if age_range=='<30' then marketing_target='Y'
       elsif age_range=='[30-50)' and city=='Chicago' then marketing_target='Y'
       elsif age_range=='[30-50)' and city=='New York' then marketing_target='N'
       elsif age_range=='[50-80]' then marketing_target='N'
       elsif age_range=='>80' then marketing_target='Y'
       else raise 'There was not enough information during training to do a proper induction for this data element'
end
```

```
id3.eval(['New York', '<30', 'M'])
  # =>  'Y'
```

AI4R implements the ID3 algorithm (Quinlan) as one of its automatic classifiers. Given a set of preclassified examples, it builds a top-down induction of decision tree, biased by the information gain and entropy measure. [2]

2.6 PRISM algorithm (Cendrowska, 1987)

PRISM algorithm is proposed as an improvement to ID3 algorithm changing its principal induction strategy. The decision tree output of Quinlan's ID3 algorithm is one of its major weaknesses because it can be incomprehensible and difficult to manipulate. The PRISM algorithm which, although based on ID3, uses a different induction strategy to induce rules which are modular, thus avoiding many of the problems associated with decision trees.

Ruby source code:

Build a new Prism classifier. You must provide a DataSet instance as parameter. The last attribute of each item is considered as the item class.
We can evaluate new data, predicting its class. e.g.

```
classifier.eval(['New York',  '<30', 'F'])  # => 'Y'
```

The get_rules() method returns the generated rules in ruby code. e.g.

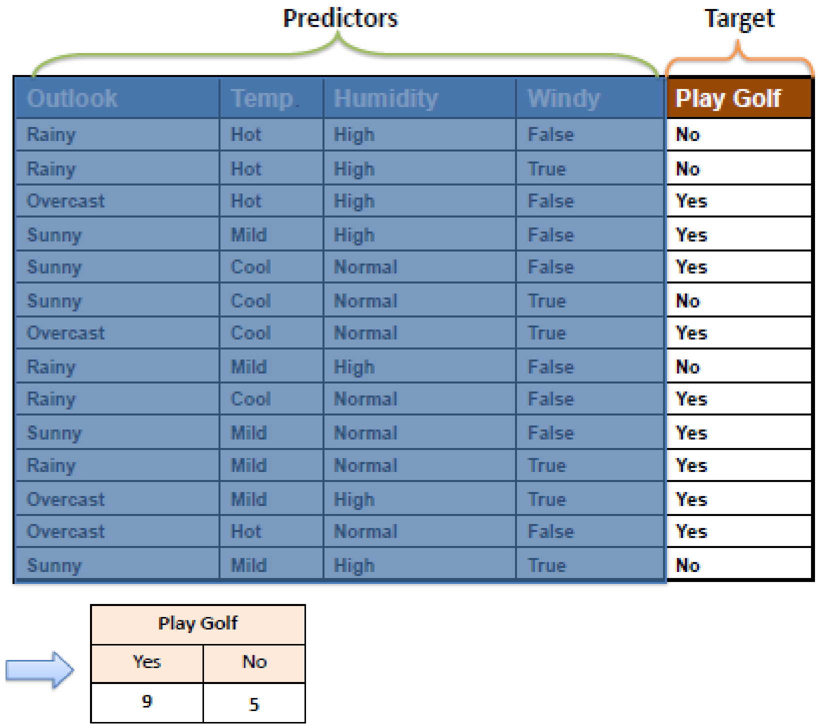
```
classifier.get_rules
  # => if age_range == '<30' then marketing_target = 'Y'
 elsif age_range == '>80' then marketing_target = 'Y'
 elsif city == 'Chicago' and age_range == '[30-50)' then marketing_target = 'Y'
 else marketing_target = 'N'
 end
```

A nice way to inspect induction results, and also to execute them:

```
     age_range = '[30-50)'
     city = 'New York'
     eval(classifier.get_rules)
     puts marketing_target
      'Y'
```

2.7 ZeroR algorithm

ZeroR is the simplest classification method which relies on the target and ignores all predictors.
The idea behind the ZeroR classifier is to identify the most common class value in the training set. It always returns that value when evaluating an instance. It is frequently used as a baseline for evaluating other machine learning algorithms.


Predictors
Target

Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Play Golf	
Yes	No
9	5

For build a new ZeroR classifier we must provide a DataSet instance as parameter. The last attribute of each item is considered as the item class.
We can evaluate new data, predicting its class. e.g.

```
classifier.eval(['New York',  '<30', 'F'])  # => 'Y'
```

The method get_rules() returns the generated rules in ruby code. e.g.

