

Ahmed Yousify

Eye-Reaction: A Real-Time Video Based on Color Eye-Tracking System for Patients with Disabilities

Bachelor Thesis

YOUR KNOWLEDGE HAS VALUE



- We will publish your bachelor's and master's thesis, essays and papers
- Your own eBook and book - sold worldwide in all relevant shops
- Earn money with each sale

Upload your text at www.GRIN.com
and publish for free



Bibliographic information published by the German National Library:

The German National Library lists this publication in the National Bibliography; detailed bibliographic data are available on the Internet at <http://dnb.dnb.de> .

This book is copyright material and must not be copied, reproduced, transferred, distributed, leased, licensed or publicly performed or used in any way except as specifically permitted in writing by the publishers, as allowed under the terms and conditions under which it was purchased or as strictly permitted by applicable copyright law. Any unauthorized distribution or use of this text may be a direct infringement of the author's and publisher's rights and those responsible may be liable in law accordingly.

Imprint:

Copyright © 2014 GRIN Verlag
ISBN: 9783668123199

This book at GRIN:

<https://www.grin.com/document/283044>

Ahmed Yousify

Eye-Reaction: A Real-Time Video Based on Color Eye-Tracking System for Patients with Disabilities

GRIN - Your knowledge has value

Since its foundation in 1998, GRIN has specialized in publishing academic texts by students, college teachers and other academics as e-book and printed book. The website www.grin.com is an ideal platform for presenting term papers, final papers, scientific essays, dissertations and specialist books.

Visit us on the internet:

<http://www.grin.com/>

<http://www.facebook.com/grincom>

http://www.twitter.com/grin_com

University of Zakho

EYE-REACTION

A REAL-TIME VIDEO BASED ON COLOR EYE-TRACKING SYSTEM FOR
PATIENTS WITH DISABILITIES

Bachelor's Thesis

*A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor
of Science in Computer Science*

By

Ahmed Loqman Mustafa

Department of Computer Science
2014

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

(بج چ)

(چ چ چ)

(چ چ چ)

(چ چ چ)

(ی ی ی)

[الرحمن : 1-5]

ABSTRACT

The influence of vision is taking one-step ahead with the introduction of sophisticated eye-tracking and gaze-tracking techniques, which track the movement of the eye and the gaze location to control various applications. This paper describes in detail the low-cost hardware development and the software architecture for a real-time eye tracking based on color system using the open-source image-processing framework of AForge.NET. The designed system uses a USB camera to detect the eye movements and uses EuclideanColorFiltering to filter colors, and then it detects the biggest object between them and gets the location of that object. The system shows to the patient some question and answers according to the questions, he will look at one of the answers, then, it will be highlighted. By this way, the doctor will know what is wrong with the patient and what should be done to cure him. This provides a highly useful alternative control system for a person suffering from ALS or in Vegetative state of brain stroke. The software system is in Kurdish language and it is designed for the patient in a way to be relaxed and be as simple as possible. Test results on systems on the correct locations and error percentage shows the performance of the developed eye tracking system.

DEDICATION

If dedication can express even a part of the fulfillment of loyalty, then

To... the teacher of humans and the source of science

Our Prophet Muhammad (peace be upon him)

To... Like fatherhood to the Supreme ... my dear father

To... the first love of my life ... My compassionate Mother

To.... Love and all the love ... my brothers....

To All family and friends

To those who smoothed the way in front of me to get to the peak of science

ACKNOWLEDGEMENTS

My first and great thanks to The Almighty Allah for blessing, protecting and guiding me throughout this period. I could never have accomplished this without the faith I have in him.

I express my deep gratitude to my supervisor and promoter. Dr.Maiwan B.Abdulrazaaq, for his constant guidance, advice, support and motivation.

I also wish to acknowledge to the head of the computer science department Dr. Nawzat Sadiq Ahmed who simplified the way of knowledge, and provided the required facilities of this research, and to all staff members of computer science department.

TABLE OF CONTENTS

Abstract.....	I
Dedication.....	II
Acknowledgements.....	III
Table of Contents.....	IV
List of Figures.....	VI
List of Tables.....	VII
Chapter One: Introduction.....	1
1. Background.....	1
2. Eye Tracking Metrics Most Commonly Reported in Usability Studies.....	2
3. Problem statement.....	4
4. Objectives of the research.....	4
5. Questions of the research.....	4
6. Scope of the research.....	5
7. Motivation of the research.....	5
8. Significant of the research.....	5
9. Research project outlines.....	6
Chapter Two: Literature Review.....	7
Chapter Three: Methodology.....	11
1. Introduction.....	11
2. Participants and Environment.....	11
3. Tools.....	12
I. Visual Studio.....	12
II. Visual C#.....	13
III. Aforge.NET.....	13
1) Comparing Image-Processing Libraries.....	14
A. Documentation and other material.....	14
B. Ease of use.....	14
C. Performance.....	15
4. Materials and instruments.....	16

I. Sunglasses	16
II. Power Plug Adapter	17
III. Webcam	17
IV. DATASHOW (Projector)	18
V. Final Appearance	19
5. Method	19
6. Limitations.....	23
I. Speed	24
II. Lighting changes	24
III. Transformations	24
1) Scaling.....	25
2) Rotation:	25
7. Ethical considerations.....	25
Chapter Four: Software System	26
1. Log in Form.....	26
2. Configuration form.....	28
3. Questions Form	31
4. Help and Sleep.....	35
5. About.....	38
Chapter Five: Results of the research	39
1. Theoretical Results	39
2. EXPERIMENTAL RESULTS	39
Chapter Six: Conclusions and future works.....	43
References.....	44

LIST OF FIGURES

Figure 1: Patients room environment and position	11
Figure 2: Microsoft Visual Studio1	12
Figure 3: grayscale processing and binarization	16
Figure 4: Sunglasses	16
Figure 5: Power Plug Adapter	17
Figure 6: Webcam	18
Figure 7: Data Show (Projector)	18
Figure 8: The Final Appearance of the eye-tracking device	19
Figure 9: General Structure of Eye-tracking system algorithm	20
Figure 10: General Structure of Eye-tracking system algorithm (contd.)	21
Figure 11: Initial image	22
Figure 12: Result image	23
Figure 13: Log in Form	26
Figure 14: Login form class diagram	27
Figure 15: flowchart of login form	27
Figure 16: Configuration form	28
Figure 17: Configuration form class diagram	29
Figure 18: Flowchart of configuration form	30
Figure 19: Questions form	31
Figure 20: Questions form class diagram	32
Figure 21: Two answers question	33
Figure 22: Four answers question	34
Figure 23: Help and Sleep form	35
Figure 24: Help and Sleep form class diagram	36
Figure 25: Alarm Window	37
Figure 26: Sleep Window	37
Figure 27: About form (instructions)	38
Figure 28: About form (about)	38

LIST OF TABLES

Table 1: Statistics of Brain Storks in Zakho section in 2013.....	3
Table 2: AForge.NET Framework for Image Processing.....	13
Table 3: DOCUMENTATION AND OTHER MATERIALS.....	14
Table 4: EASE OF USE.....	14
Table 5: Top Left Location for points ($X > 20$ & $Y < 90$).....	39
Table 6: Bottom Left Location for points ($X > 20$ & $Y > 90$).....	40
Table 7: Top Right Location for points ($X < 20$ & $Y < 90$).....	41
Table 8: Top Right Location for points ($X < 20$ & $Y > 90$).....	41

CHAPTER ONE: INTRODUCTION

1. BACKGROUND

Several people around the world suffer from several physical disabilities that prohibit them from leading a normal life. Several conditions like Amyotrophic Lateral Sclerosis (ALS), cerebral palsy, traumatic brain injury, or stroke may result in loss of muscle movements in the body [1], thus rendering the person paralyzed. However, in several cases, not all parts of the body are paralyzed and the person may have limited voluntary movements of the head, eyes or tongue and may even be able to blink, wink or smile [2] [3]. In our case, we have vegetative state; it is an enhanced state of brain store.

Assistive technology systems, that make use of these available movements, can be developed to aid people suffering from these conditions not only communicate with other people, but also control various electronic devices [4]. For people suffering from quadriplegia, one of the only reliable and convenient sources of voluntary movement that may be used by an assistive technology system is the movement of the eye [5]. Hence, the person's eye movements may be used to control computers for browsing the internet, simple word processing or even paint like application [3].

In this paper, we present A Real-Time Video Based on Color low-cost Eye-Tracking System for patients With Disabilities called Eye-Reaction. The Eye-Reaction system lets the patient answer questions that are addressed to him by the doctor, and gives the patient a chance to be more comfortable by letting him decide whether someone stays with him or not, and the chance to call the doctor when needed for emergency cases. In the future, the Eye-Reaction system aims to provide a single control system to help people suffering from disabilities to continue with their day-to-day activity, helping them to move around without being dependent on somebody and control various household utilizations.

2. EYE TRACKING METRICS MOST COMMONLY REPORTED IN USABILITY STUDIES

The usability researcher must choose eye-tracking metrics that are relevant to the tasks and their inherent cognitive activities for each usability study individually. To provide some idea of these choices and concepts, we have attempted to use a common set of definitions as follows: [6]

Fixation: The moment when the eyes are relatively stationary, taking in or “encoding” information. Fixations last for 218 milliseconds on average, with a range of 66 to 416 milliseconds [7].

Gaze Duration: Cumulative duration and average spatial location of a series of consecutive fixations within an area of interest. Gaze duration typically includes several fixations and may include the relatively small amount of time for the short saccades between these fixations. A fixation occurring outside the area of interest marks the end of the [6].

Area of interest: Area of a display or visual environment that is of interest to the research or design team and thus defined by them (not by the participant) [6].

Scan Path: An eye-tracking metric, usually a complete sequence of fixations and interconnecting saccades [7].

Bellow, we present some statistics of peoples with brain stork injuries, the number of males and females and the number of deaths and patients, all in 2013 for Zakho city in Kurdistan region.

