Eye Pupil Based Control of Electric Wheelchair

By

Muhammad Affan Zia

NUST201260401MSMME62112F

Supervisor

Dr Yasar Ayaz

Department of Robotics and Artificial Intelligence

A thesis submitted in partial fulfillment of the requirements for the degree of Masters in Robotics and Intelligent Machine Engineering (MS RIME)

In
School of Mechanical and Manufacturing Engineering,
National University of Science and Technology (NUST),
Islamabad, Pakistan

(March 2016)

Plagiarism Certificate

It is certified that MS Thesis Titled **Eye Pupil Based Control of Electric Wheelchair** by **Muhammad Affan Zia (NUST201260401MSMME62112F)** has been examined by us. We undertake the follows:

a. Thesis has significant new work/knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.
b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as Author own work.
c. There is no fabrication of data or results which have been compiled/analyzed.
d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.
e. The thesis has been checked using TURNITIN (copy of originality report attached) and found within limits as per HEC plagiarism Policy and instructions issued from time to time.

Thesis Supervisor

Dr Yasar Ayaz

Signature: ______________

Abstract

This project is aimed to developed a technique for such disabled people, who suffer with Amyotrophic Lateral Sclerosis (ALS) or quadriplegia and are not able to use a manual or even a motorized wheelchair. As ALS patients or quadriplegic patients are not able to move their hands, or use their hand muscles to perform a basic task, the aim is use such a body part which they are able to use and control properly. Some of these parts are head, tongue and eyes. In this thesis we developed a technique through which a patient will be able to control a wheelchair just by movement of his/her eye. A basic pupil detection technique is developed which is able to detect pupil in almost all kinds of lightning conditions using Infra-Red light. To use the light and dark pupil features of eye, a normal webcam was modified into an infrared camera. An eye gaze detection and estimation method was designed to find the gaze direction of patient. A stop/start and left/right protocol was tabulated and implemented for an easy control of wheelchair using the moving of eye.

Acknowledgements

First of all, I would like to pay gratitude to my thesis advisor Dr Yasar Ayaz for excellent guidance, strong support, expert advice and providing me the life time chance for doing research under his supervision even though I had no research experience. It was only due to his continuous motivation and guidance that I was able to complete this project.

I would also like to thank my co-advisor Dr Umar Ansari who was very insightful throughout the project and helped me step by step whenever any issue was faced. He has been very kind, soft of nature and always motivating.

I extend my gratitude to the rest of thesis committee Dr Muhammad Naveed, Dr Umar Ansari and Fahd Islam for taking out time off their busy schedule and providing me support, suggestions and helpful comments. Also I pay special thanks for Fahd Islam, who was always there to thoroughly discuss the ideas and refine techniques according to the cutting edge research.

I would also like to thank my friends and group fellows Usama Siraj and Ali Bin Wahid who were member of RISE Lab and were working on the smart wheelchair project, which was a part of their research work. Without their encouragement, motivation and support, this project would not have been possible. I am also thankful to other members of RISE Lab for their encouragement and maintaining a jolly and healthy research environment inside the lab.

In the end, last but not the least, I thank my father and mother for their prayers, support, encouragement and believe in me. My brothers and especially my sister were always there for brainstorming sessions and these sessions played a phenomenal role in making this project a success.

Table of Contents

List of Figures

List of Tables

Chapter 1

1 Introduction

ALS or Amyotrophic Lateral Sclerosis, is a progressive neurodegenerative disease that affects nerve cells in the brain and the spinal cord. As the disease evolves the ability of the brain to control muscle movement is lost. With time the patient may lose the ability to speak, eat, move and breathe. Within few years of the disease the ALS patient is likely to lose his ability to move freely. Eye muscles are one of the last muscles effected by this disease, which first occur after a long duration of the disease, often 20 years. This ability can thus be used to find a solution for such patients to move at their own will [1].

This project is a part of Intelligent Wheelchair Project carried out in the RISE Lab in the department of Robotics and Intelligent Machine Engineering, SMME, NUST.

1.1 Thesis Outline

This thesis comprises of five chapters including *introduction, literature review, eye detection and tracking algorithm, results and conclusion.*

- **Introduction:** This Chapter briefly discusses the motivation and need of this research topic. A brief introduction of the algorithm is described and our research goals are defined to set the expectations of the reader.
- **Literature Review:** A brief review of work carried out in the past on areas related to this topic is discussed in this chapter. Current research

work being carried out on this topic and some new state of the art ideas are discussed in this chapter.

- **Eye Detection and Tracking Algorithm:** In this chapter, our eye detection and tracking algorithm is discussed. This algorithm was divided into three parts, eye detection, calibration and tracking and creating ROS nodes for sending velocity commands to the wheelchair. The algorithm is built on top of OPENCV library for computer vision in C++.
- **Results:** In the fourth chapter the results of eye detection & tracking are discussed and a review of ease of driving the wheelchair with eye by a number of users is mentioned.
- **Conclusion:** In this chapter, the work carried out in this research project is summed up and the directions for future work in this area are laid down.

1.2 Research Goal

In this project we are aiming to develop a technique for disabled people, with which a person will be able to control an electric wheelchair with the help of his eyes only. A regular web camera modified into an infra-red range camera mounted on the head of the person captures the image of one eye. By simply looking in the direction in which the user wants to go, the controller connected to the wheelchair will take him in that direction. The user will be able to start or stop moving the wheelchair at his own will, and safety features, such as collision detection and avoidance, will be incorporated to avoid any kind of injury to the user.

1.3 Powered Wheelchair

For handicapped people, a wheelchair becomes a necessity and daily part of their lives. Before the advent of powered wheelchairs, commonly known as electric wheelchair, a wheelchair user was required to use raw muscle strength of upper limbs to move his/her wheelchair. For patients, that lacked the required upper body strength, another person was required to assist them in moving their wheelchair. In short, their handicap become a hindrance that limited their integration into the society. After the invention of powered wheelchair, it became possible for them to move without assistance, requiring no physical strength [2].

Most of the powered wheelchair in common use today, are battery powered devices, that use electric motors to drive the wheels of wheelchair. These powered wheelchairs are driven by a joystick, that is easily operable by a wheelchair user. With the advancements in control theory, robotics and computational capacity, it has now become possible to introduce new novel concepts and technology in assistive technologies, so that the effect of their handicap is mitigated to allow for their easier integration into society [2].

1.4 Wheeled Mobile Robot

A mobile robot is a wheel-driven or track driven automatic machine that is able to loco mate on-top of a terrain, which disregarding exceptions, is locally planar. These robots are battery driven, with electric motors driving the locomotion mechanics, though some outdoor robots are combustion engine driven. The steering and locomotion speed is controlled by drive circuitry present on the robot.

State of art mobile robots, also have a generic computation unit present, that has an Operating System (OS), which is mostly a variant of Linux, to provide higher level functionality. Generic computation, also called On- Board Computer or Single Board Computer (SBC), have network devices like Wi-Fi and Ethernet present, to provide connectivity with servers and remote terminals.

Mobile robots that requires extensive computation, have multiple On- Board computers present, in master-slave configuration, connected via Ether- net. These add-ons SBC perform computational intensive tasks like computer vision algorithms and Simultaneous Localization and Mapping (SLAM), which due to complexity are difficult to run on a single computer [2].

1.5 Sakura Wheelchair Project

Sakura Wheelchair project in RISE Lab SMME, is an initiative taken by SMME in collaboration with Sakura, Japan to carry out research on powered wheelchairs to facilitate amputated people, especially in Pakistan. Sakura in its joint venture with milestones, Pakistan donated two powered wheelchairs to RISE Lab, SMME.

Sakura in collaboration with a Pakistani Organization have launched a project, in which they provide used electric wheelchairs to such people in Pakistan who are disbaled and cannot afford the cost of these wheelchairs. Sakura is a well known company in Japan which manufactures such electric wheelchairs. The aim and vision of this company is to provide smart, active and disablity based maunal and electric wheelchairs to people with disablites, so that these people can be included into the societey [3].

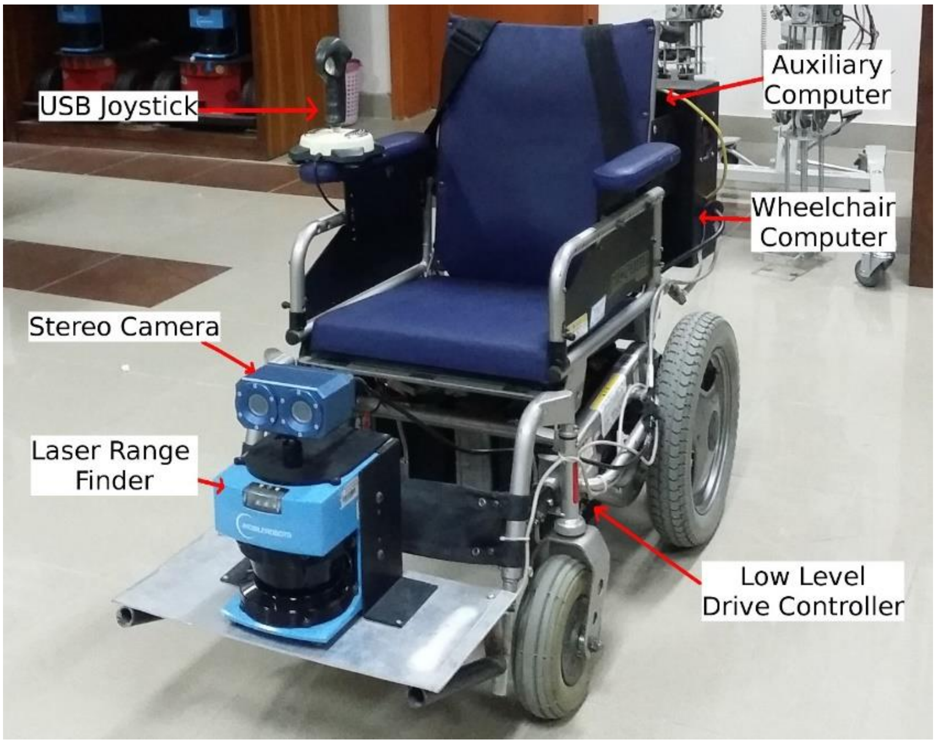

Figure 1: Motorized wheelchair transformed into a smart wheelchair (RISE Lab)

1.6 Computer Vision

Technical advancements in electronics have resulted in the cost of digital cameras being driven down. Lately, the computational hardware has also matured to the point, that practical applications of computer vision algorithms are now possible, with some constraints. Computer vision algorithms are mostly used in robotics for object detection and tracking, obstacle detection and avoidance, pose estimation and environment mapping.