```
 classifier.get_rules
   # =>  marketing_target='Y'
```

A nice way to inspect induction results, and also to execute them:

```
   marketing_target = nil
   eval classifier.get_rules
   puts marketing_target
     # =>  'Y'
```

2.8 NaiveBayes Classifier

This is an implementation of a Naive Bayesian Classifier without any specialisation (ie. for text classification). Probabilities P(a_i | v_j) are estimated using m-estimates, hence the m parameter as second parameter when isntantiating the class.

The estimation looks like this:

$$(n\_c + mp) / (n + m)$$

the variables are:

```
n = the number of training examples for which v = v_j
 n_c = number of examples for which v = v_j and a = a_i
 p = a priori estimate for P(a_i | v_j)
 m = the equivalent sample size
```

stores the conditional probabilities in an array named @pcp and in this form

```
@pcp[attributes][values][classes]
```

This kind of estimator is useful when the training data set is relatively small.
If the data set is big enough, set it to 0, which is also the default value

Parameters

```
      *:m => Optional. Default value is set to 0. It may be set to a value
      greater than 0 when the size of the dataset is relatively small
```

How to use it

```
   data = DataSet.new.load_csv_with_labels "bayes_data.csv"
   b = NaiveBayes.new.
     set_parameters({:m=>3}).
     build data
   b.eval(["Red", "SUV", "Domestic"])
```

You can evaluate new data, predicting its category. e.g.

```
  b.eval(["Red", "SUV", "Domestic"])
    => 'No'
```

Calculates the probabilities for the data entry data. Data has to be an array of the same dimension as the training data minus the class column. Returns a map containint all classes as keys:

{Class_1 => probability, Class_2 => probability2 … } Probability is <= 1 and of type Float. e.g.