Table 1: Statistics of Brain Storks in Zakho section in 2013

Month		less than a year		from 1 to 4		from 5 to 9		from 15 to 49		from 50 and older		Total		
		Male	Female	Male	Female	Male	Female	Male	Female	Male	Female	Male	Female	Total
January	Patients							6	6	9	16	15	22	37
	Deaths								1	3	8	3	9	12
February	Patients				1			3	4	14	13	17	18	35
	Deaths									1	5	1	5	6
March	Patients				1			4	3	6	10	10	14	24
	Deaths							1	1	6	4	7	5	12
April	Patients							2	4	6	10	8	14	22
	Deaths									4	7	4	7	11
May	Patients							3	3	6	8	9	11	20
	Deaths									4	6	4	6	10
June	Patients							5	3	7	16	12	19	31
	Deaths									1	2	1	2	3
July	Patients							3	4	13	11	16	15	31
	Deaths									4	6	4	6	10
August	Patients							4	4	6	13	10	17	27
	Deaths									3	1	3	1	4
October	Patients							2		3	4	5	4	9
	Deaths										2	0	2	2
September	Patients							1		8	10	9	10	19
	Deaths									4	3	4	3	7
November	Patients							1	1	7	4	8	5	13
	Deaths									2	2	2	2	4
December	Patients							6	5	9	8	15	13	28
	Deaths									4	3	4	3	7
Total	Patients	0	0	0	2	0	0	40	37	94	123	134	162	296
	Deaths	0	0	0	0	0	0	1	2	36	49	37	51	88
	Total	0	0	0	2	0	0	41	39	130	172	171	213	384

3. PROBLEM STATEMENT

- 1- Around 88 persons died because of brain strokes in 2013 in Zakho.
- 2- We have patients we cannot communicate with them to know their conditions.
- 3- We cannot identify their needs.
- 4- We cannot tell exactly where they suffer.
- 5- We cannot find out whether they feel something or not.
- 6- We cannot find out when they want to rest or to express their thoughts.

4. OBJECTIVES OF THE RESEARCH

- 1- Helping ALS patients interact with their doctors using an easy-going system.
- 2- This device maybe a reason that the patient could be cured.
- 3- Getting new technology inside our hospitals in Kurdistan region.
- 4- Create and use a new technology that has not been used in Kurdistan.
- 5- This technology can be used in other fields rather than healthcare.

5. QUESTIONS OF THE RESEARCH

How are alternative interaction methods evaluated and compared to identify those that work well, and deserve further study, and those that work poorly, and should be discarded?

These are some sorts of questions can be answered with a valid and robust methodology for evaluating eye trackers for computer input.

Who will use this application? Where will it be used? Is it simple to use, or complex? Does it require high technology? Is it too much expensive or cheap to buy? Does it need an expert to use it? Or anyone can use it? When an eye tracker is used for computer input, how well does the interaction work? Can common tasks be done efficiently, quickly, accurately? What is the user's experience?

6. SCOPE OF THE RESEARCH

The Eye-Reaction system is created and designed for peoples who suffer from several disabilities and can only move their eyes. The application created in this research is made in Kurdish and English languages, English when only the doctor uses it, and Kurdish when the patient use it. When the patient answers some questions; he needs to know Kurdish reading to do that, but if he can hear the doctor, then it is not a problem because then the doctor will ask him vocabulary and the patient will try to answer.

7. MOTIVATION OF THE RESEARCH

The motivation of this research conveyed from a need to help patients with disabilities. Uses for such detection span many markets-from hands-free mouse interfaces for the disabled, to pilot cognitive activities during flight for military applications or healthcare applications just like our system. Advertisement agencies use such devices to study what people focus on when they observe the advertisers' products.

Other important applications have developed to maintain or monitor operator alertness in nuclear power plants as well as in trucks and other vehicles. Such research relies on the eye as a simple communicator of human interest or awareness through which a computer can understand and respond to a person's needs. In addition, the data formulated by a computer during a subject's use of this equipment can prove valuable for studies into the human mind by using the insight that eye motion can provide.

Giving hope to patients and knowing that if I make such a device, in somehow it will help them in their lives was another main motivation for me to make this research.

8. SIGNIFICANT OF THE RESEARCH

- 1- Patients can communicate easily with others.

- 2- Without it, the communication between the doctors and the patient or the patient and the peoples is very difficult.
- 3- We are able to find things and reasons, which were not previously known.
- 4- Its importance is to give the patient relaxation, confidence and be safe by fulfilling their requests.
- 5- Gives the patient hope and peace.

9. RESEARCH PROJECT OUTLINES

In chapter 1, it gives a general introduction on eye tracking systems, where and when it is used in real time environment, the problem statement, the research questions, and the scope of the research. Significance of the research are also being presented in this chapter.

Eye tracking and object tracking are in various fields like computer vision, robotics, artificial intelligent systems, surveillance systems, navigation systems, traffic-monitoring systems in heavy crowded urban areas. Chapter 2 gives some examples eye tracking systems that have been used previously. In chapter 3 it discusses the methodology of the research, what has been used, which tool were chosen and how did it accomplish and answered the research questions.

In chapter 4 different steps on how the software program works, and presents the flowcharts on how each form works. In chapter 5, it presents the results of the research, what we have and what came out through all of the experiments. Finally, in chapter 6, it talks about the conclusions and the future works.

CHAPTER TWO: LITERATURE REVIEW

This chapter presents the literature review on early eye tracking systems, new eye tracking systems and its applications.

- 1- In 1995, Xangdong Xie, Raghavan Sudhakar et al. in [8] Proposed an efficient approach for real-time eye feature tracking from a sequence of eye images. first formulated a dynamic model for eye feature tracking, which relates the measurements from the eye images to the tracking parameters. In the model, the center of the iris was chosen as the tracking parameter vector and the gray level centroid of the eye was chosen as the measurement vector. The used procedure for evaluating the gray level centroid, the preprocessing step such as edge detection and curve fitting was needed to be performed only for the first frame of the image sequence. A discrete Kalman filter was then constructed for the recursive estimation of the eye features, while taking into account the measurement noise. Experimental results were presented to demonstrate the accuracy aspects and the real-time applicability of the proposed approach.
- 2- In 2000, Takashi AOKI, Goh HIDAKA, et al. in [9]. Proposed an approach to tracking control of multiple objects in plural hand-eye systems. Then the exchanging motion of target object was also introduced to realize the collision avoidance motion. The algorithm of the collision avoidance motion made it possible to obtain the dexterous motion in the plural hand-eye system and was one of the remarkable points of the proposed approach. Several simulations and experiments were implemented to confirm the validity of the proposed controller in the two hand-eye systems.
- 3- In 2002, Xia Liu, Fengllang Xu et al. in [10] Proposed a new method to improve eye tracking under various illumination conditions with and without sunglasses. The. System worked very well under non-infrared lighting and ordinary ambient infrared lighting.

- 4- In 2004, Louisa Pui Sum Ipy, Truong Q. Nguyen et al. in [11] proposed a real time internal eye tracking software based system. By combining intensity based object tracking Adaptive Mean Shift algorithm with Kalman filtering techniques together, a robust real time eye tracking algorithm was proposed. The system was a robust in terms of its acceptance to low resolution, coarsely quantized, or extremely noisy images and still was able to produce desirable results with no pre-processing of images.
- 5- In June 2005, Yin, Jun Wang and Lijun. [12]. Proposed a system for eye detection and tracking through matching the terrain features. The proposed system worked automatically using an ordinary webcam without special hardware's. The major contribution of this work lied in the proposed unique approach for topographic eye feature representation, by which both eye detection and eye tracking algorithms can employ the probabilistic measurement of eye appearance in the terrain map domain rather than the intensity image domain. Fitting function based on the mutual information to describe the similarity between terrain surfaces of two eye patches. With fairly small number of terrain types, the *p.d.f.* of marginal and joint distributions was easily estimated, and eventually, the eye location was determined by optimizing the fitting function efficiently and robustly.
- 6- In 2007, Zhiwei Zhu and Qiang Ji Proposed, in [13] proposed two novel techniques to improve the existing gaze tracking techniques based on exploiting eye's anatomy. First, a simple direct method was proposed to estimate the 3-D optic axis of a user without using any user-dependent eyeball parameters. The method was more feasible to work on different individuals without tedious calibration. Second, a novel 2-D mapping-based gaze estimation technique was proposed to allow free head movement and minimize the number of personal calibration procedures. Therefore, the eye gaze was estimated with

high accuracy under natural head movement, with the personal calibration being minimized simultaneously. By the novel techniques proposed two common drawbacks of the existing eye gaze trackers eliminated or minimized nicely so that the eye gaze of a user be estimated under natural head movement, with minimum personal calibration.

- 7- In 2009, Jianxiong Tang, Jianxin Zhang, in [14] Explored out the latest law of motion to overcome the shortcoming that a linear filter must assume the motion law in advance. Thus, it can achieve robust tracking of eye. The grey system did not need to build the track model. For predicting of maneuvering object, it had the advantages of less original detection data, lower calculation scale and higher prediction accuracy. Furthermore, it can build the real-time online grey prediction model under the polar coordinate system, making it unnecessary to convert the model. Therefore, it can be widely applied into tracking of Maneuvering objects.
- 8- In 2012, Li Li, Yanhao Xu, Andreas K"onig, in [15] Presented a novel approach to robust depth camera based multi-user eye tracking for auto stereoscopic displays. The proposed object tracking method, which was based on difference map of consecutive depth images, achieved superior results compared to state-of-the-art intensity image based tracking techniques with respect to tracking accuracy and occlusion handling. The eye positions of multiple users can be detected and tracked accurately with relatively low overall computational cost. An auto stereoscopic display system in combination of the proposed technique that supported multiuser head/eye tracking was able to provide immersive 3-D experience to multiple users without the need of glasses or other headgear. Currently they are involved in the development of a novel TOF based depth camera that can promise more compact design better energy efficiency for 3-D embedded vision systems, which aim at robot control, surveillance, driver assistance as well as consumer applications. Further validation of the pre-

- sented approach on current and emerging depth cameras, analysis of parameter sensitivity and adaptation to other applications will be addressed in future work.
- 9- In 2013, Robert Gabriel Lupu, Florina Ungureanu, et al. In [16] paper a reliable, mobile and low-cost system based on eye tracking mouse was presented. A head mounted device detected the eye movement and consequently the mouse cursor was moved on the screen. A click event denoting a pictogram selection was performed if the patient gazes a certain time the corresponding image on the screen.
 - 10- In 2013, Erik English¹, Alfredo Hung¹, et al., in [17] paper, they proposed the EyePhone framework, a mobile HCI that allows users to control mobile phones through intentional eye or facial movements. A proof of-concept prototype was developed based on a wearable electroencephalograph (EEG) headset and an Android smartphone. Specifically, a graphical window can receive and display continuous EEG data acquired from the headset; a mouse emulator allowed users to move a cursor around the smartphone screen by moving their heads and eyes; and an emergency dialer allowed users to make an emergency phone call through a certain pattern of eye/facial movements. The launching and switching of each functional module were also implemented through predefined head movement patterns, in order to achieve a true “hands-free” environment.
 - 11- Our proposed method is a low-cost hardware development and the software architecture of a real-time eye tracking based on color system using the open-source image-processing framework of AForge.NET. The system designed uses a USB camera to detect the eye movements. The system shows to the patient a question and answers according to the question, the patient then will look at one of the answers and it will be highlighted. This provides a highly useful alternative control system for a person suffering from ALS (i.e. full body paralysis). Test results on system and error percentage show the performance of the developed eye tracking systems.