Image sensing hardware usually involved are monocular or stereo camera, with pan-tilt units if gaze control is required. Most computer vision techniques are independent of electromagnetic spectrum used to capture the image. Visible spectrum cameras are mostly used in computer vision systems. For operation in poorly-lit environment, near infra-red (IR) cameras are mostly used for mobile robots. [2]

Chapter 2

2 Literature Review

Human Computer Interaction (HCI) is an emerging field that develops and improves techniques which allow a user friendly interaction between humans and computers. It has a large number of applications in the field of rehabilitation robotics, biomedical industry and in improving the lives of disabled and amputated people. According to reports 10% of individuals affected with tetraplegia (65 million) require a wheelchair for locomotion [4]. Conventional electric wheelchairs which comes under the category of Wheeled Mobile Robot were built for indoor and outdoor environments with maneuvering capability. These class of wheelchairs helped millions of disabled people in their everyday life routine tasks [5], [6]. However, for some people with motor disabilities even these simple tasks become difficult. So electric wheelchairs are one of the solution but they don't provide completed solution to all kinds of disabled people. For many people, an appropriate, well-designed and well-fitted wheelchair can be the first step towards inclusion and participation in society[5], [6].

Many techniques have been developed by many research while trying to improve the life of such patients. These techniques include, tongue [6], head [7]–[9], brain [10], [11] and eyes movement based approaches. The tongue and head movement based approaches for maneuvering wheelchair are highly accurate but the tongue movement based systems are very invasive and involve certain risks, such as infection. Moreover, invasive methods are hard to setup and not very user friendly. Some patients are unable to use head based technique because of their disability to move their head. This is common in spinal injury patients. In the brain control based technique, the patients could easily command their thought, but the accuracy is still very low and the setup protocol is inconvenient. Eye movement based approach is a technique which is usable for most patients and flexible to use [12]. In this project we chose eye base approach for developing a wheelchair for these disabled patients.

During the last few decades, lots of research has been done to extensively investigate Eye gaze in HCI [13]–[15]. Initially the method adopted was based on electrooculography, but these methods involved certain level of invasiveness. In this approach, an electric potential difference was measured through electrodes that were placed on skin. Gaze movements were determined through the acquitted signals, called the electrooculogram. In the method used by Duchowski, EOG signals were acquired through contact lens based system, but the limitation of this system was that it was quite expensive [16]. To explore further techniques for non-intrusive eye tracking methods, a mechanism utilizing infrared light and webcam was suggested, but the involuntary eye and head movement compromised the accuracy of this system [13], [14], [16]. Luis Figueiredo et al developed a similar interface but an expensive camera exhibiting higher frame rate was used. [17].

2.1 Different Eye Detection Techniques

Some of the most precise techniques for finding eye accurately are invasive or they involve expensive equipment. A method mentioned in [18] uses a contact lens which is fixed in place of eye. This design reduces the problem to track something attached to the lens. However, although this method is accurate, it is invasive and uncomfortable to use. This make it highly impractical solution to this problem.

Robust, nonintrusive eye detection and tracking technique is very crucial for Human Machine Interaction technology to be widely accepted for use in common environments, such as homes and offices. Existing eye detection and tracking techniques can be broadly classified into two categories: traditional image based passive approach and active IR based approach. Traditional techniques use unique features of eye such as intensity distribution, or the shape of eye. The basic assumption is that eyes appear different from rest of the features on face, both in shape and intensity. By exploiting these basic characteristics, most features base their research work for detection and tracking of eye. On the other hand, the active IR based approach uses a unique quality of eye pupil, that is its spectral or reflective property in the presence of near IR illumination. Traditional methods can be further classified into three categories: template based methods [19]–[27], appearance based methods [28]–[30] and feature based methods [31]–[38].

2.1.1 Template Based Methods

In **templated based methods**, based on shape of eye, a generic eye model is designed first. Template matching is then used to search for eye in an image. A template based method was proposed by Nixon [26] for accurate measurement of eye space using Hough transform. The iris of the eye was modeled using a circle and a "tailored" ellipse was used for the sclera boundary. However, their method was computationally costly, needed a high contrast image, and worked only with frontal face. Another approach commonly used is deformable templates [19]–[21]. First an eye model is designed which is allowed to translate, rotate, and deform to fit the best representation of the eye shape in the image. The eye position can be then obtained through a recursive process. Although this method is accurate, the eye model needs to properly initialized near the eyes. Moreover, this method is computationally expensive and requires high contrast images.

2.1.2 Appearance Based Methods

The appearance based methods detects eyes based on their photometric appearance. Such methods usually require to collect a large amount of training data. This data consists images, containing eyes of different subjects, under different lightning conditions and different face illuminations. A classifier method such as a neural network or support vector machine is used to train a classifier from this data. Detection is achieved via this classifier. Pentland et al. [28] further extended the eigenface technique by yielding eigeneyes, eigennoses and eigenmouths for the description and coding of facial features. A principal component projective space called the Eigeneyes was constructed by training from an appropriate eye templates, used for eye detection. By comparing a query image with an eye image in the eigeneyes space, eye detection can be accomplished. This method was also employed by Huang et al. [29] to perform initial eye detection. Wavelets were incorporated by Huang et al. [30] to represent eye image, and RBF NN classifiers were used to perform eye detection. Several improvements were proposed by Reinders et al. [36] on the neural network based eye detector. Scaled or rotated eyes under different lightning conditions can be detected by the trained neural network. But this is trained for the frontal face only.

2.1.3 Feature Based Methods

Feature based methods identify some distinctive features of the eye by exploring some basic characteristics such as edge and intensity if iris and color distribution of sclera and flesh. A feature based method was proposed by Kawato et al. [31] for detection and tracking of eyes. They proposed to detect a point between two eyes, instead of detecting the eye directly. According to these authors, this center point was more stable and easier to detect. Eyes were detected as two dark parts, located symmetrically on each side between eye point. A new eye model was designed by Feng et al. [24], [25] which consisted of six landmarks or eye corner points. Variance projection function (VPF) was used to locate the eye landmarks which were further employed to guide the eye detection. The problem in this method is that, detection will fail in case of a closed or partially occluded eye by hair or due to such a face orientation. Sometimes, eye brows and other facial hair are mistaken for eyes. Another method was proposed by Tian et al [35] to detect and track eye. However, this method requires manual initialization of eye model in the first frame. A modified version of Lucas-Kanade tracking algorithm [39] for tracking of inner corner of eyes and eyelids. The shape information of eye was extracted from edge and intensity values of eyes. However, for good detection a high contrast image was required. So in a nutshell, traditional image based eye tracking methods exploit the different shape and appearance features of eyes from the rest of face. However, these features are diminished due to some reasons, such as different orientation of face, changing light conditions, eye closure or occlusion, or variation of scale or location. That is why these kind of algorithms are not too good for real time tracking applications, such as for wheelchair.

Another approach that is often applied is the use of active IR illumination. There is a peculiar spectral behavior of eye in red light, infrared light, and near infrared illumination due to which this method is often used in such scenarios. Often the pupil of eye appears red when captured by images. This effect is commonly named as the red eye effect. Many techniques and even some commercial eye trackers [40], [41] are developed on the base of this principal. In the presence of IR light, a dark or bright pupil effect is produced. In the research of Ebisawa et al. [42] bright/dark pupil images were generated by using a differential lighting scheme. Two IR light sources were used, one was place on camera axis and other off camera axis. Subtracting the dark pupil image from the bright pupil image, gave the location of eye pupils. They improve this method later by eliminating the glass reflection using pupil brightness stabilization [43]. This approach was also utilized by Morimoto et al. [44] to generate bright and dark pupil images. These kinds of systems rely heavily on brightness and size of the pupils, which are affected due to several reasons, such as distance to camera, external lightning conditions, eye closure and eye occlusion etc.