```
  b.get_probability_map(["Red", "SUV", "Domestic"])
    => {"Yes"=>0.4166666666666667, "No"=>0.5833333333333334}
```

3 Conclusions

As a conclusion to the things presented in this paper Ruby is a very powerful language for AI software solutions and a new alternative for AI programing besides consecrated languages like Lisp and Java. The information from this paper are just a smooth introduction to ruby paradigms for artificial intelligence programming that can be implemented in all sorts of applications from web to simple desktop task specialized programs.

References

[1] Uziel Sandler and Lev Tsitolovsky: Neural Cell Behavior and Fuzzy Logic. Springer 2008

[2] Artificial Intelligence for Ruby, http://ai4r.rubyforge.org/

[3] S. Russell and P. Norvig. Artificial Intelligence: A Modern Approach. Prentice Hall, third edition, 2009.

Volosincu Bogdan
“Lucian Blaga” University of Sibiu
Department of Mathematics and Informatics
No. 5-7, Dr. Ion Ratiu st., 550012 Sibiu
ROMANIA

Rusu Andrei
“Lucian Blaga” University of Sibiu
Department of Mathematics and Informatics
No. 5-7, Dr. Ion Ratiu st., 550012 Sibiu
ROMANIA

AUTHORS LIST

ALDULEA CRISTIAN-IONEL

Transilvania University of Brasov
Departament of Mathematics and Informatics
Str.Iuliu Maniu Nr. 50 Brasov , 500091
ROMANIA
E-mail: alduleacristi@yahoo.com

ANGHEL IOANA MADALINA

Transilvania University of Brasov
Department of Mathematics and Informatics
Street Iuliu Maniu, no. 50, Brasov
ROMANIA
E-mail: madalina.i_anghel@yahoo.com

BARAR ANA

"Politehnica" University of Bucharest
Splaiul Independentei 313, Bucuresti, 060042,
Corp JE 105
ROMANIA
E-mail: c.luthien@yahoo.com

BĂRBĂTEI IOANA ALEXANDRA

Polytechnic University of Bucharest
Department of Engineering in foreign languages
Splaiul Independentei, 313, sector 6, Bucharest
ROMANIA
E-mail: ioanalxndra@yahoo.com

BARKO PETRU

Transilvania Universitry of Brasov
Department of Mathematics and Informatics
Street Iuliu Maniu nr. 50, Brasov
ROMANIA
E-mail: petrubarko@yahoo.com

BECHERU ALEXANDRU PETRUȚ

University of Craiova
Faculty of Exact Sciences
Department of Computer Science
Str. A.I. Cuza, Nr. 13, Craiova
ROMANIA
E-mail: alex.becheru@inf.ucv.ro

BOTA FLORENTIN

“Lucian Blaga” University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: botaflorentin@yahoo.com

CAPUZZO FRANCESCO

University Lucian Blaga of Sibiu
Faculty of Sciences
5-7 I. Ratiu str., Sibiu, 550021
ROMANIA
Email: francesco.capuzzo@gmail.com

CULIC IOANA

Politehnica University of Bucarest
Faculty of Engineering in Foreign Languages
Splaiul Independentei 313
ROMANIA
E-mail: ioana mculic@yahoo.com

DUCA DIANA MARIA

“Lucian Blaga” University
Faculty of Sciences, Informatics
Sibiu, 5-7, Ioan Ratiu str.
ROMANIA
E-mail: duca.diana92@yahoo.com

FARCAŞ NICOLAE

Technical University Cluj-Napoca,
North University Center Baia-Mare
Str. Victoriei, 76
ROMANIA
E-mail: nfarcas@mail.com

FIERARU BOGDAN MIHAI

Transilvania University of Brasov
Departament of Mathematics and Informatics
Str. Iuliu Maniu Nr. 50 Brasov , 500091
ROMANIA
E-mail: fierarubogdan2000@yahoo.com

FILIPOV FILIP F.

Technical University of Sofia
Faculty of Computer Systems
1000 Sofia, blvd. Kl. Ohdiski 8
BULGARIA
E-mail: pilif.pi.lif@gmail.com

FLOREA ALEXANDRU IOAN

"Vasile Alecsandri" University of Bacau
Department of Informatics
Calea Marasesti, 157, 600115,
ROMANIA
E-mail: alexflorea89@gmail.com

GADZHEV DIMITAR

Technical University of Sofia
Dept. Computer Systems
8 Kliment Ochridsky blv.
BULGARIA
E-mail: dimityr.gadjev@gmail.com

KALOYAN MIRONOV

University of Ruse
Informatics and information technology
#8 Studentska Str.
BULGARIA
E-mail: mail@kaloian.net

KOTSEV VLADIMIR

Ruse University
Department of Informatics and Information Technologies
8 Studentska Str., Ruse
BULGARIA
E-mail: vladimir.kotsev@gmail.com

LAZĂR VALENTINA

”Lucian Blaga” University of Sibiu
Faculty of Sciences
No. 5-7, Dr. Ion Ratiu st., 550012 Sibiu
ROMANIA
E-mail: lyah_valy@yahoo.com

LIENKO DMYTRO

Kharkiv National University of Economics
Department of Information Systems
Kharkiv, 9a Lenin Ave.
UKRAINE
E-mail: dmytrolenko@gmail.com

MAMINA LAURA NICOLETA

Transilvania University of Brasov
Applied Informatics
Str. Iuliu Maniu, nr. 50, Brasov
ROMANIA
E-mail: laura.mamina@xu.unitbv.ro

MANEA MIHAI

"Vasile Alecsandri" University of Bacau
Department of Informatics
Calea Marasesti, 157, 600115,
ROMANIA
E-mail: manea_a_mihai@yahoo.com

MANTA ALIN

"Lucian Blaga" University of Sibiu,