CHAPTER THREE: METHODOLOGY

1. INTRODUCTION

The purpose of this study is to develop an algorithm that helps people around the world, who suffer from several physical disabilities. Whom they can move only their eyes to interact with their doctors. This chapter outlines the research methodology of this study. It summarizes the participants involved in this project, as well as the materials and instruments used. It describes the reasons of choosing the tools, also includes sections on limitations and ethical considerations. Moreover, the research questions examined.

2. PARTICIPANTS AND ENVIRONMENT

The participants in this study were patients in Zakho's General Hospital in Zakho City. The participant (patient) is not able to talk, and cannot move anything from his body and only capable of moving his eyes. This is an enhanced condition of brain stroke named vegetative state.

The experiments of this project were in Zakho's General Hospital. Each patient has his own room.

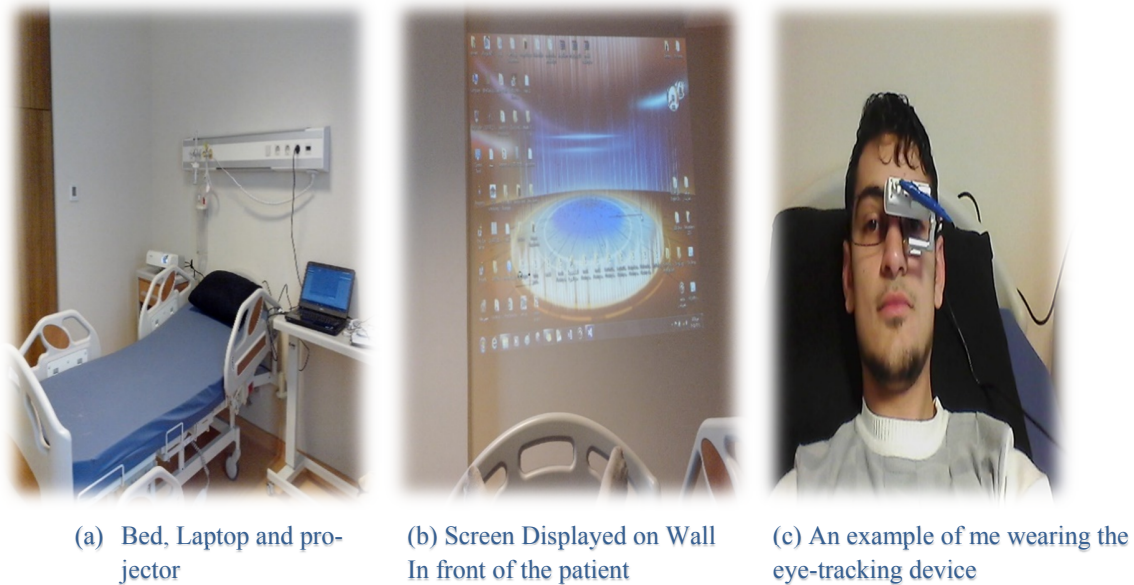


Figure 1: Patients room environment and position

3. TOOLS

I. VISUAL STUDIO

Visual Studio is a comprehensive collection of tools and services to help you create a wide variety of apps, both for the Microsoft platform and beyond. Visual Studio also connects all of your projects, teams, and stakeholders. Now your team can work with greater agility from virtually anywhere irrespective of development tool, including Eclipse and Xcode [18].

Visual Studio supports different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language provided a language-specific service exists. Built-in languages include C, C++ and C++/CLI (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010). Support for other languages such as M, Python, and Ruby among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Individual language-specific versions of Visual Studio also exist which provide more limited language services to the user: Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++. [19]



Figure 2: Microsoft Visual Studio¹

¹ You can find this picture in this website <http://www.visualstudio.com/> .

II. VISUAL C#

Microsoft C# (pronounced C sharp) is a new programming language designed for building a wide range of enterprise applications that run on the .NET Framework. An evolution of Microsoft C and Microsoft C++, C# is simple, modern, type safe, and object oriented. C# code it compiles as managed code, which means it benefits from the services of the common language runtime. These services include language interoperability, garbage collection, and enhanced security [20].

III. AForge.NET

"AForge.NET is an open source C# framework designed for developers and researchers in the fields of Computer Vision and Artificial Intelligence - image processing, neural networks, genetic algorithms, fuzzy logic, machine learning, robotics, etc." [21].

The framework has set of libraries; In the Following table the framework's features in this research that are being used are presented [3]:

Table 2: AForge.NET Framework for Image Processing

Namespace	Class	Method	Field/Property
AForge.Imaging	BlobCounter (used to get and specify dimensions of the blob)	GetObjectsRectangles ProcessImage	Field <ul style="list-style-type: none">• imageHeight• imageWidth
			Property <ul style="list-style-type: none">• MaxHeight• MinHeight• MaxWidth• MinWidth
AForge.Imaging.Filters	ColorFiltering (used to specify the RGB colour range)	Apply(Bitmap)	N/A
	Grayscale (convert bitmap image to grayscale image)	Apply(Bitmap)	Field <ul style="list-style-type: none">• Red• Green• Blue
AForge.Video.DirectShow	FilterInfoCollection (enumerate video devices)	N/A	N/A
	FilterCategory (get video devices)	N/A	N/A
	VideoCaptureDevice (locates the USB video device)	SignalToStop WaitForStop Start	Event <ul style="list-style-type: none">• NewFrame

1) COMPARING IMAGE-PROCESSING LIBRARIES

At present, there are some well-known and fully functional image-processing libraries, such as OpenCV, Emgu CV and AForge.NET. This part gives a brief comparison of these image-processing libraries concerning documentation, ease of use, performance, and so on [22].

A. DOCUMENTATION AND OTHER MATERIAL

The following table shows the comparison of documentation and other material:

Table 3: DOCUMENTATION AND OTHER MATERIALS.

Library	Book	Website	Documentation	Sample	Forum
OpenCV	Many	Multilanguage	Multilanguage	Many	Hot
Emgu CV	Few	English	English	Less	Poor
Aforge.NET	Few	English	English	Many	Poor

Since its alpha release in January 1999, OpenCV is been used in many applications, products, and research efforts. Therefore, it is easier for a beginner to find plenty of tutorials. Emgu CV is a .NET wrapper to the OpenCV image-processing library. Maybe we cannot find many samples of Emgu CV, but a better understanding of OpenCV also helps a lot. These two are just like brothers and share the same spirit.

The AForge.NET computer vision framework provides not only with different libraries and their sources but also with many sample applications, which demonstrate the use of the library. AForge.NET documentation's help files can be downloaded in HTML help format. [22]

B. EASE OF USE

The following table shows what library is most comfortable to work with:

Table 4: EASE OF USE.

Library	Ease of use
OpenCV	Not so friendly
Emgu CV	Good
Aforge.NET	Very good

Most of OpenCV's functions are in C style, but a few of them have a C++ class. After the 2.0 edition, more features have been encapsulated into the C++ classes. Emgu CV is a .NET wrapper to the OpenCV, which encapsulated most of the OpenCV features. AForge.NET is a pure .NET library, which means it is more user-friendly. [22] This is the reason why I choose to work with this library (Aforge.NET) to complete my thesis project.

C. PERFORMANCE

An image-processing library can do plenty of things. Here, we select the most basic Functions to test their performance. One is grayscale processing and the other is binarization. These two algorithms can be considered as representatives of the Average image-processing operations because most of the work in image processing Involves memory read/write and matrix operations. [22]

Here, we use five ways to achieve grayscale processing and binarization:

- Call OpenCV functions with C
- Call AForge.NET functions with C#
- Call Emgu CV with C#
- Use P/Invoke to call OpenCV in C# code
- Custom processing method in C#

Record the time and calculate the average of ten times of the execution period. Each method will get a runtime. We can represent these data graphically to make things clearer, as shown in the following diagram: [22]

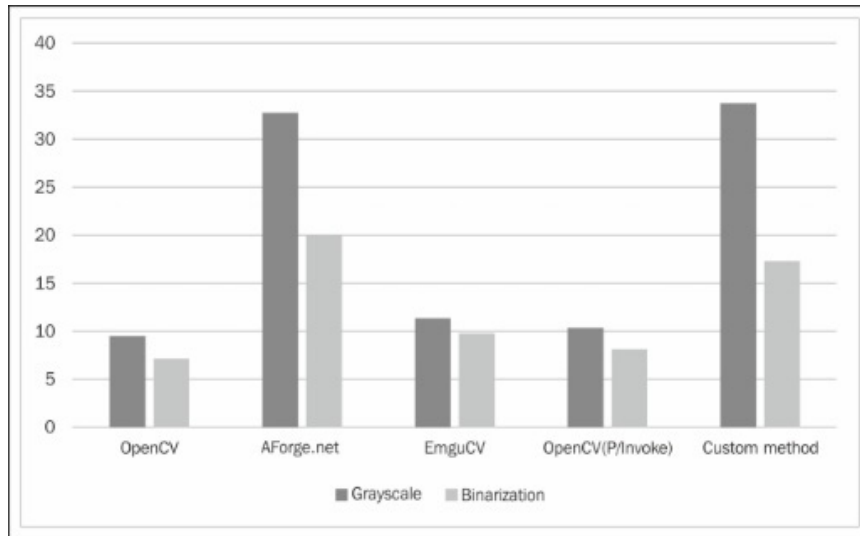


Figure 3: grayscale processing and binarization

4. MATERIALS AND INSTRUMENTS

To build the Eye-Reaction device and make our software work, we needed some instruments, which consists of a webcam, glasses to hold the webcam, a Data Show (projector) and some other simple things along with them.

I. SUNGLASSES

The Sunglasses are used to carry out the webcam, and the patient will just have to wear them to use it.



Figure 4: Sunglasses ²

² You can find this picture in this website <http://www.ebay.com/bhp/oakley-rx-glasses> .

II. POWER PLUG ADAPTER

In the patient's room, all the mounted adapters had two holes in it, so we had to get one that fits with the laptop and the Projector's cables.



Figure 5: Power Plug Adapter ³

III. WEBCAM

The Webcam is the main part for the project to work, with the webcam constructed to the glasses, the program will then be able to take the frames and do its work. It is built only to stare at one eye (left eye), leaving the other eye to watch the questions showing on the wall and answer them.