All these IR based eye tracking methods require stable lightning conditions, closer view of the eye, open or un-occluded eye and stable lighting conditions to ensure bright pupils. These limitations reduce the application scope and flexibility. For a complete eye tracking algorithm, it is necessary that it must be able to accurately and robustly track eyes under these conditions. An image difference method was proposed by Ebisawa [43] which was based on two light sources. The advantage of two light sources was that it was able to perform pupil detection under various lighting conditions [45]. Background elimination was performed using image difference method and pupil was detected using the lowest possible threshold in the difference image. To eliminate glares on the glasses, they presented an ad hoc algorithm using thresholding and morphological operations. However, determining thresholding values and structuring element size of morphological operation, automatically, is difficult if not impossible. So this algorithm needed some tuning before starting to detect and track eyes accurately. Haro [46] proposed that if we combine eye appearance, bright pupil effect and motion characteristics of eye, pupils can be singled out easily from other bright objects in the image. A conventional appearance based matching method and the motion characteristics of eyes was proposed by them to verify the pupil blobs. But this algorithm failed in the scenario of closed or occluded eyes or eyes with weak pupil intensity because of external sources of illumination. In the method of Ji et al. [47], real time subtraction was used to detect possible pupil blobs and to eliminate spurious pupil blobs shape and size based pupil verification method was performed, but it was difficult to distinguish real pupil blobs from noise blobs on the basis of shape and size only [48].

2.2 Structure and working of human eye

Understanding the workings of the eye is vital to any sort of eye or gaze tracking system. In order to understand how to find the focus point of the gaze it is necessary to first understand the relevant biology of the eye and what eye behavior is relevant to finding the focus point.

It is very important to understand the structure and working of eye before proposing any kind of eye or gaze tracking system. A structure of eye is illustrated in Figure 2 below which shows the shape and important components of eye.

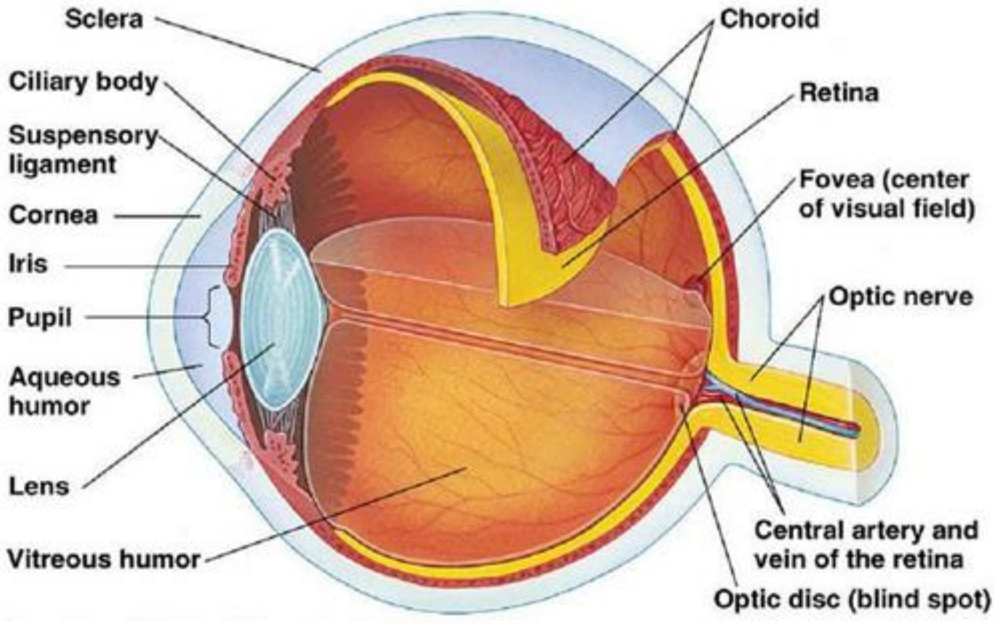

Figure 2: Internal Structure of Human Eye

When looking at the structure of eye, one of the important factors to consider is the location at which the receptors of eye are located. The eye receptors are spread in the form of high density areas and low density areas instead of a uniform spread which would have been a problem as stated by [49]. One such area is the fovea, which is a spot on the back of the eye as shown in Figure 2. According to [49, p. 3] , the fovea covers a one-degree width with the vertex of the eye, which means that as the eye moves away from the screen or point of gaze, the area uncertainty becomes larger. The vision of a person drops to half or less than that outside of the fovea. It was found in [49], that for reading text peripheral vision is not good enough. For our application we do not need the user read any kind of text, however for accurate estimate of gaze direction and proper control of wheelchair it is important that we determine eyes focus point and that is why fovea is important for our application. In [50] it is mentioned that a person moves his eyes in such a way that the focus area remains within fovea area, in order to focus on an object. However, a problem with is that, within fovea there is still one degree of error, which means in case of a very small focus area, the focus point can shift without any movement of eye. This is because focus area would still be inside the fovea area. This means a less than one degree of error will always be there when we measure fovea using the eye position. Also since the fovea is very small and within the eye socket, as can be seen in Figure 2, it is difficult to be used as a feature for eye tracking. So, the pupil or iris is tracked and fovea’s focus point is extracted from that. [49]–[51] described eye movements as sudden jumps and rests point in between. These jumps are called Saccades and the rests in between are call fixations.

2.2.1 Saccades

Saccades are movements of pupil or quick jumps of iris between the focus points. It was found in [52], a paper on Saccadic Suppression, that in order to maintain perceptual stability, trans saccadic spatial memory is sacrificed. This means that no visual information is gained during the saccadic motion, in order to maintain smooth and stable perception. This effect is called saccadic suppression [50]. So from this we understand that to gain information of focus points, we need to focus on the fixations of eye as they are more important for us than saccades. Three other eye movements were described by [53], referred to as Saccadic Eye Movements (SACs). According the research in [50], it is noted that we need to distinguish the quick movements of eyes from SACs. These movements are pursuit, vergence, and vestibular. Following a moving focus point is called pursuit, which involves performing saccades to catchup if the point is moving too rapidly, but normal eye movement is slower than saccades. The movement of eye, when the eyes move inward to focus on a nearby point, is called vergence. The rotations of eye to compensate for larger movements of head or body are called vestibular movements [50].

2.2.2 Fixations

The rests in between saccades, where the eye is focusing and gaining visual information, are called fixations. Jacob describes fixations as a period of relative stability during which an object can be viewed" [49, p. 5]. The reason for the term relatively in the above statement is because there are still tiny movements during the fixations. These movements are named smooth pursuit eye movements (SPEMs) in [49, p. 5], [53]. There are three different types of these small movements and they are usually within a one-degree radius [49, p. 5] which agrees with the previously mentioned fovea error. The first is a constant tremor of the eye called a nystagmus. The second movement is known as drift, which is where the eye occasionally drifts slightly off the focus point due to a less-than-perfect control of the oculomotor system by the nervous system [50, p. 374]. These drifts are then compensated for by micro-saccades which are an even more rapid eye movement then saccades, that brings the focus of the eye back to the object [50].

As mentioned above we need only look at the fixations to and the focus of the eye. This is good since the fixations last longer than the saccades [50].

CHAPTER 3

3 Implementation

This project was mainly split up into two parts: *Hardware Design, Software Implementation.*

- **Hardware Design:** Hardware design part consisted of two subparts
    - Image Acquisition Module**:**
      It consisted of designing and manufacturing an image acquisition module to acquire a clear image of eye.
    - Wheelchair Hardware Module**:**
      Conversion of a regular powered wheelchair into a smart wheelchair which included installing a controlling and communication module and sensors to get the data of the wheelchair surroundings. This was done by replacing the traditional joy stick with a computer module, and a LASER Range Finder and a Stereo camera was installed. The Stereo camera is not used in this part of the wheelchair project. The wheelchair mounted with all its sensors is shown in Figure 1
- **Software Implementation**
    - Accurate Eye Detection and Tracking in Real Time
    - ROS based driving and control of Electric wheelchair, with data receiving from sensors.
    - ROS node for sending eye tracking data to wheelchair.

3.1 Image Acquisition Module:

Traditional, non-contact Eye tracking systems had several problems that need to be taken into consideration when calculating the focus point or when finding the eye. The first and foremost of these is head movement, as mentioned by [54]. Older eye tracking systems required the user to keep their head still in order to accurately capture the data. More comprehensive systems, like [55], [56] and [34] take this problem into account in their algorithms. These algorithms all tend to incorporate infrared lighting.

Figure 3: Image acquisition module

To solve these issues, we designed our eye tracker to be mounted on the user's head with a very small camera in front of his one eye located in such a way that it does not disturb the view of user. The camera was positioned in front of right eye as shown in Figure 3. This design gave us two advantages:

- First, the problem of head movement or head tracking was not faced and obstruction of eye in the image was avoided.
- Second advantage was that noise was automatically reduced as the image contained scene of only some part of face, eye brows and eye. Any problem arising due to background was solved automatically.

However, the problem of background noise was solved algorithmically. Another advantage of this design was a closed up and clear view of eye. As this eye

tracker was mainly designed for driving a wheelchair, so the clear view of eye was most important, as little obstruction can be proved harmful for the user.
The camera that we used was a low cost 5 mega pixel Dany PC 909 web camera. The camera was modified by removing the IR filter and replacing it with a photographic film which worked as an ambient light filter. This converted camera to a near infrared region camera and the image obtained was greyscale. Presence of infrared source was important to acquire proper image. 4 IR LEDs were installed on four sides of the camera. The IR beam produced by these LEDs was reduced to a very low intensity, into a safe range to avoid any harmful effects.

3.2 Wheelchair Hardware Module:

Our algorithm was implemented on an electric wheelchair (Suzuki MC 2000). This wheelchair comes with a joystick as an input method for navigation which is replaced by a computer (Cobra EBX-12) [57]. The motor drive of the wheel chair was connected to our ROS based computer to control its motion. This ROS based system gave us the flexibility to control the wheelchair via different HCI (Human Computer Interaction) input devices. In this project a head based eye tracker is used as the input device. The overall hardware design of the wheelchair is shown in Figure 4.