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: mantalin@gmail.com

MOŞNEAG DIANA ALINA

“Lucian Blaga” University
Faculty of Sciences, Informatics
Sibiu, 5-7, Ioan Ratiu str.
ROMANIA
E-mail: mosneag.diana@yahoo.com

NECHIFOR VASILE NICUŞOR

"Lucian Blaga" University of Sibiu,
Faculty of Sciences
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: vasilenicusor@yahoo.com

OBANCEA DRAGOŞ IULIAN

Transilvania University from Braşov
Faculty of Mathematics and Informatics
Informatics Department
Street Iuliu Maniu, No. 50, Braşov
ROMÂNIA
E-mail: obancea_dragos@yahoo.com

PANASENKO SVETLANA A.

Kharkiv National University of Economics
Information Systems
61001, Kharkiv, Lenina Avenue 9A
UKRAINE
E-mail: pansvetik6@yandex.ru

PARAICU IULIA

University Politehnica of Bucharest
Faculty of Engineering in Foreign Languages
Splaiul Independentei 313, Bucharest, 060042
ROMANIA
E-mail: iulia.paraicu@gmail.com

PINTEA ALEXANDRU

National College "Emil Racoviță"
Str. M. Kogalniceanu, 9-11
Cluj-Napoca,
ROMANIA
E-mail: al_pintea@yahoo.com

PÎRVU DORU

"Lucian Blaga" University of Sibiu
Computer Science
Dr. I. Rațiu Street No.7-9, Sibiu, 550012
ROMANIA
E-mail: pirvu.doru@ymail.com

POPA ADRIAN

"Lucian Blaga" University of Sibiu
Faculty of Sciences, Informatics
Sibiu, 5-7, Ioan Ratiu str.
ROMANIA
E-mail: ad_robotics@yahoo.com

RADU SORIN DANIEL

"Lucian Blaga" University of Sibiu,
Faculty of Sciences
Department of Informatics
5-7 Dr. Ioan Ratiu, Sibiu, 550012
ROMANIA
E-mail: radu_s_o_r_i_n@yahoo.com

RANKOVIC NIKOLA

University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: nikola.s.rankovic@gmail.com

ROȚ RADU MARIN

“Lucian Blaga” University
Faculty of Sciences, Informatics
Sibiu, 5-7, Ioan Ratiu str.
ROMANIA
E-mail: radu_rot8@yahoo.com

ROTARI RADU

"Vasile Alecsandri"
University of Bacau
Department of Informatics
Calea Marasesti, 157, 600115,
ROMANIA
E-mail: rotari.radu@gmail.com

RUSU ANDREI

”Lucian Blaga” University of Sibiu
Faculty of Sciences
Computer Department
Adress: No. 5-7, Dr. Ion Ratiu st., 550012 Sibiu,
ROMANIA

SLAVOV FILIP

Technical University of Sofia
Computer Systems and Technologies
Sofia 1000, bul. “Kl. Ohridski” 8
BULGARIA
E-mail: nibblebyte@mail.bg

STANCU MIHAI

“Lucian Blaga” University of Sibiu
Computer Science
Dr. I. Rațiu Street No.7-9, Sibiu, 550012
ROMANIA
E-mail: stancu_meehigh@yahoo.com

ŞTEFAN ALEXANDRU

University Politehnica of Bucharest
Faculty of Engineering in Foreign Languages
Splaiul Independentei 313, Bucharest, 060042
ROMANIA
E-mail: stefan_alexandru_2005@yahoo.com

SUCIU GEORGE

Spiru Haret University Bucharest
ROMANIA
E-mail: george@beia.ro

SUKHONOS MAKSYM

Kharkiv National University of Economics
Department of Information Systems
Kharkiv, 9a Lenin Ave.
UKRAINE
E-mail: maksim.sukhonos@gmail.com

SZASZ ZSIGMOND-ATTILA

Transilvania University of Brasov
Departament of Mathematics and Informatics
Str.Iuliu Maniu Nr. 50 Brasov , 500091
ROMANIA
E-mail: zsigmond.szasz@xu.unitbv.ro

SZORNYI-CHIS IZABELLA

Transilvania University of Brasov
Departament of Mathematics and Informatics
Str.Iuliu Maniu Nr. 50 Brasov , 500091
ROMANIA
E-mail: izabella.szornyi@xu.unitbv.ro

TAMASI ZOLTAN

Transilvania University of Brasov
Departament of Mathematics and Informatics
Str.Iuliu Maniu Nr. 50 Brasov , 500091
ROMANIA
E-mail: tamasi_zoltan22@yahoo.com

TASIC JELENA

University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: jelenatasicsd@yahoo.com

TECŞA SIMONA

“Lucian Blaga” University of Sibiu
Dr. I. Rațiu Street No.7-9, Sibiu, 550012
ROMANIA
E-mail: tecsa_simona@yahoo.com

TUBA ANA

University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: anatuba89@gmail.com

VASILE VLAD

Transilvania University of Brasov
Departament of Mathematics and Informatics
Str. Iuliu Maniu Nr. 50 Brasov , 500091
ROMANIA
E-mail: vlad.vasile91@yahoo.com

VISNIUC GEORGE VIOREL

"Vasile Alecsandri" University of Bacau
Department of Informatics
Calea Marasesti, 157, 600115,
ROMANIA
E-mail: visniuc_george@yahoo.com

VOLOSINCU BOGDAN

”Lucian Blaga” University of Sibiu
Faculty of Sciences
No. 5-7, Dr. Ion Ratiu st., 550012 Sibiu,
ROMANIA
E-mail: bogdan_voloshincu@yahoo.com

WANDSCHNEIDER JOSEF-ALEXANDER

”Lucian Blaga” University of Sibiu
Faculty of Sciences, Informatics
Sibiu, 5-7, Ioan Ratiu str.
ROMANIA
E-mail: alexander91@versanet.de

Sponsors:

EBS Romania – Expectations Beyond Software

OMERON TECHNOLOGIES

Pentalog ROMANIA

ROPARDO

Top Tech

PAN FOOD

CRISPUS PUB