Some of the webcam specifications are:

- Webcam for both PC and Mac.
- You can place either on the table or on the screen.
- For Skype, MSN etc.
- Still image mode also with 5.0 Megapixel resolution.
- USB connection.

³You can find this picture in this website <http://www.blackbox.com/Store/Detail.aspx/Power-Plug-Adapter-U-S-to-Europe-the-Middle-East-Africa-Asia-and-South-America/MC167A> .



Figure 6: Webcam ⁴

IV. DATASHOW (PROJECTOR)

The idea behind placing A Data Show (projector) in the patient's room was to make the patient more comfortable and to see more clearly with big objects and words in front of him/her shown on the wall.



Figure 7: Data Show (Projector) ⁵

⁴ You can find this picture in this website http://cdon.fi/kodin_elektroniikka/grundig/grundig-webcam-high-performance-p22636971 .

⁵ You can find this picture in this website http://www.projectorpoint.co.uk/projectors/Epson_EMP-S5 .

V. FINAL APPEARANCE

This is the shape of the eye-tracking device after collecting and mounting the web camera with sunglasses. The webcam can be placed for the left eye or the right eye, but both eyes should be working correctly to get its position.



Figure 8: The Final Appearance of the eye-tracking device

5. METHOD

This main part of the algorithm is that it uses color to detect objects, when we used it for our purpose; it became an eye-tracking system.

The flow chart of this algorithm is on Figure9 and Figure10, the Following paragraphs are on how the eye-tracking algorithm works, and will give you explanations that are more detailed.

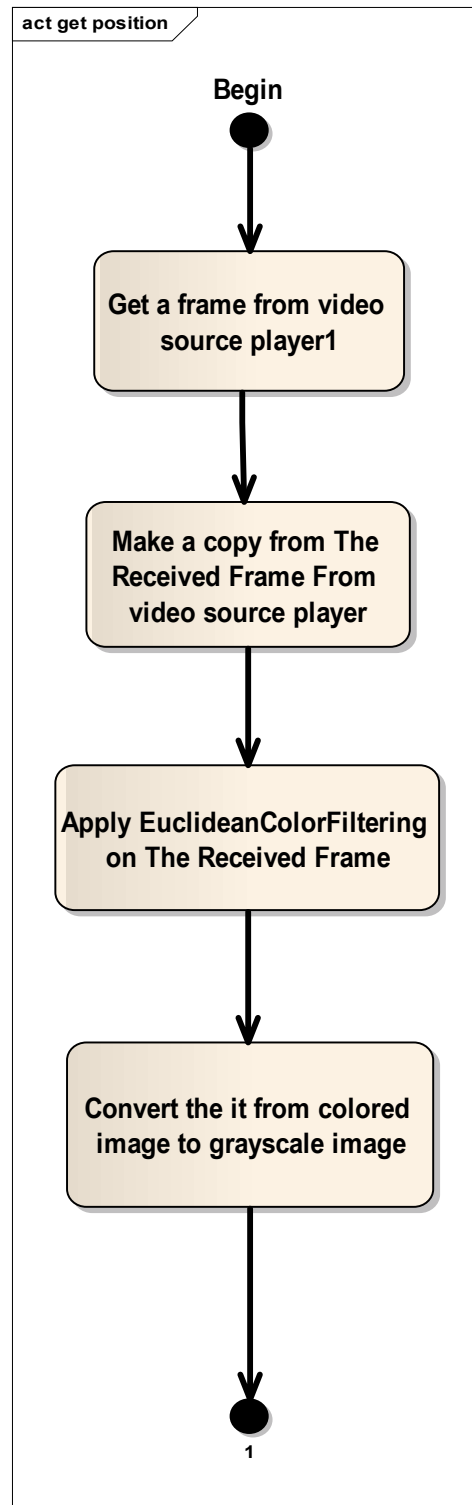


Figure 9: General Structure of Eye-tracking system algorithm

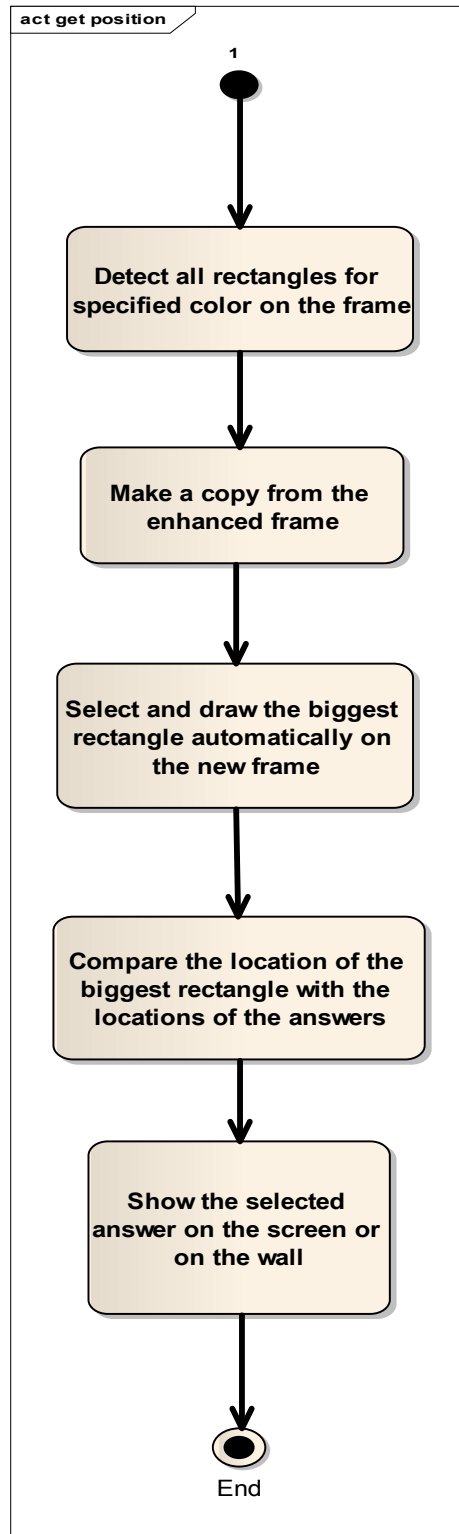


Figure 10: General Structure of Eye-tracking system algorithm (contd.)

The first step is to get a frame from the video source player (from the camera). Once it has the retrieved frame, it makes a copy of that frame, and then applies the EuclideanColorFiltering on it.

The filter filters pixels, which color is inside/outside of RGB sphere with specified center and radius - it keeps pixels with colors inside/outside of the specified sphere and fills the rest with specified color. The filter accepts 24 and 32 bpp color images for processing [23].

```
// create filter
```

```
EuclideanColorFiltering filter = new EuclideanColorFiltering ( );
```

```
// set center color and radius
```

```
filter.CenterColor = new RGB (215, 30, 30);
```

```
filter.Radius = 100;
```

```
// apply the filter
```

```
filter.ApplyInPlace (image);
```



Figure 11: Initial image⁶

⁶ You can find this picture in this website <http://www.aforogenet.com/framework/docs/html/67fa83b5-dede-8d3a-8d3b-b7a6b9859538.htm>



Figure 12: Result image⁷

The next step is to convert the frame (image) from colored image to grayscale image. Once the image is in grayscale mode, it will detect all the rectangles that has the same color that the user chose.

Make a copy from the enhanced frame, and then a new frame will contain the biggest rectangle that was inside the collection of rectangles for the specified color. Compare the location of the biggest rectangle with the locations of the answers. At last Show the selected answer on the screen or on the wall.

It does all these steps all over again to get another frame each time, because this algorithm is used for real-time video. It takes up to 25 frames per second.

6. LIMITATIONS

While we worked on this research project, some limitations came across our way, so this section covers those limitations. In addition, ways to avoid getting into them.

⁷ You can find this picture in this website <http://www.aforogenet.com/framework/docs/html/67fa83b5-dede-8d3a-8d3b-b7a6b9859538.htm>

I. SPEED

Because we are dealing with the memory, the speed is a limitation for us. Each time the algorithm brings a frame from the webcam and stores it in to the memory, then it makes a copy of it , do all the things with it and bring another one in less than a second, this all takes memory space (Ram) and sometimes makes the process of running the program go slow.

We are using this program application on a computer its processor specifications are Core i3, 2.54GHz, and a ram of 3G. so, to avoid getting slow while running the program application consider using a laptop with a ram of 4G or more for best performance.

II. LIGHTING CHANGES

Different illumination conditions have always been a problem with face and eye tracking between various lighting conditions, first, the color of the eye can change dramatically, and secondly the difference within the eye can change as well. This depends mostly on the type of the emitted light and depends on what kind of the light source is.

Different types of light are Daylight: This is probably the best kind of light because it usually is ambient than other types, but it can also be directional. Cameras usually perform best under this kind of natural light. Home light: Usually semi-bright light that can come from different directions. Screen light: The light that comes from the TV or computer screen. This light can have many different colors and is dependent on the content shown on the screen. [24] The kind of light that we are using is yellow light.

III. TRANSFORMATIONS

Transformations are when the face of the viewer changes due to movement of the viewer (or camera). The transformation conditions that are make difficulties for us has two types:

1) SCALING

This means that the viewer moves towards the screen or away from the screen. The optimal viewing distance is highly dependent on the type of display used. All The tests conducted in this project are done in a room with the projector is in a distance of approximately 330cm away from the wall. The height of a table that the projector will be upon it is about 87cm, and the distance between the head of the patient and the projector is almost 40cm. these are the measurements that I used, but anyone can change them according to the programming to get the best result.

2) ROTATION:

When the viewer rotates, the face can move along different axes and so as the eyes. For example looking to the right or left, or looking upwards or downwards. When this happens, the viewer is no longer looking at the display, so the position of the eyes is not that important when these kinds of rotations occur. [24]On the other hand, when a viewer tilts the face, he or she can still be watching the content on the display, so the eye tracker has to be able to keep tracking the eyes. The tracker has to be able to track the eyes up to 45 degrees of tilting. However, our eye tracker does not support 45 degrees, you have to adjust it in run time each time the face tilts, or if you do not adjust it, he, she can tilt their face to the same position as before and it will work as it was the first time.

7. *ETHICAL CONSIDERATIONS*

In this research, a percentage of its base tracking code were brought from the developer Mr. Kishor datta gupta in Bangladesh and had been modified and restructured to do its current work.

CHAPTER FOUR: SOFTWARE SYSTEM

In This chapter, it introduces the software and its features, what lays behind each form and the flowcharts of how every form works. For the patient to be conferrable when working with the program, the software program is designed in the best way. Because of our patients can only move their eyes, the design had to be made as simple as possible to fulfil that requirement.