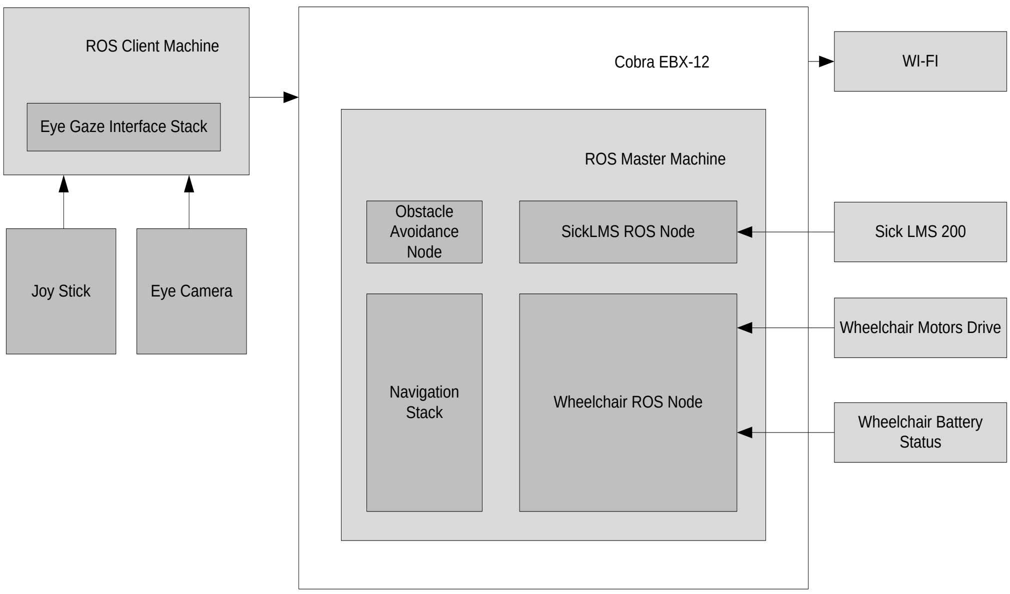

Figure 4 : Complete System Design of eye controlled wheelchair.

The wheelchair consisted of two electric motors, connected with the rear wheels, which were connected to a main controller powered by batteries. The main controller or the wheelchair motors drive was used to translate the command signal given by the master computer and control the speed and direction of motors. The main controller of the wheelchair was connected to a master computer which is a Cobra EBX-12 onboard computer and has a 1.8 GHz Intel Pentium M processor and 512MB of RAM. A Wi-Fi module was mounted on master computer for communication with other systems. A LASER Range Finder (SICK LMS 200) connected to the master computer via serial port was mounted on the front end of wheelchair.

The command protocol between the main motor controller and wheelchair computer was reverse engineered by logging the data from wheelchair joystick into a computer.

The on board computer runs on Linux (Ubuntu 12.04) operating system and Robot Operating System (ROS) was installed on it. This computer was set as the master ROS server. Eye tracking module communicated with ROS master via Wi-Fi and sent input in the form of command velocity (cmd Vel). This command velocity was then translated by ROS master to a form that is understood by the main controller of wheelchair which in turn converts it to the speed and direction of motors.

3.3 Implementation of ROS on wheelchair:

Robot Operating System or its short form ROS, as normally used, is not actually an Operating System (OS) in a traditional kind of way, as an OS is responsible for management and scheduling of processes. Rather, ROS is actually a structured framework which provides a layer of a communication layer, which operates upon the host operating system of a heterogeneous computing cluster. The purpose of designing this framework was to encounter certain challenges faced when researchers were developing large scale service robots. This research was part of the STAIR project at Stanford University and Personal Robot Program at willow garage. However, the design of framework that was formed at the end of the project was very general than the service robot and mobile-manipulation domains [58].

3.4 Structure of ROS:

The fundamental building blocks of ROS implementation are explained below.

- **ROS Core**

  This is the main program that manages information of all the running nodes and message topics, and binds them together at runtime. ROS core maintains a table of running nodes, storing their machine host- names, list of their subscribed and published topic names and services. All nodes communicate with the running single instance of core to retrieve information of topics and services when subscribing and publishing new topics. This communication takes place using XML-RPC.

- **ROS Nodes**

  All programs running in ROS framework are called nodes. Nodes perform different tasks and have different behaviors which are used by other nodes in the ROS run time environment. The nodes provide functionalities to the robot that are required by the robot. Some nodes are sensor hardware specific, like drivers for a particular camera or a laser range finder, while there are other nodes which are independent of a specific hardware, and work with different hardware of same type. Once such example is the node for GMapping. API bindings in several languages is possible with ROS, such as C++, Python and LISP which gives user the flexibility to write a ROS node in a programming language of his/her choice.

- **ROS Messages**

  ROS Messages are used by nodes to share information between each other. A message uses standard types of data structure or data structure developed by the user. A message can consist of other messages, or an array of messages. A command line tool rosmsg can be used to get information about the message [58].

- **ROS Topics**

  Topics are mediums used by nodes to transmit data. A topic can have various subscribers that’s why data can be transmitted between nodes without a direction connection. These messages can be simple such as single number or text string, or they can be a complex data, such as map of environment. This communication is unidirectional. These topics are used to correctly bind messages from publishers and subscribers, so that the message reaches its intended nodes. The binding happens only once at time of registering, thus reducing communication overhead [58].

- **ROS Services**

  Since topics are unidirectional, if a need arises to reply to a message, it can’t be done using topics. This can be done using service. Services are developed by the users and standard services do not exist for nodes [58].

- **ROS Packages**

  The Nodes and messages are combined together to form a package, for easy distribution. The purpose of these packages is to benefit ROS community by sharing these packages with different research groups which gives it the ability to be easily reused. New algorithms for different robot related tasks are shared between researchers through these packages, which gives these packages and ROS collectively a continuously improving capability [58].

- **Parameter Server**

  Though logically separate, ROS core provides a parameter server functionality, that is visible to all nodes. Parameter server stores a lookup table of variables that is used by running nodes to retrieve optional parameters. These optional parameters are node specific, and provides a platform independent mechanism to alter the behavior running nodes [58].

3.5 Feature Based Pupil Detection Algorithm:

A feature based algorithm was developed in this thesis to detect the pupil of eye. The main reason for selection of this detection method was the observation that when an eye image is captured by a near infrared camera, in the presence of IR light we get very strong eye features.

Infrared illumination produces a bright pupil effect if the source is located very close to the axis of the camera. As a result, the pupil is very clearly demarcated as a bright region [59]. As we can see in Figure 5 below, different eye regions can be easily identified in the image. This image was taken by regular webcam, which was modified into a near IR region camera used in this experiment. An algorithm for the pupil localization and gaze tracking and wheelchair navigation was developed in this research. The first step of the algorithm was image acquisition of a single eye using a head mounted near infrared camera. The image obtained with the IR camera clearly distinguishes between the bright sclera, gray iris and the dark pupil as it can be seen in the Figure 5.

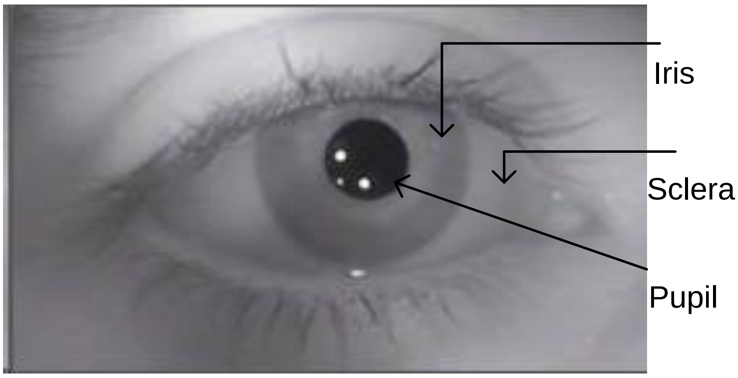

Figure 5 : Image of eye captured by a near Infrared camera in the presence of IR light source at RISE Lab.

A complete flow diagram of whole process can be seen in Figure 6. After obtaining the image, a method developed by viola et al. [60] known as Haar Classifiers was used to rapidly detect and extract eye from the image. This algorithm uses AdaBoost classifier cascades which are based on Haar-like features and not pixels.

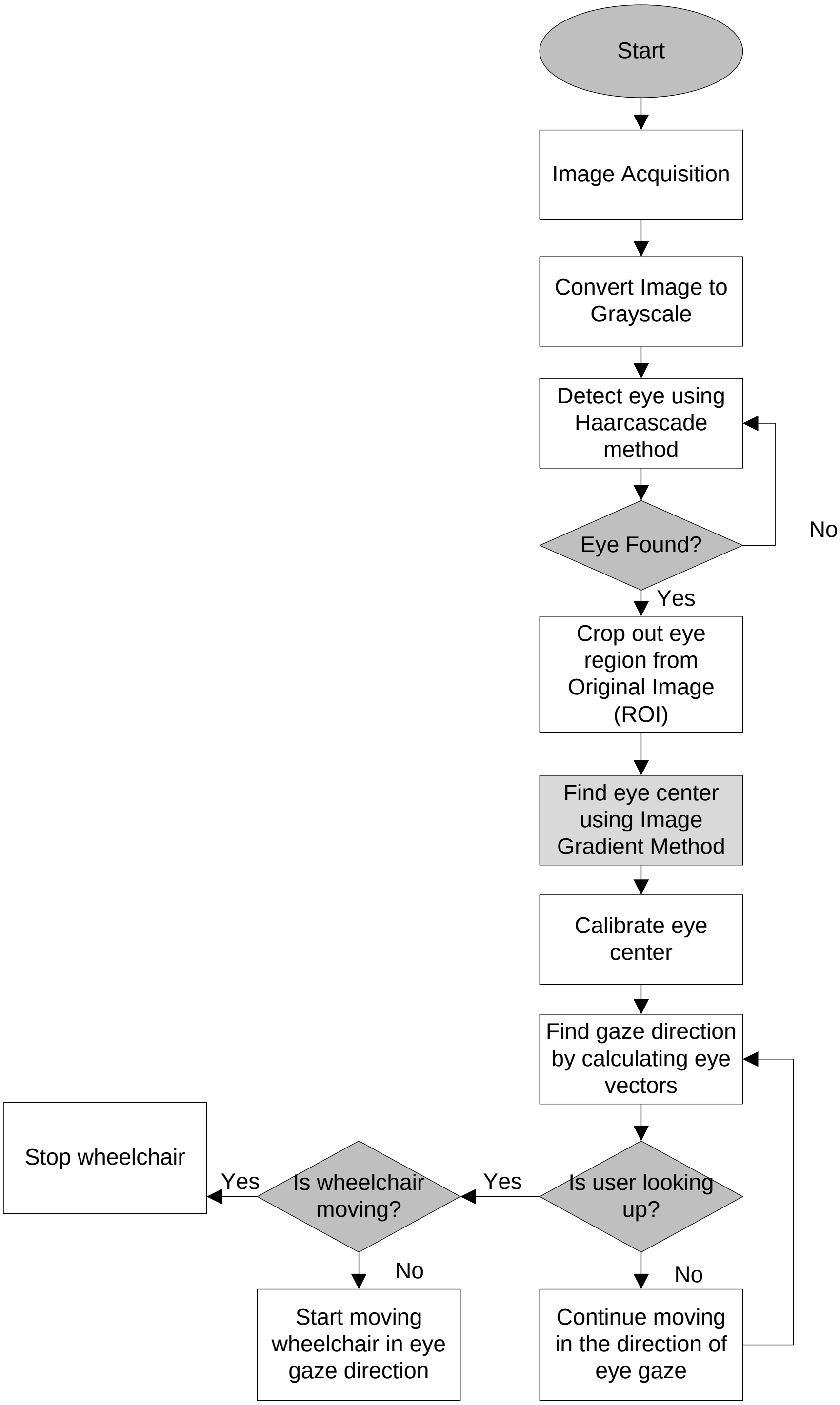

Figure 6: Flow diagram of complete process.

3.5.1 Extraction of Eye:

3.5.1.1 Haar Cascade Classifiers:

Haar-like features are the core basis for Haar classifier object detection. Instead of using the intensity values of a pixel, these features use the change in contrast values between adjacent rectangular group of pixels. To determine relative light and dark areas, the contrast variances between the pixel groups are used. Two or three adjacent groups with a relative contrast form a Haar-like feature. Haar like features used to detect an image are shown in Figure 7. These features can be easily scaled by increasing or decreasing the size of pixel group being examined. This allows these features to be scalable and so they can be used to detect object of variable sizes [61].

3.5.1.2 Integral Image:

An intermediate representation of an image, called the integral image [60],is used to calculate simple rectangular features of an image. The integral image is an array, which contains the sums of the pixels' intensity values located directly to the left of a pixel and directly above the pixel at location (x, y) inclusive. So an integral image can be represented using equation 1 where BI[p, q] is the integral image and B[p ,q] is original image. This is also illustrated in Figure 7.

$$BI[x,y] = \sum_{p' \le p, q' \le q} B(x',y') \qquad 1$$

Lienhart and Maydt introduced another feature, which were rotated by 45 degrees as shown on left side in Figure 7. These features require another intermediate representation called the rotated integral image or rotated sum auxiliary image [62]. The rotated integral image is calculated by finding the sum of the pixels' intensity values that are located at a forty-five-degree angle to the left and above for the p value and below for the q value. So if B [p, q] is the original image and BX [p, q] is the rotated integral image then the integral image is computed as shown in equation 2 and illustrated in Figure 7.

$$BX[p,q] = \sum_{p' \le p, p' \le p - |q - q'|} B(p',q') \qquad 2$$

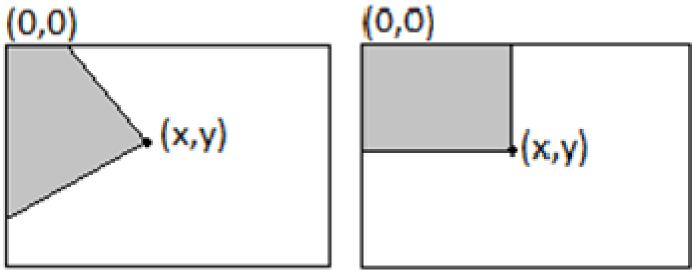

Figure 7: Haar Features

In order to determine both integral images arrays, two steps are required, one step per each array. A feature of any scale can be extracted by using suitable integral image, and taking the difference between six to eight arrays which form two or three connected rectangles. As a result, extracting a feature is very fast and efficient [60]. Similarly extracting other features of different sizes, require the same amount of computational effort as a feature of only two or three pixels. Since scaling require no additional effort, different sizes of same object are detected in same amount of time and take same computation power [60].

3.5.2 Classifier Cascaded:

Feature extraction is a very fast and efficient process, but calculating all 180,000 features contained with a 24 x 24 sub image is not practical [60], [61]. However, luckily a very small portion of those features are required to determine that whether a sub image potentially contains the desired object [63]. To reduce the calculation time, and remove as many sub images as possible, very less number of such features that define the properties of an object are used while analyzing the sub image. The target is to remove a significant amount, say 50% of the sub images which do not contain the desired object. As this process continues at each stage, the number of features used to analyze sub images are increased. This process of cascading the classifiers allow only those sub images, which have the highest possibility to be analyzed for all the Haar-features that characterize an object. Due to this reason, the accuracy of a classifier can vary. By decreasing or increasing the number of stages, both the false alarm rate, and positive hit rate can be altered. In the research of Viola and Jones, they were able to achieve an accuracy rate of 95 % for detection of human face, using only 200 simple features [60]. The computation speed of a Haar-cascade classifier, in their case was at a rate of five frames per second using a 2 GHz computer [62].

3.5.3 Training Classifiers

There can be many approaches of machine learning techniques used to learn a classification function for a given feature set and a training set of positive and negative images. A variant of AdaBoost algorithm was used in this technique for the selection of a small feature set and training the classifier [64]. Originally, AdaBoost algorithm was used to increase the classification performance of a weak learning algorithm. This learning method, provides several formal guarantees. In the research of Freund and Schapire, it was proved that as the number of learning steps increases the training error of a strong classifier exponentially approaches to zero. Significantly a number of results were later proved regarding the generalization performance [65]. A very important point in this regard is that the generalization performance is related to the margin of the examples, and that AdaBoost achieves large margins quickly. This must be noted here that there are over 180, 000 rectangle features associated with the sub-window of each image, which is a larger number than the number of pixels in the sub image. Though the feature extraction process is computationally very fast, but computing the complete set of features makes it excessively expensive. Viola and Jones hypothesized that a very small number of such features can be combined to form an effective classifier [60]. The main challenge is then finding these features. In order to achieve this goal, the weak learning classifier is designed in such way that it selects the single rectangular feature which separates the positive and negative images in the best possible way (the approach used in [66] is similar to this approach for the retrieval of image database). For each feature, the weak learner determines the optimal threshold classification function, such that the minimum number of examples are misclassified. A weak classifier hj (z) thus consists of a feature fj, a threshold 0, and a polarity pj indicating the direction of the inequality sign:

$$h_j(x) = \begin{cases} 1 & if\ p_j f_j(x) < p_j \theta_j \\ 0 & if\ otherwise \end{cases} \qquad 3$$

In this equation x is a sub window of size 24 x 24 pixels of an image. Practically no single feature can perform this task of classification is low error. Features which are selected in early rounds of the boosting process had error rates between 0.1 and 0.3. Features selected in later rounds, as the task becomes more difficult, yield error rates between 0.4 and 0.5.

A requirement to create classifiers for specific objects is to train the classifiers using a set of positive samples containing the image to be detected and a negative set not containing the image. Once trained, the classifiers can be used to detect the objects [67].

These trained classifiers were used to detect eye region inside the obtained image which was further used as region of interest to detect the pupil of eye. Advantage of this was that by limiting the pupil detection process to the eye region only decreased the computational cost by many times and chances of errors were lessened as there were many pupil candidates outside the eye region such as eye brows, other facial hair and any other noise in the image.

3.6 Pupil Detection:

A complete process diagram of detecting pupil after extraction is illustrated in Figure 8. This eye region was then extracted from the video stream using cascade classifier detection methods and a method proposed by Timm and Barth [17] was applied to detect pupil inside the eye. Timm and Barth's method [17]of pupil detection works by geometrically finding the center of a circle by analyzing vector field of the gradient image. This method is an inspiration of a technique proposed by Kothari and Mitchel [68]who proposed that if we draw lines in the orientation of gradient vectors across an image then the point at which most of the lines intersect will be the geometric center of a circle in that image.