1. LOG IN FORM

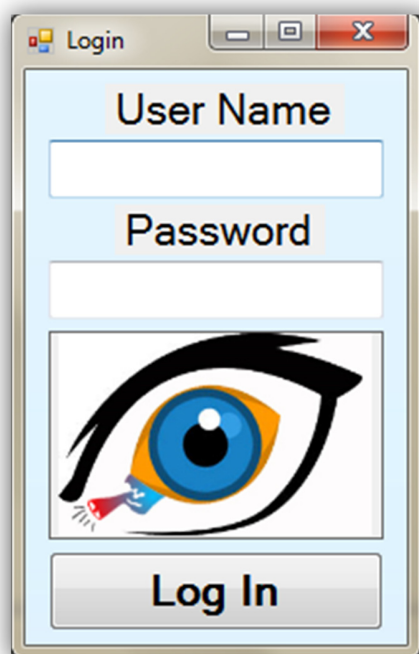


Figure 13: Log in Form

In this form, the user (doctor) will have to enter the username and the password that has been giving to him by the administrator or the programmer. If the username or the password is wrong, it will not enter the system, but gives the user a chance to try again until it is correct. Ones the username and the password are correct; it will enter the system.

For creating the login form and making it work, it needs some variables; in bellow figure14, it shows what programming variables that are used.

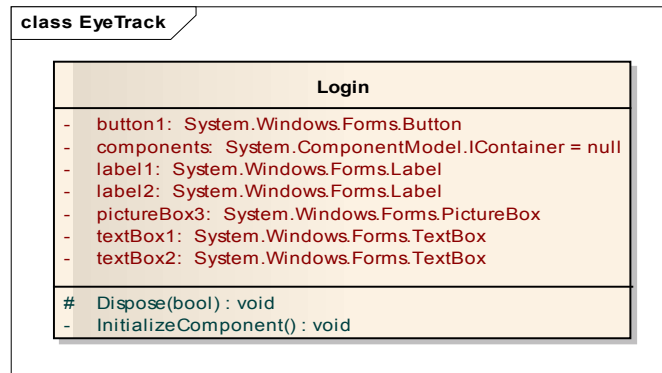


Figure 14: Login form class diagram

We present the flowchart of Login form in the bellow figure. When the username and the passwords are correct, it will then enter to the system directly to configuration form.

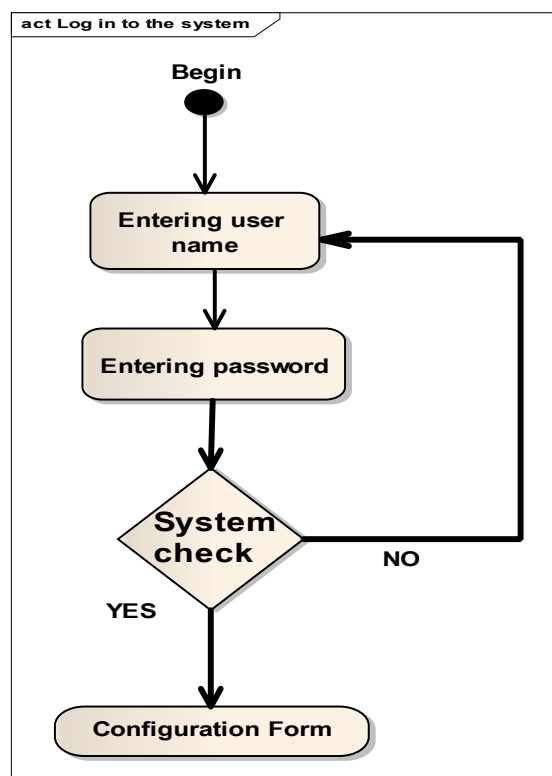


Figure 15: flowchart of login form

2. CONFIGURATION FORM



Figure 16: Configuration form

In configuration form, the system is entirely configured. There are two screens; the first one receives the video and the second one is a video being processed with filters. At first, the doctor or the user will have to choose the webcam that he is using, and then click on the start button to play the video. To detect the object, in our case to detect the eye, we have to choose the color, after selecting the color of the eye, if needed, the range is also selected to minimum the range of the detect object. In the right side of the form, a vertical box (Rich Textbox) shows the location of the rectangle that detected the eye.

The doctor has three options to select from the configuration form. Either to show the about form and see how the program works and see more informations about the creator of the program, or to ask questions to the patient that are important and are the main objectives of the research, or to leave the patient alone with the form help and sleep.

class EyeTrack

Configuration

- button1: System.Windows.Forms.Button
- button10: System.Windows.Forms.Button
- button2: System.Windows.Forms.Button
- button3: System.Windows.Forms.Button
- button7: System.Windows.Forms.Button
- button8: System.Windows.Forms.Button
- camerasCombo: System.Windows.Forms.ComboBox
- colorDialog1: System.Windows.Forms.ColorDialog
- components: System.ComponentModel.IContainer = null
- groupBox1: System.Windows.Forms.GroupBox
- groupBox2: System.Windows.Forms.GroupBox
- groupBox3: System.Windows.Forms.GroupBox
- label1: System.Windows.Forms.Label
- + numericUpDown1: System.Windows.Forms.NumericUpDown
- panel1: System.Windows.Forms.Panel
- pictureBox2: System.Windows.Forms.PictureBox
- pictureBox4: System.Windows.Forms.PictureBox
- + richTextBox1: System.Windows.Forms.RichTextBox
- videoSourcePlayer1: AForge.Controls.VideoSourcePlayer
- videoSourcePlayer3: AForge.Controls.VideoSourcePlayer

Dispose(bool) : void

- InitializeComponent() : void

Figure 17: Configuration form class diagram

In figure 17-it shows configuration class diagram, What It includes and what controls did we use.

Now in figure 18 it presents the flowchart of how the configuration form works from entering the configuration form until choosing one of the three choices.