The most significant feature of the eyes, in a grayscale image, are the iris' ellipsoidal shape and the stark contrast between the iris and the surrounding area-the sclera [68]. A dark circular area, like the iris, on a light background, causes an outwardly radiating flow field to appear in a gradient direction plot of the image. Since the flow field around an iris radiates outwards it must intersect at a point if extrapolated in a direction opposite to the gradient (the gradient points in the direction of increase of a function, hence points outwards - from the darker iris to the lighter sclera) [68].

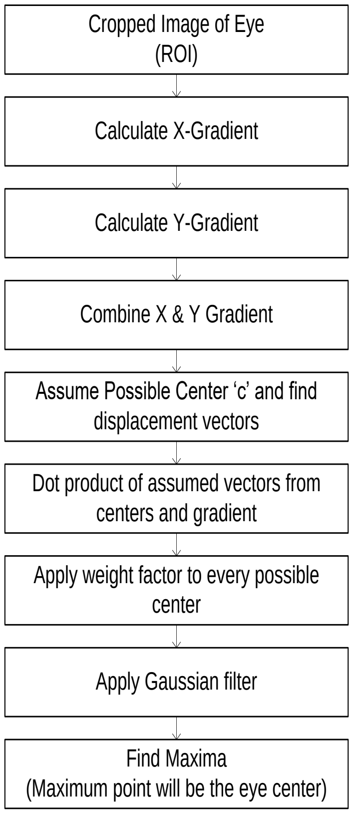

Figure 8 : Pupil localization using gradient method by Timm and Barth.

Their technique provided reasonable results, however a mathematical proof was missing. Timm and Barth proposed the mathematical formulation of gradient vector fields to relate the possible eye center with gradients orientation in the image. If **c** is the possible center and $\mathbf{g_i}$ is the gradient vector at position $\mathbf{x_i}$, then the normalized displacement vector $\mathbf{d_i}$ from a possible center **c** to point $\mathbf{x_i}$ can be found using relationship 4. If we use the vector field of (image) gradients, we can exploit his vector field by computing the dot products between the normalized displacement vectors (related to a fixed center) and the gradient vectors gi [17].

$$\boldsymbol{d}_i = \frac{x_i - c}{||x_i - c||} \qquad 4$$

$$c^* = \arg max_c \frac{1}{N} \sum_{i=1}^{N} w_c (\boldsymbol{d}_i^T \boldsymbol{g}_i)^2 \qquad 5$$

All the displacement vectors were scaled to unit length to obtain an equal weight for all pixels. An objective function was defined to find the optimal center of the eye by finding the dot product between all the possible centers and the gradient vectors. To increase the robustness and minimize the effect to due to change in intensity of light and other noise following post processing techniques were applied.

To decrease the possibility of wrong center estimations due to dominant eyelids and eye lashes or wrinkles, a weight factor was applied to each possible eye center so that the possibility of darkest center increases, as in IR images that we acquired; pupil was mostly the darkest point. To remove the effect due to bright outliers such as reflections of glasses, a Gaussian filter was applied. In the end a threshold, based on the maximum value, was applied on the objective function and all the values that were connected to the borders were removed for proper center estimation [17].

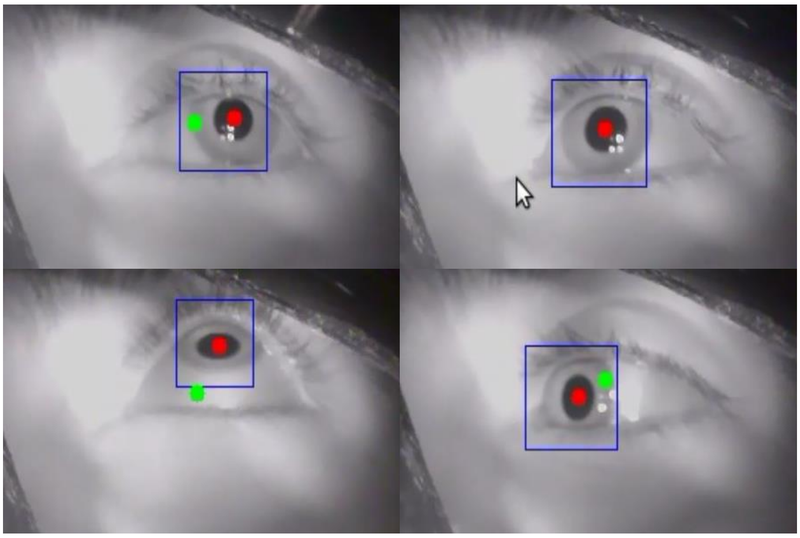

Figure 9: Result of algorithm after detection of pupil. The image shows up, down, left and right positions of the eye and a mean position is also shown.

3.7 Pupil Tracking and Gaze Estimation:

The next step in our algorithm is calibration and tracking of pupil. Calibration is done by recording the mean position of the pupil by taking input from the user while he is looking straight ahead. After calibration, the recorded point which is the mean center point is saved and used as a reference for future. This reference center point is the mean of eye center of five previous frames in order to avoid any kind of error. An eye vector is continuously drawn between the mean center point of the eye and the current pupil location using equations 6 and 7. Our novel eye tracking method computes the orientation and direction of eye vector by using following equations of geometry where x_c and y_c are the location of possible eye pupil and x and y are locations of other image pixels.

$$Vector_{length} = \sqrt[2]{(x_c - x)^2 + (y_c - y)^2} \quad 6$$

$$Vector_{angle} = tan^{-1}\left(\frac{y_c - y}{x_c - x}\right) \quad 7$$

After recording data of several test subjects we defined a rule table to detect the gaze direction and intent of movement of a user.

Table 1 shows the defined rules and graphical illustration of up, down, left, right and straight regions is also shown in Figure 10. On the basis of this rule table we define the method of controlling the wheelchair. A Start/Stop protocol is incorporated by using the looking above gesture of the user as a switch. After calibration to start moving the user is required to look in the up direction for few milliseconds and if user wants to stop a moving wheelchair he can do it by looking in the up direction.

Length of vector	**Angle of vector (Degrees)**	**Direction of Gaze**	**Notation**
Less than 20 pixels	135 to 235	Straight	A
Less than 20 pixels	285 to 75	Straight	B
Less than 8 pixels	75 to 135	Straight	C
Less than 8 pixels	235 to 285	Straight	D
Greater than 8	75 to 135	Up	E
Greater than 8	235 to 285	Down	F
Greater than 20	135 to 235	Right	G
Greater than 20	75 to 135	Left	H

Table 1: Rule table to define direction of motion of wheelchair with respect to eye.

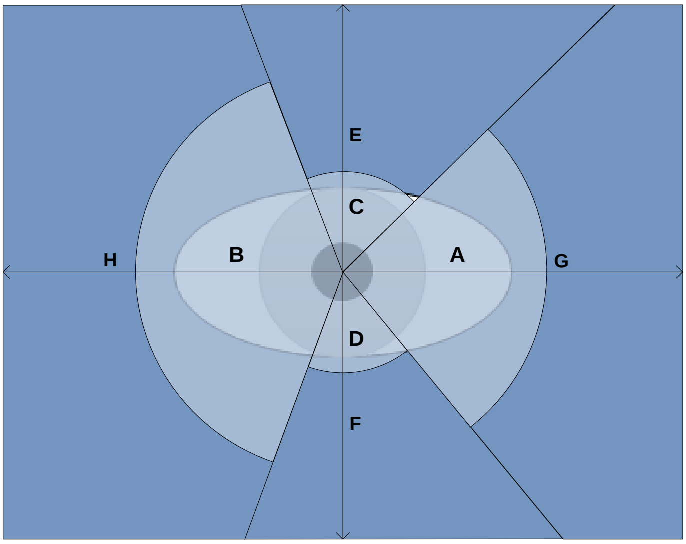

Figure 10: Graphical illustration of Rule table.

3.8 ROS Node for connecting Algorithm with wheelchair

Processes that communicated with other processes of ROS through topics and messages are called nodes. To link our eye tracking algorithm and sending commands to the wheelchair module a node was written. This node was responsible for translating the current gaze position of the eye of the user into speed and directions, and send this translated signals to the wheelchair. The wheelchair in response would act accordingly and move in the specified direction with the specified speed.

CHAPTER 4

4 Results & Discussion

In this chapter various tests conducted to verify the accuracy and precision of the pupil detection algorithm is discussed. Comparison with other methods by implementing on a standard data base of images containing faces with different orientations and lightning conditions is discussed and tabulated. Also the results of gaze tracking and driving the wheelchair by various test subjects is discussed.

4.1 Accuracy of Pupil Detection

The pupil detection algorithm was tested on the famous BioID database of facial images. This algorithm was then compared with the results of state of the art eye detection algorithms which were applied on the same set of images. The BioID database consist of 1521 grey level images of 23 different subjects. These images are taken in a variety of backgrounds and lighting conditions. Variation of poses, and images from different angles of faces are also included in this database. The algorithm was applied on this data base in multistage. The first step was detection of face in each image. After detecting the face, we concentrated on the left and right eye of the face only by cropping out this region geometrically. The reason for going through this exercise was because the algorithm that we designed was for a single eye only, as the camera was located in front of the face, directly taking image of a single eye. So to apply the algorithm on a single eye, we had to narrow down our region of interest.