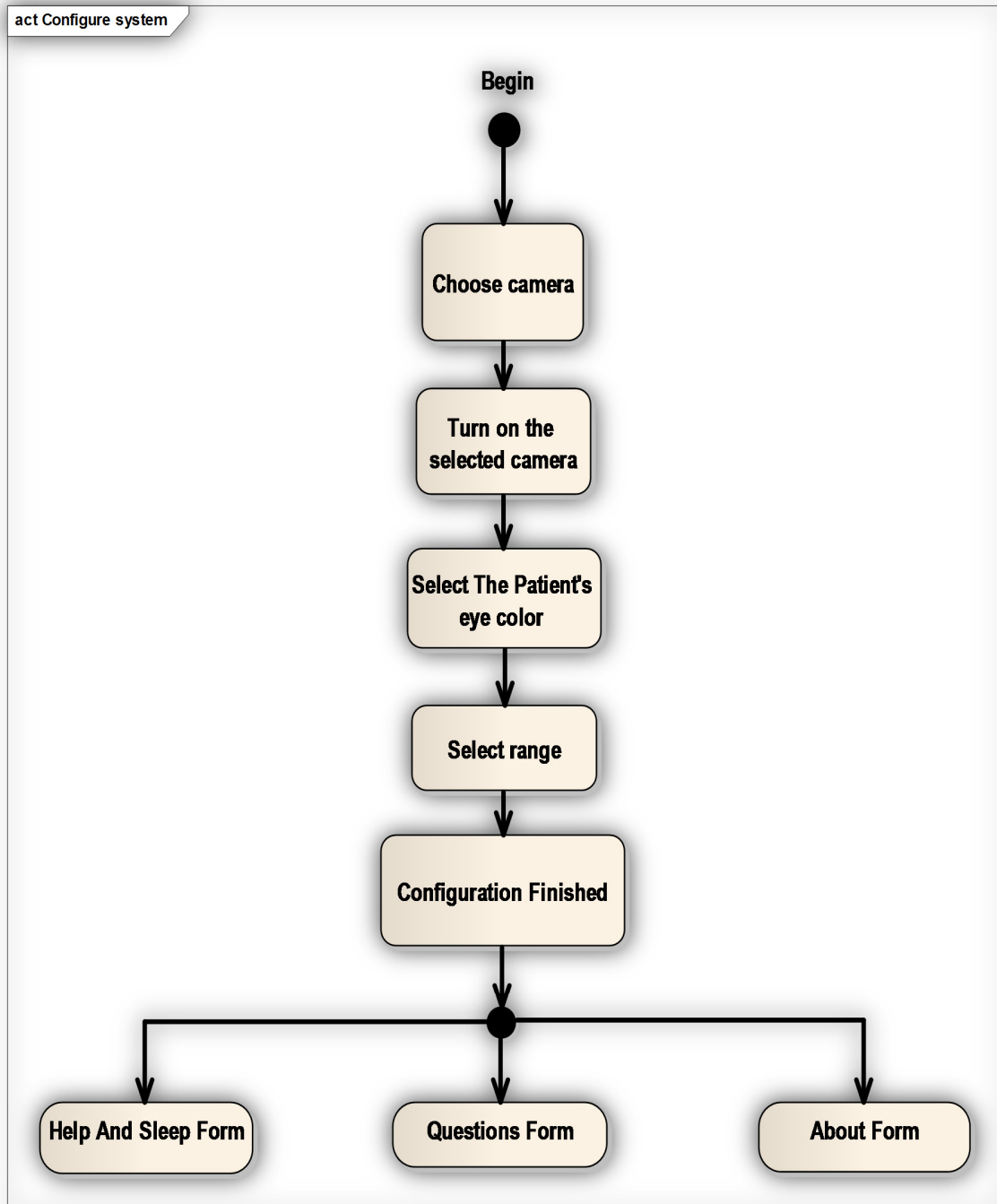


Figure 18: Flowchart of configuration form

3. *QUESTIONS FORM*

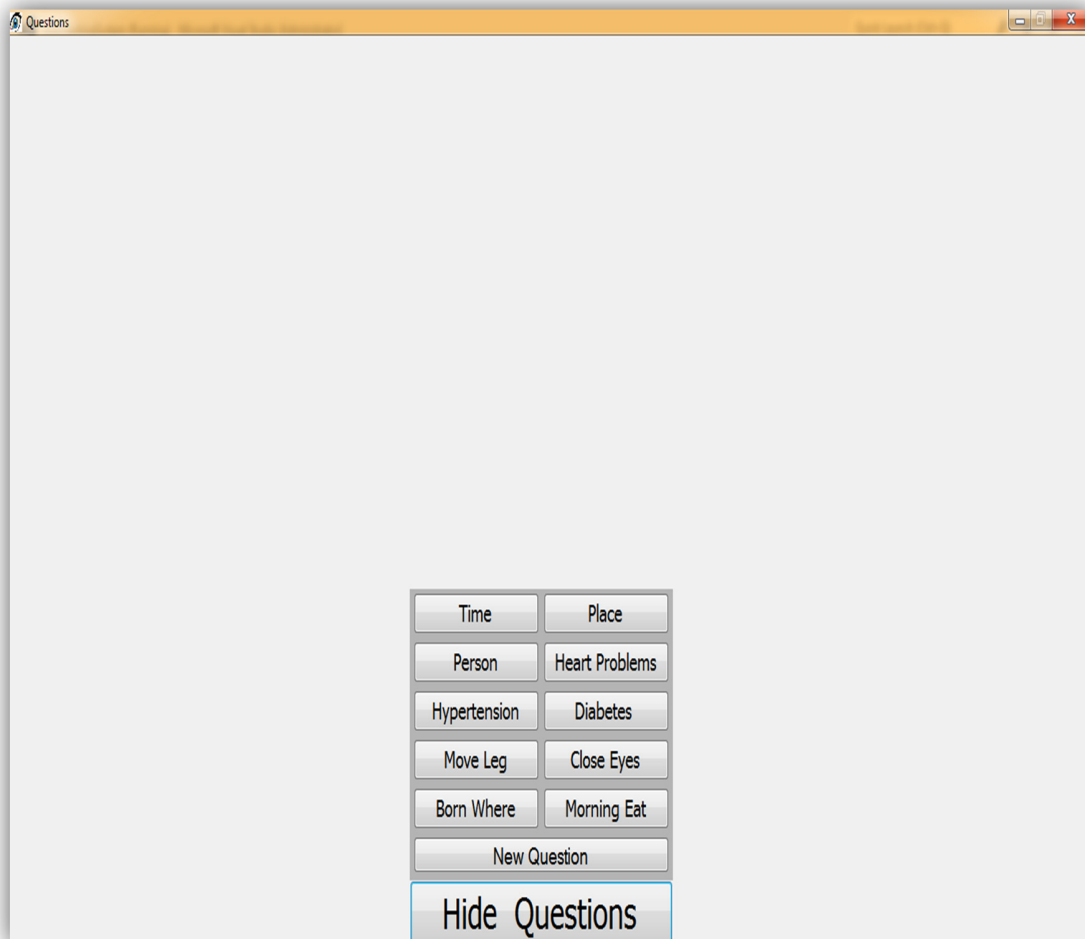


Figure 19: Questions form

If the doctor (user) selects the Questions form, and that's the main objective form of this program, the window in figure19 will appear, the doctor will select a question that he would like to ask the patient, or create a new one and submit it then the patient will try to answer by his/her eyes.

The answer highlights when the patient looks at it, and after the patient finishes from selecting an answer, the doctor or the supervisor will see what he/she chose.

class EyeTrack

Questions

```

- button1: System.Windows.Forms.Button
- button10: System.Windows.Forms.Button
- button11: System.Windows.Forms.Button
- button12: System.Windows.Forms.Button
- button13: System.Windows.Forms.Button
- button14: System.Windows.Forms.Button
- button15: System.Windows.Forms.Button
- button16: System.Windows.Forms.Button
- button17: System.Windows.Forms.Button
- button18: System.Windows.Forms.Button
- button19: System.Windows.Forms.Button
- button2: System.Windows.Forms.Button
- button20: System.Windows.Forms.Button
- button21: System.Windows.Forms.Button
- button22: System.Windows.Forms.Button
- button23: System.Windows.Forms.Button
- button24: System.Windows.Forms.Button
- button25: System.Windows.Forms.Button
- button26: System.Windows.Forms.Button
- button27: System.Windows.Forms.Button
- button28: System.Windows.Forms.Button
- button29: System.Windows.Forms.Button
+ button3: System.Windows.Forms.Button
- button30: System.Windows.Forms.Button
- button31: System.Windows.Forms.Button
- button32: System.Windows.Forms.Button
- button33: System.Windows.Forms.Button
- button34: System.Windows.Forms.Button
- button4: System.Windows.Forms.Button
- button5: System.Windows.Forms.Button
- button6: System.Windows.Forms.Button
- button7: System.Windows.Forms.Button
- button8: System.Windows.Forms.Button
- button9: System.Windows.Forms.Button
+ camerasCombo: System.Windows.Forms.ComboBox
+ colorDialog1: System.Windows.Forms.ColorDialog
- components: System.ComponentModel.IContainer = null
- groupBox1: System.Windows.Forms.GroupBox
- groupBox2: System.Windows.Forms.GroupBox
- groupBox3: System.Windows.Forms.GroupBox
- label1: System.Windows.Forms.Label
- label10: System.Windows.Forms.Label
- label11: System.Windows.Forms.Label
- label12: System.Windows.Forms.Label
- label2: System.Windows.Forms.Label
- label3: System.Windows.Forms.Label
- label4: System.Windows.Forms.Label
- label5: System.Windows.Forms.Label
- label6: System.Windows.Forms.Label
- label7: System.Windows.Forms.Label
- label8: System.Windows.Forms.Label
- label9: System.Windows.Forms.Label
+ numericUpDown1: System.Windows.Forms.NumericUpDown
- panel1: System.Windows.Forms.Panel
- pictureBox2: System.Windows.Forms.PictureBox
- pictureBox3: System.Windows.Forms.PictureBox
- pictureBox4: System.Windows.Forms.PictureBox
- pictureBox5: System.Windows.Forms.PictureBox
+ richTextBox1: System.Windows.Forms.RichTextBox
- textBox1: System.Windows.Forms.TextBox
- timer1: System.Windows.Forms.Timer
- timer10: System.Windows.Forms.Timer
- timer2: System.Windows.Forms.Timer
- timer3: System.Windows.Forms.Timer
- timer4: System.Windows.Forms.Timer
- timer5: System.Windows.Forms.Timer
- timer6: System.Windows.Forms.Timer
- timer7: System.Windows.Forms.Timer
- timer8: System.Windows.Forms.Timer
- timer9: System.Windows.Forms.Timer
+ videoSourcePlayer1: AForge.Controls.VideoSourcePlayer
+ videoSourcePlayer3: AForge.Controls.VideoSourcePlayer

# Dispose(bool) : void
- InitializeComponent() : void

```

Figure 20: Questions form class diagram

There are two types of questions, the first ones is YES/NO questions, they represent a simple response a yes or no. the doctor will ask the patient a question and will select one from the program, if he can't hear the doctor, he will see it on the wall as it is writing in Kurdish language.

The second type of questions has four answers; each different form one another, some have pictures for more details and some are only text.

In figure 21, we will present the first type of the questions, which is two answers question.

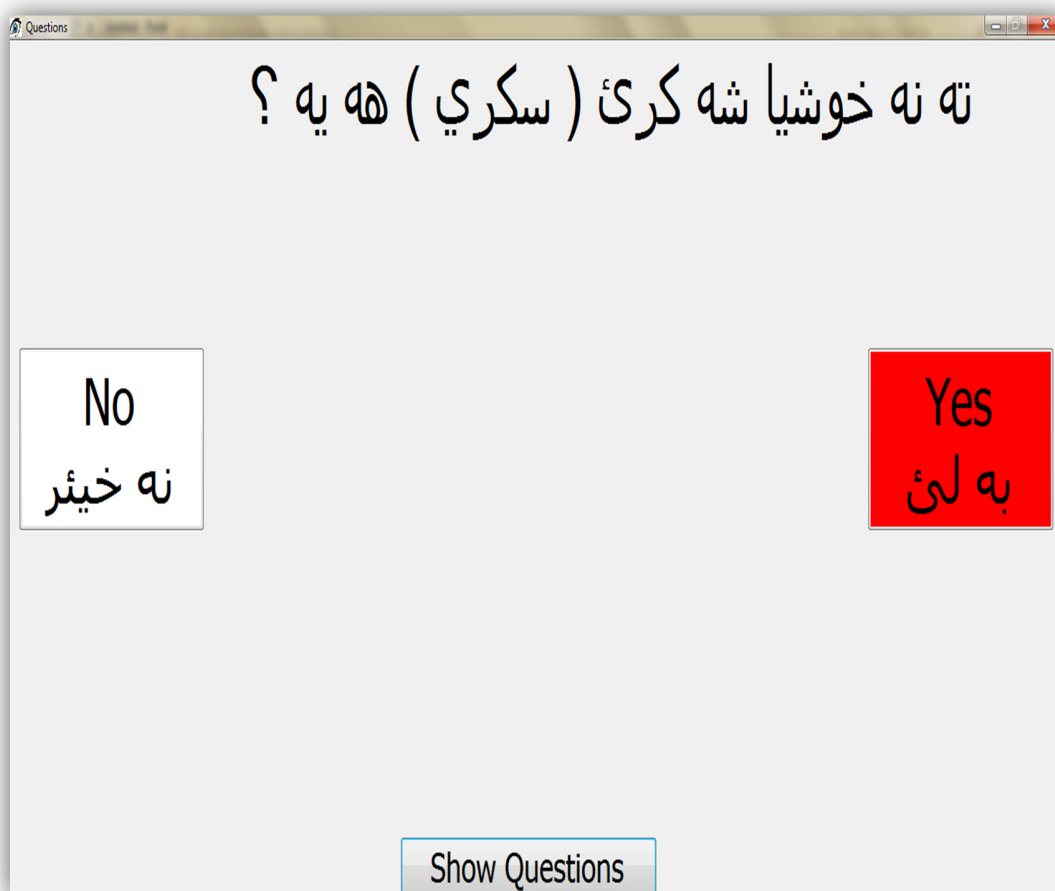


Figure 21: Two answers question

In figure 22, we present the second type of questions, which is four answers question.



Figure 22: Four answers question

The patient has four answer to choose from, according to his knowledge and to his ability, he will try to answer the correct one from them. The right answer highlights each time the patient looks at it.

When the patient selects one answer, every other answer borders turn into white color so that the doctor know what answer did the patient chose.

4. *HELP AND SLEEP*

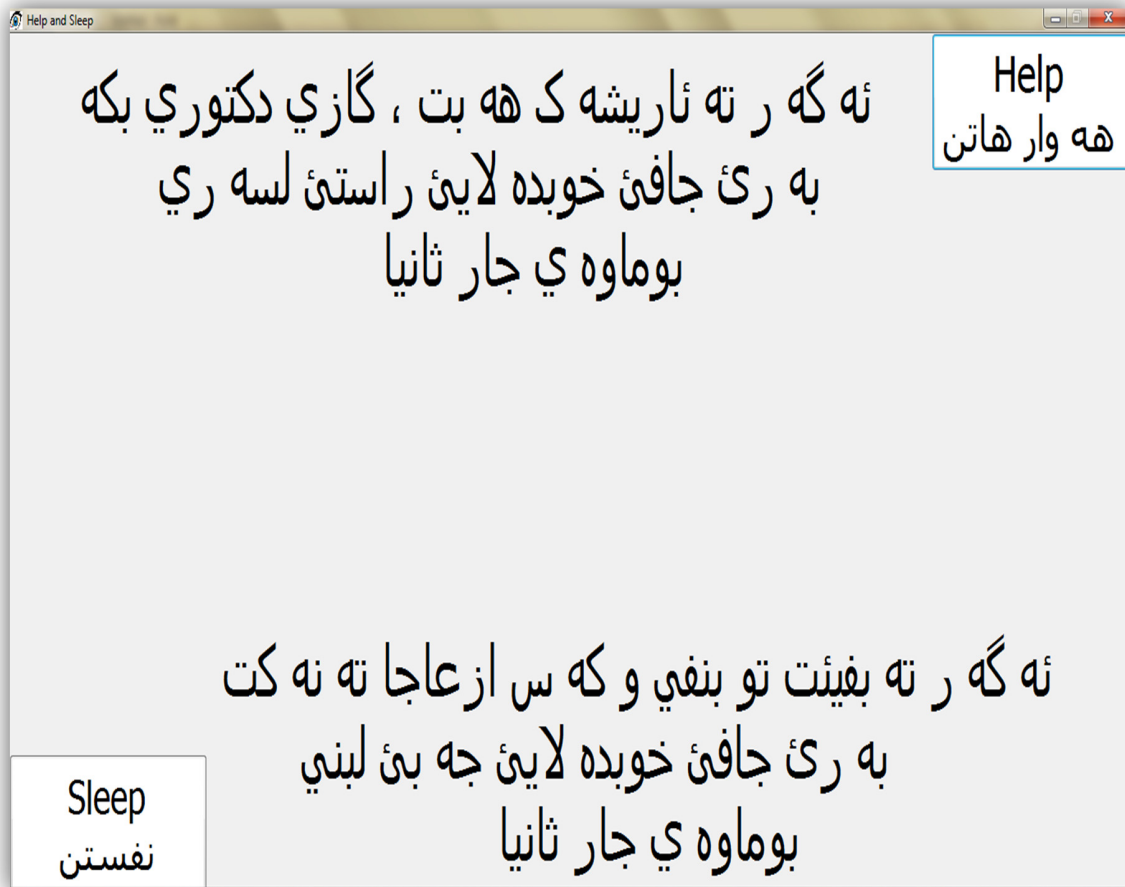


Figure 23: Help and Sleep form

The time that the doctor wants to leave the patient alone, and if he selects Help and Sleep form, figure 23 will appear. There are two options in this window, help option and sleep option.

class EyeTrack

HelpAndSleep

- button1: System.Windows.Forms.Button
- button2: System.Windows.Forms.Button
- + button3: System.Windows.Forms.Button
- button4: System.Windows.Forms.Button
- button5: System.Windows.Forms.Button
- + camerasCombo: System.Windows.Forms.ComboBox
- + colorDialog1: System.Windows.Forms.ColorDialog
- components: System.ComponentModel.IContainer = null
- groupBox1: System.Windows.Forms.GroupBox
- groupBox2: System.Windows.Forms.GroupBox
- groupBox3: System.Windows.Forms.GroupBox
- label1: System.Windows.Forms.Label
- label2: System.Windows.Forms.Label
- label3: System.Windows.Forms.Label
- + numericUpDown1: System.Windows.Forms.NumericUpDown
- + richTextBox1: System.Windows.Forms.RichTextBox
- timer1: System.Windows.Forms.Timer
- + videoSourcePlayer1: AForge.Controls.VideoSourcePlayer
- + videoSourcePlayer3: AForge.Controls.VideoSourcePlayer

- # Dispose(bool) : void
- InitializeComponent() : void

Figure 24: Help and Sleep form class diagram

The help button is used when the patient is having some health problems or he needs to call anyone for help, he will look at the top right of the wall and wait for 4 seconds then an alarm will start and an animated picture of an alarm device will appear.



Figure 25: Alarm Window

The other option that the patient can select is Sleep, the patient selects this option when he feels he wants to go to sleep and do not want anybody to bother him. He will look at bottom-left of the wall and wait for 4 seconds then an A window will appear and in it is a picture of a nurse that tells everybody to keep quiet and do not disturb the patient.



Figure 26: Sleep Window

5. ABOUT

The about form is used whenever the doctor needs to know how this program works, or has a question and wants to ask the programmer or administrator.

The first section is the instruction section; it explains the program in details.

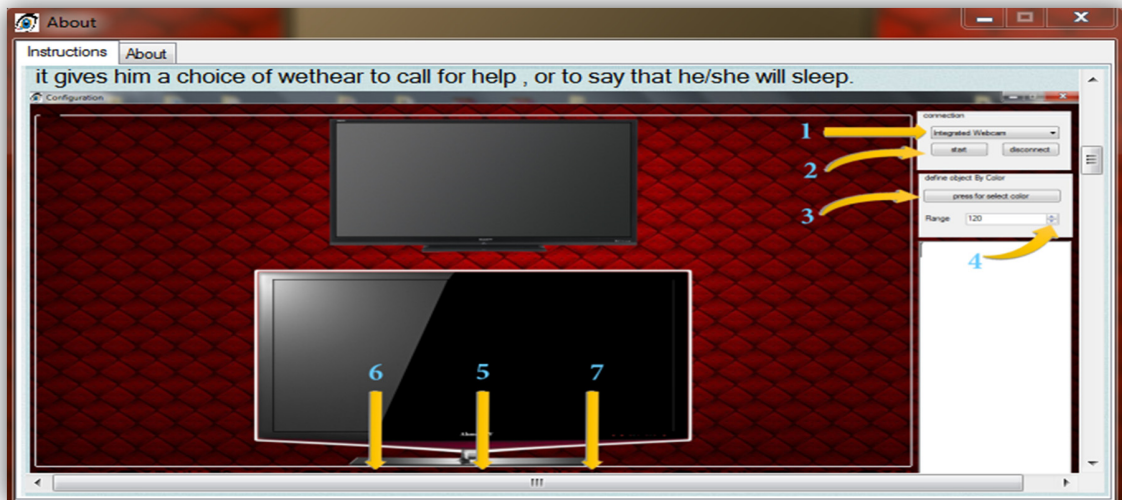


Figure 27: About form (instructions)

The second section contains informations of the developer of this program and is contained ways for communicating with him by using the project's website.



Figure 28: About form (about)

CHAPTER FIVE: RESULTS OF THE RESEARCH

1. THEORETICAL RESULTS

We made an eye-tracking device, and it worked. As discussed in the research, for some limitations, like lighting change it can effect on the efficiency of the software. We can use this as an equipment inside our hospitals to help patients with disabilities to interact with others.

2. EXPERIMENTAL RESULTS

All tests were performed on the Intel Core-i3, 2.53 GHz PC with a 15 inch LCD Monitor of resolution 1024x768. Bellow we present the tables of locations and the error percentage. The error is measured using four positions just as a four question in the software to test dots on the screen and comparing the positions received from the system to the dot coordinates. The results of the testing are given in this chapter, with a simple discussion.

Table 5: Top Left Location for points (X>20 & Y <90)

Dot X	Dot Y	Error in X	Error in Y	Error
42	77	0%	0%	0%
42	78	0%	0%	0%
43	78	0%	0%	0%
42	78	0%	0%	0%
44	77	0%	0%	0%
45	77	0%	0%	0%
44	77	0%	0%	0%
44	78	0%	0%	0%
43	77	0%	0%	0%
48	79	0%	0%	0%
44	80	0%	0%	0%
44	81	0%	0%	0%
44	81	0%	0%	0%
44	80	0%	0%	0%
44	80	0%	0%	0%
43	80	0%	0%	0%
41	79	0%	0%	0%
43	81	0%	0%	0%

43	81	0%	0%	0%
42	80	0%	0%	0%
43	80	0%	0%	0%
44	80	0%	0%	0%

In this table, it shows a great response between the locations of the detected object (eye) by color and the locations of the answers, for example the patient is looking at the top left of the wall, and the result is perfect, nothing goes wrong when looking at that direction.

Table 6: Bottom Left Location for points (X>20 & Y >90)

Dot X	Dot Y	Error in X	Error in Y	Error
167	186	0%	0%	0%
166	186	0%	0%	0%
164	187	0%	0%	0%
164	186	0%	0%	0%
164	186	0%	0%	0%
163	186	0%	0%	0%
166	185	0%	0%	0%
163	185	0%	0%	0%
163	185	0%	0%	0%
163	184	0%	0%	0%
164	183	0%	0%	0%
161	183	0%	0%	0%
163	183	0%	0%	0%
162	182	0%	0%	0%
162	182	0%	0%	0%
163	181	0%	0%	0%
175	181	0%	0%	0%
163	181	0%	0%	0%
163	181	0%	0%	0%
162	181	0%	0%	0%
163	181	0%	0%	0%
174	182	0%	0%	0%
163	181	0%	0%	0%
165	179	0%	0%	0%
165	178	0%	0%	0%
165	177	0%	0%	0%
164	177	0%	0%	0%
165	177	0%	0%	0%
177	176	0%	0%	0%
166	175	0%	0%	0%
176	175	0%	0%	0%
178	174	0%	0%	0%

Table 7: Top Right Location for points (X<20 & Y <90)

Dot X	Dot Y	Error in X	Error in Y	Error
0	60	0%	0%	0%
0	61	0%	0%	0%
0	60	0%	0%	0%
0	61	0%	0%	0%
0	62	0%	0%	0%
0	61	0%	0%	0%
0	63	0%	0%	0%
0	64	0%	0%	0%
0	67	0%	0%	0%
0	68	0%	0%	0%
0	68	0%	0%	0%
0	69	0%	0%	0%
0	71	0%	0%	0%
0	73	0%	0%	0%
0	73	0%	0%	0%
0	73	0%	0%	0%
0	74	0%	0%	0%
0	77	0%	0%	0%
0	78	0%	0%	0%
0	80	0%	0%	0%
0	86	0%	0%	0%
0	88	0%	0%	0%
0	95	0%	5%	5%

In this table, only one point were not the same as the desired location, but the total percentage is still great.

Next, we will show the last location, which is the bottom left, and see the results of it when looking at that direction.