4.1.1 Detection Error

A formula is tabulated and often used by most researchers to best describe the efficiency of their algorithm. This formula can be called detection error.

$$e = \frac{max\{\|L - \acute{L}\|, \|R - \acute{R}\|\}}{L - R} \qquad 8$$

Where L and R are the actual positions of left and right pupil respectively while L' and R' are positions calculated by the applied algorithm. For a given detection error d_{max} the efficiency of the pupil detection algorithm is defined as the number of images for which the method calculates pupil in range of d < dmax. The obtained results can be tabulated using the percentages at different values of d. This table can be seen bellow.

d_{max}	Our Method	(Asadifard and Shanbezadeh, 2010)	(T¨urkan et al., 2007)	(Valenti and Gevers, 2008)
0.05	82.1 %	47.0	18.6	84.1
0.10	92.0 %	86.0	73.7	90.9
0.15	94.2 %	89.0	94.2	93.8
0.20	96.3 %	93.0	98.7	97.0
0.25	97.0 %	96.0	99.6	98.5

Table 2: Comparison of Results with other state of the art methods

4.1.2 Accuracy of Eye tracking algorithm.

The eye tracking algorithm was tested on 50 test subjects in a controlled environment. Two experiments were designed to test the accuracy of eye tracking and its implementation on wheelchair. In the first experiment, a person following approach was used. A complex environment was created within the lab by using several obstacles, and a hallway and person following approach was used. Each test subject had to start the wheelchair, start following the person walking in the area through the obstacles taking turns and reach the goal point. Upon reaching the goal point the user had to stop the wheelchair with the stop protocol of eye. All of the test subjects were able to reach the goal point in the person following approach. Out of 50 test subject, 28 were able to reach the goal point without any external assistance. 10 test subjects were able to reach the goal point in the second try. For 12 persons, we had to retune our parameters of pupil detection, and for some the gaze tracking parameters were also tuned.

In the second experiment instead of following the person, the test subject had to drive the wheelchair from starting point to goal point by maneuver the wheelchair by himself using his eye through a complex environment. Obstacles were placed in the path and the user was supposed to avoid those obstacles just by the movement of his eyes. The accuracy of the gaze estimation algorithm was calculated by finding the difference between path followed by these users and the actual path which they had to follow. In this experiment more than 70% of our users were able to reach the goal point with little effort. For 30 percent users we had to retune some of the features to improve eye detection and tracking.

4.1.3 Testing in an rviz Simulation Environment

An rviz simulation environment was created by the wheelchair research group in RISE Lab, which was used to test different algorithms, before actually testing them on wheelchair. This simulation environment contained the model of wheelchair, including all the sensors. Commands can be sent and data from sensors can be received in a similar fashion as with a real robot through ROS.

The eye detection and tracking algorithm was tested in this simulation environment on multiple test subjects for training and testing purpose. Figure 11 shows the rviz environment with the wheelchair model driven by a person using an eye tracking, only with the movement of eye. This simulation environment was loaded without any world, however a world of choice can be loaded to test an algorithm in the required kind of environment. A world can also be created using rviz tools.

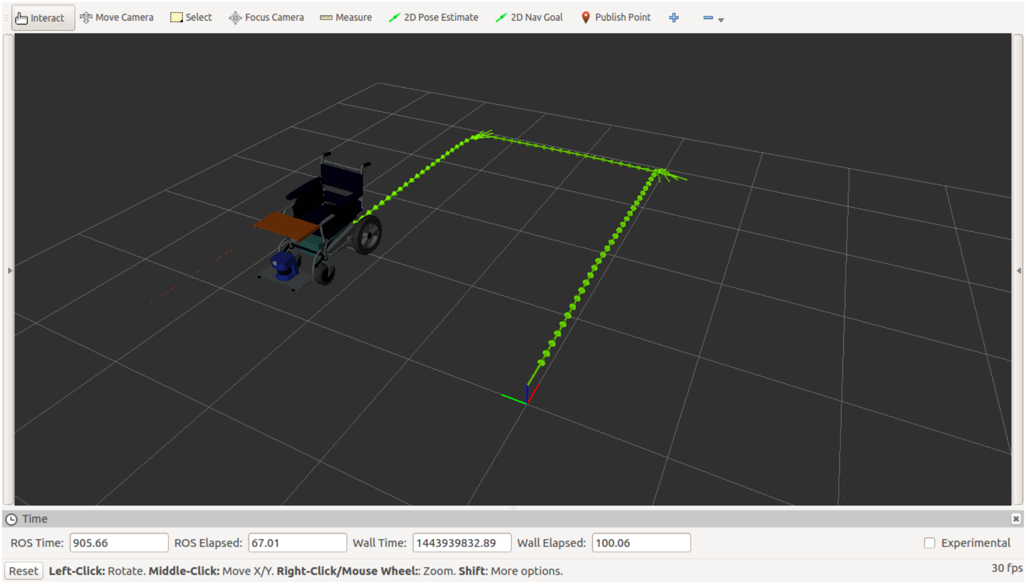

Figure 11: RVIZ Model of wheelchair and simulation environment in rviz. The green dotted lines show the trail of path of wheelchair which was driven by using the eye tracker.

4.2 Limitations

Though the algorithm for detection and tracking of eyes was very accurate and communication between the eye tracking module and wheelchair module was fast and lag-less, still there are few limitations of this project which needs to be catered before it can be transformed into a completely usable product.

The first problem was tuning of eye detector for different people, depending upon the size and shape of eye. The parameters used for one person, sometimes could not be used for another person with a different size or shape of the eye. A method which is able to adapt to any kind of eye and tune the size and shape features automatically is important.

Second limitation was the calibration process. Although we are using only a single point calibration, still it requires the user to calibrate the eye tracker before being able to use it. In order to remove this limitation, an automatic calibration process needs to determined. For a disabled person, calibrating the eye tracker is a tough job, so automatic calibration is a very important feature that needs to be added.

4.3 Conclusion

This project was aimed to find a method to facilitate such patients who cannot use their arm muscles or are amputated and cannot even use a motorized wheelchair, in moving and controlling a wheelchair by some other means. Different techniques developed by different research groups and their advantages and disadvantage were explained in this thesis. These techniques are broadly classified as, head movement based, EEG based, EOG based, tongue movement based and eye tracking based techniques. The technique used in this research was eye tracking based technique, and an algorithm was developed for detection, tracking and driving the wheelchair.

5 Bibliography

[1] "What is ALS?," *ALSA.org*. [Online]. Available: http://www.alsa.org/about-als/what-is-als.html. [Accessed: 25-Sep-2016].

[2] U. Siraj, "Real-time Stereo Visual Odometry and its application to Semi-Autonomous Wheelchair," National University of Science and Technology, Islamabad Pakistan, 2016.

[3] "Milestones Project." [Online]. Available: http://www.milestone.pk/User/Projects. [Accessed: 25-Sep-2016].

[4] *Guidelines on the Provision of Manual Wheelchairs in Less Resourced Settings*. Geneva: World Health Organization, 2008.

[5] G. C. Rascanu and R. Solea, "Electric wheelchair control for people with locomotor disabilities using eye movements," in *2011 15th International Conference on System Theory, Control, and Computing (ICSTCC),* 2011, pp. 1–5.

[6] K. A. Unnikrishna Menon, R. Jayaram, and P. Divya, "Wearable wireless tongue controlled assistive device using optical sensors," in *2013 Tenth International Conference on Wireless and Optical Communications Networks (WOCN)*, 2013, pp. 1–5.

[7] E. J. Rechy-Ramirez, H. Hu, and K. McDonald-Maier, "Head movements based control of an intelligent wheelchair in an indoor environment," in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2012, pp. 1464–1469.

[8] A. Z. Neto, A. Mesquita, M. Machado Spindola, and M. Magnani, "Prototype of a Wheelchair Controlled by Cervical Movements," in *2014 IEEE International Conference on Bioinformatics and Bioengineering (BIBE),* 2014, pp. 134–140.

[9] P. B. Taylor and H. T. Nguyen, "Performance of a head-movement interface for wheelchair control," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2003,* 2003, vol. 2, pp. 1590–1593 Vol.2.

[10] R. Upadhyay, P. K. Kankar, P. K. Padhy, and V. K. Gupta, "Classification of drowsy and controlled EEG signals," in *2012 Nirma University International Conference on Engineering (NUiCONE)*, 2012, pp. 1–4.

[11] C. S. Ashwin, K. R. Rangarajan, and S. Ramachandran, "Disance (enhance the disabled) brain mapped mobility for physically challenged," in *2010*

2nd International Conference on Software Engineering and Data Mining (SEDM), 2010, pp. 555–560.

[12] D. Puanhvuan, S. Khemmachotikun, P. Wechakam, B. Wijam, and Y. Wongsawat, “Automated navigation system for eye-based wheelchair controls,” in *Biomedical Engineering International Conference (BMEiCON), 2014 7th*, 2014, pp. 1–4.

[13] R. J. K. Jacob and K. S. Karn, “Eye Tracking in Human–Computer Interaction and Usability Research: Ready to Deliver the Promises,” .

[14] W. Tunhua, B. Baogang, Z. Changle, L. Shaozi, and L. Kunhui, “Real-time non-intrusive eye tracking for human-computer interaction,” in *2010 5th International Conference on Computer Science and Education (ICCSE)*, 2010, pp. 1092–1096.

[15] W. D, “Gaze Tracking Using A Regular Web Camera,” 2012.

[16] A. T. Duchowski, E. Medlin, N. Cournia, A. Gramopadhye, B. Melloy, and S. Nair, “3D Eye Movement Analysis for VR Visual Inspection Training,” in *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*, New York, NY, USA, 2002, pp. 103–110.

[17] F. Timm and E. Barth, “Accurate Eye Centre Localisation by Means of Gradients.,” in *ResearchGate*, 2011, pp. 125–130.