Table 8: Top Right Location for points (X<20 & Y >90)

Dot X	Dot Y	Error in X	Error in Y	Error
0	122	0%	0%	0%
12	166	0%	0%	0%
30	186	10%	0%	10%
1	177	0%	0%	0%
0	176	0%	0%	0%
0	176	0%	0%	0%
23	177	3%	0%	3%
22	176	2%	0%	2%
25	177	5%	0%	5%
20	174	0%	0%	0%
42	187	22%	0%	22%

12	155	0%	0%	0%
62	176	42%	0%	42%
11	174	0%	0%	0%
42	189	22%	0%	22%
39	205	19%	0%	19%
4	147	0%	0%	0%
3	146	0%	0%	0%
3	147	0%	0%	0%
5	147	0%	0%	0%
0	148	0%	0%	0%
37	203	17%	0%	17%
42	190	22%	0%	22%
42	190	22%	0%	22%
37	188	17%	0%	17%
8	149	0%	0%	0%
38	204	18%	0%	18%
39	206	19%	0%	19%
16	178	0%	0%	0%
7	149	0%	0%	0%
6	148	0%	0%	0%
15	178	0%	0%	0%
14	176	0%	0%	0%
14	178	0%	0%	0%
15	178	0%	0%	0%
8	148	0%	0%	0%
40	203	20%	0%	20%
14	179	0%	0%	0%

What came out of bottom left location was a little disappointing because compared to the other locations, they never had an issue, in this location the average percentage of error is 7% this percent is more than the other three locations combined.

CHAPTER SIX: CONCLUSIONS AND FUTURE WORKS

This paper presents an eye-tracking system and a software program called Eye-Reaction. The AForge.NET framework used to develop the Eye-Reaction program, and it is a useful tool in software development since it is an open source framework with the available library files for image processing. However, the sequence of filters chosen using the AForge.NET framework proved to produce a processing delay since the image is being processed individually by the filters, thus, making the our system prone to errors. Hence, through this paper, the performance of a low-cost real-time eye tracking system has been tested using the AForge.NET framework in the field of image processing and gave use what we needed and answered all the questions that were asked in chapter 1. The aim of this project was to create an eye tracking system using a regular web camera and the AForge.NET library. As has been shown in chapters 3, the system has been created. In test results, the Eye-Reaction produced an average of 7% error percentage. This chapter will also discuss possible future works.

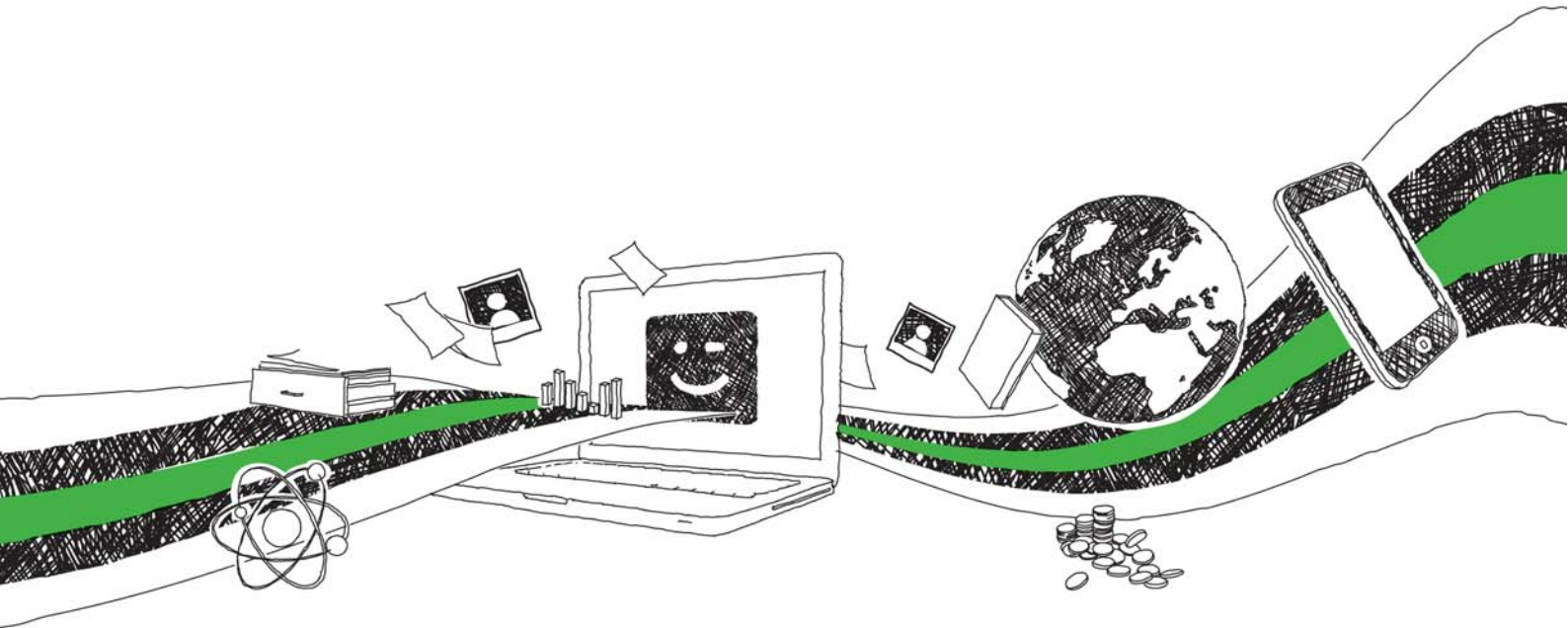
For future works, we attend to make other versions of the software in other languages, V1.2 in Arabic language and V1.3 in English language. Create a new software program for peoples who have solitude disease to make them learn things like numbers or letters or to help them improve their vocabulary and speaking. Use GSM with the Eye-Reaction device to send text messages to the patient's family whenever he feels uncomfortable or having some problems and no one is there. Use the eye tracking system in a mobile application program.

REFERENCES

- [1] M. B. e. al., The Camera Mouse: Visual tracking of body features to provide computer access for people with several disabilities., IEEE trans Nueral Systems and Rehabilitation Engineering, 2002.
- [2] J. S. M. W. B. a. B. M. Magee, "Eyekeys: a real-time vision interface based on gaze detection from a low-grade video camera," in *Real-Time Vision for Human-Computer Interaction*, Washington, DC., 2004.
- [3] P. P. a. Y.-F. H. Suraj Verma, "AN EYE-TRACKING BASED WIRELESS CONTROL SYSTEM," in *26th International Conference on CAD/CAM, Robotics and Factories of the Future 2011*, 2011.
- [4] N. C. I. L. J. a. F. I. Garay, "Assistive technology and effective mediation," *Interdisciplinary Journal on Humans in ICT Environments*, vol. 2, no. 1, 2006.
- [5] S. K. A. a. K. M. Azam, "Design and implementation of a human computer interface tracking system based on multiple eye features," *Journal of Theoretical and Applied*, vol. 9, no. 2, 2009.
- [6] P. K. S. K. P. Robert J.K. Jacob, "Commentary on Section 4. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises.," USA.
- [7] A. P. a. L. J. Ball, "Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects," UK.
- [8] R. S. S. M. I. a. H. Z. S. M. I. Xangdong Xie, "Real Time Eye Feature Tracking from a Video Image Sequence using Kalman Filter," *IEEE*, vol. 25, no. 12, 1995.
- [9] G. H. T. M. K. O. Takashi AOKI, "A Tracking Control to Multiple Objects for Plural Handd-Eye Systems," in *Advanced Motion Control, 2000. Proceedings. 6th International Workshop on 1-1 April 2000 conference*, 3-14-1 Hiyoshi Kouhokuku Yokohama JAPAN, 2000.
- [10] F. X. K. F. Xia Liu, "Real-Time Eye Detection and Tracking for Driver Observation Under Various," Columbus, Ohio , and 800 California St. Mountain View,, 17-21 June 2002.
- [11] T. Q. N. D.-U. B. Louisa Pui Sum Ipy, "FUNDUS BASED EYE TRACKER FOR OPTICAL COHERENCE TOMOGRAPHY," in *IEEE EMBS* , San Francisco, CA, USA , 2004.
- [12] J. W. a. L. Yin, "Detecting and Tracking Eyes Through Dynamic Terrain Feature Matching," Binghamton, NY, 13902, USA, 2005.
- [13] S. M. I. Zhiwei Zhu and Qiang Ji*, "Novel Eye Gaze Tracking Techniques Under Natural Head Movement," *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, vol. 54, no. 12, 2007.
- [14] J. Z. Jianxiong Tang, "Eye Tracking Based on Grey Prediction," First International Workshop on Education Technology and Computer Science, 2009.
- [15] Y. X. A. K. Li Li, "Robust Depth Camera Based Multi-User Eye Tracking for Autostereoscopic Displays," in *International Multi-Confrance on systems, Signals and Devices*, 67663 Kaiserslautern, Germany, 2012 - 9th.
- [16] F. U. V. S. Robert Gabriel Lupu, "Eye Tracking Mouse for Human Computer," in *The 4th IEEE International Conference on E-Health and Bioengineering - EHB 2013*, Iasi, Romania, 2013.
- [17] A. H. E. K. D. L. a. Z. J. Erik English1, "EyePhone: A Mobile EOG-based Human-Computer Interface for Assistive Healthcare," in *6th Annual International IEEE EMBS Conference on Neural Engineering*, San Diego, California, 2013.
- [18] Microsoft, [Online]. Available: <http://www.visualstudio.com/>. [Accessed 4 april 2014].
- [19] Microsoft, [Online]. Available: http://en.wikipedia.org/wiki/Microsoft_Visual_Studio. [Accessed 4 4 2014].

- [20] "Visual C# Language," Microsoft, [Online]. Available: [http://msdn.microsoft.com/en-us/library/aa287558\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/aa287558(v=vs.71).aspx). [Accessed 4 4 2014].
- [21] A. Kirillov, "Aforge.NET Framework," [Online]. Available: <http://www.aforgenet.com/framework/>. [Accessed 4 4 2014].
- [22] S. Shi, Emgu CV Essentials, Packt Publishing, November 2013.
- [23] A. Kirillov, "EuclideanColorFiltering Class," AForge.NET, [Online]. Available: <http://www.aforgenet.com/framework/docs/html/67fa83b5-dede-8d3a-8d3b-b7a6b9859538.htm>. [Accessed 30 april 2014].
- [24] E. Maessen, "Accurate Eye Tracking For Autostereoscopic Displays," Utrecht University, 2013.

YOUR KNOWLEDGE HAS VALUE



- We will publish your bachelor's and master's thesis, essays and papers
- Your own eBook and book - sold worldwide in all relevant shops
- Earn money with each sale

Upload your text at www.GRIN.com
and publish for free