[18] R. J. K. Jacob, “Virtual Environments and Advanced Interface Design,” W. Barfield and T. A. Furness III, Eds. New York, NY, USA: Oxford University Press, Inc., 1995, pp. 258–288.

[19] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, “Feature extraction from faces using deformable templates,” *Int. J. Comput. Vis.*, vol. 8, no. 2, pp. 99–111, 1992.

[20] X. Xie, R. Sudhakar, and H. Zhuang, “On improving eye feature extraction using deformable templates,” *Pattern Recognit.*, vol. 27, no. 6, pp. 791–799, Jun. 1994.

[21] Kin-Man Lam and Hong Yan, “Locating and extracting the eye in human face images,” *Pattern Recognit.*, vol. 29, no. 5, pp. 771–779, May 1996.

[22] L. Zhang, “Estimation of eye and mouth corner point positions in a knowledge-based coding system,” 1996, vol. 2952, pp. 21–28.

[23] M. Kampmann, L. Zhang, and L. Zhang, “Estimation of Eye, Eyebrow and Nose Features in Videophone Sequences,” in *in Proc. International Workshop on Very Low Bitrate Video Coding*, 1998, pp. 101–104.

[24] G. C. Feng and P. C. Yuen, “Variance projection function and its application to eye detection for human face recognition,” *Pattern Recognit. Lett.*, vol. 19, no. 9, pp. 899–906, Jul. 1998.

[25] G. C. Feng and P. C. Yuen, “Multi cues eye detection on gray intensity images,” *Pattern Recognit.*, pp. 1033–1046, 2001.

[26] M. Nixon, “<title>Eye Spacing Measurement for Facial Recognition</title>,” 1985, pp. 279–285.

[27] P. W. Hallinan, “Recognizing human eyes,” 1991, vol. 1570, pp. 214–226.

[28] A. Pentland, B. Moghaddam, and T. Starner, “View-based and modular eigenspaces for face recognition,” in *, 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994. Proceedings CVPR ’94*, 1994, pp. 84–91.

[29] W. Huang and R. Mariani, “Face detection and precise eyes location,” in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, 2000, vol. 4, pp. 722–727.

[30] J. Huang and H. Wechsler, "EYE DETECTION USING OPTIMAL WAVELET PACKETS AND RADIAL BASIS FUNCTIONS (RBFs)," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 13, no. 07, pp. 1009–1025, Nov. 1999.
[31] S. Kawato and J. Ohya, "Two-step approach for real-time eye tracking with a new filtering technique," in *2000 IEEE International Conference on Systems, Man, and Cybernetics*, 2000, vol. 2, pp. 1366–1371 vol.2.
[32] S. Kawato and J. Ohya, "Real-time detection of nodding and head-shaking by directly detecting and tracking the 'between-eyes,'" 2000, pp. 40–45.
[33] S. Kawato and N. Tetsutani, "Real-time detection of between-the-eyes with a circle frequency filter," in *Proceedings of The 5th asian conference on computer vision (ACCV2002)*, 2002, vol. 2, pp. 442–447.
[34] S. Kawato and N. Tetsutani, "Detection and tracking of eyes for gaze-camera control," *Image Vis. Comput.*, vol. 22, no. 12, pp. 1031–1038, Oct. 2004.
[35] Y. Tian, T. Kanade, and J. F. Cohn, "Dual-state parametric eye tracking," in *In Proceedings of Conference on Automatic Face and Gesture Recognition*, 2000, pp. 110–115.
[36] M. J. T. Reinders, R. W. C. Koch, and J. J. Gerbrands, "Locating Facial Features in Image Sequences using Neural Networks," in *In 2 nd International Conference on Automatic Face and Gesture Recognition*, 1997, pp. 230–235.
[37] S. Sirohey, A. Rosenfeld, and Z. Duric, "A method of detecting and tracking irises and eyelids in video," *Pattern Recognit.*, vol. 35, no. 6, pp. 1389–1401, 2002.
[38] S. A. Sirohey and A. Rosenfeld, "Eye detection in a face image using linear and nonlinear filters," *Pattern Recognit.*, vol. 34, no. 7, pp. 1367–1391, 2001.
[39] B. D. Lucas, T. Kanade, and others, "An iterative image registration technique with an application to stereo vision.," in *IJCAI*, 1981, vol. 81, pp. 674–679.
[40] "Eye Tracking | Advanced Eye-Tracking Technology by LC Technologies." [Online]. Available: http://www.eyegaze.com/. [Accessed: 25-Sep-2016].
[41] "Applied Sciences Laboratory." [Online]. Available: http://www.asl.wsu.edu/. [Accessed: 25-Sep-2016].
[42] Y. Ebisawa and S. i Satoh, "Effectiveness of pupil area detection technique using two light sources and image difference method," in *Proceedings of the 15th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1993*, 1993, pp. 1268–1269.
[43] Y. Ebisawa, "Improved video-based eye-gaze detection method," *IEEE Trans. Instrum. Meas.*, vol. 47, no. 4, pp. 948–955, 1998.
[44] C. H. Morimoto, D. Koons, A. Amir, and M. Flickner, "Pupil detection and tracking using multiple light sources," *Image Vis. Comput.*, vol. 18, no. 4, pp. 331–335, 2000.
[45] P. Viola, M. J. Jones, and D. Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance," *Int. J. Comput. Vis.*, vol. 63, no. 2, pp. 153–161, Feb. 2005.
[46] A. Haro, M. Flickner, and I. Essa, "Detecting and tracking eyes by using their physiological properties, dynamics, and appearance," in *IEEE*

Conference on Computer Vision and Pattern Recognition, 2000. Proceedings, 2000, vol. 1, pp. 163–168 vol.1.
[47] Q. Ji and X. Yang, "Real time visual cues extraction for monitoring driver vigilance," in *International Conference on Computer Vision Systems*, 2001, pp. 107–124.
[48] Z. Zhu and Q. Ji, "Robust real-time eye detection and tracking under variable lighting conditions and various face orientations," *Comput. Vis. Image Underst.*, vol. 98, no. 1, pp. 124–154, 2005.
[49] R. J. Jacob, "Eye tracking in advanced interface design," *Virtual Environ. Adv. Interface Des.*, pp. 258–288, 1995.
[50] K. Rayner, "Eye movements in reading and information processing: 20 years of research.," *Psychol. Bull.*, vol. 124, no. 3, p. 372, 1998.
[51] R. J. Jacob, "The use of eye movements in human-computer interaction techniques: what you look at is what you get," *ACM Trans. Inf. Syst. TOIS*, vol. 9, no. 2, pp. 152–169, 1991.
[52] S. Klingenhoefer and F. Bremmer, "Saccadic suppression of displacement in face of saccade adaptation," *Vision Res.*, vol. 51, no. 8, pp. 881–889, Apr. 2011.
[53] S. Grossberg, K. Srihasam, and D. Bullock, "Neural dynamics of saccadic and smooth pursuit eye movement coordination during visual tracking of unpredictably moving targets," *Neural Netw. Off. J. Int. Neural Netw. Soc.*, vol. 27, pp. 1–20, Mar. 2012.
[54] W. Sewell and O. Komogortsev, "Real-time Eye Gaze Tracking with an Unmodified Commodity Webcam Employing a Neural Network," in *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, New York, NY, USA, 2010, pp. 3739–3744.
[55] Y. Huang, Z. Wang, and A. Ping, "Non-contact gaze tracking with head movement adaptation based on single camera," *World Acad. Sci. Eng. Technol. 59 2009*, 2009.
[56] Y. Huang, Z. Wang, and X. Tu, "A real-time compensation strategy for non-contact gaze tracking under natural head movement," *Chin J Electron*, vol. 19, pp. 446–450, 2010.
[57] "VL-EBX-12e VersaLogic Single Board Computer - Industrial.net." [Online]. Available: https://vl-ebx-12e.industrial.net/. [Accessed: 01-Oct-2016].
[58] M. Quigley *et al.*, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, 2009, vol. 3, p. 5.
[59] D. Li, D. Winfield, and D. J. Parkhurst, "Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, 2005, pp. 79–79.
[60] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001*, 2001, vol. 1, pp. I–511–I–518 vol.1.
[61] P. I. Wilson and J. Fernandez, "Facial Feature Detection Using Haar Classifiers," *J Comput Sci Coll*, vol. 21, no. 4, pp. 127–133, Apr. 2006.
[62] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, 2002, vol. 1, pp. I–900.

[63] P. Menezes, J. C. Barreto, and J. Dias, “Face tracking based on haar-like features and eigenfaces,” in *IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, 2004, vol. 500.

[64] Y. Freund and R. E. Schapire, “A desicion-theoretic generalization of on-line learning and an application to boosting,” in *European conference on computational learning theory*, 1995, pp. 23–37.

[65] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, “Boosting the margin: A new explanation for the effectiveness of voting methods,” *Ann. Stat.*, pp. 1651–1686, 1998.

[66] K. Tieu and P. Viola, “Boosting image retrieval,” *Int. J. Comput. Vis.*, vol. 56, no. 1–2, pp. 17–36, 2004.

[67] Z. Zhu and Q. Ji, “Novel Eye Gaze Tracking Techniques Under Natural Head Movement,” *IEEE Trans. Biomed. Eng.*, vol. 54, no. 12, pp. 2246–2260, Dec. 2007.

[68] R. Kothari and J. L. Mitchell, “Detection of eye locations in unconstrained visual images,” in *, International Conference on Image Processing, 1996. Proceedings*, 1996, vol. 3, pp. 519–522 vol.3.