High Accuracy Eye Tracking for Proton Therapy

R Gokul Nayar

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

High Accuracy Eye Tracking for Proton Therapy

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

R Gokul Nayar

November 17, 2017

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Abstract

Accurate and safe proton beam delivery is one of the most crucial tasks during Proton Therapy (PT) of ocular melanoma. The eye movement and gaze angle tracking system should be able to monitor in real time the eye position and orientation (6 degree of freedom) to exactly localize the tumor location inside the eye with respect to the proton beam. The system should also immediately switch the beam off if the tumor goes out of the irradiated area to protect vital organs and keep the non affected cells healthy. The non-invasive eye-tracking system will replace the painful surgical procedure of implantation of radio-opaque tantalum clips on the eye. In order to estimate accurately enough the eye gaze angle and torsion, a stereo imaging system consisting of two high-resolution imaging cameras and two infra-red beacons can be used. The six coordinates of the eye are extracted by image analysis of the acquired stereo camera images using the beacons reflections (glints) and location of the eye pupil. The accuracy of the method can be affected by motion artifacts and difficulty of pupil segmentation in some eyes. Further, the method is inclusive to cyclo-torsion (rotation of the eye about its optical axis).

The eye-tracking system can thus benefit from motion artifacts suppression and implementation of analysis of the eye surface features, e.g. iris pattern. Another addition is to use steerable mounts and mirrors to improve measurements and, thus, eye tracking accuracy. The goal of this thesis is to build a prototype stereo eye-tracking system, write a tracking code and investigate the viability of this method for further improving eye tracking accuracy.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisor prof.dr.ir. M. Verhaegen and dr. O.A. Soloviev for the opportunity to work towards tackling the challenges in the field of eye tracking with complete freedom to work and develop this prototype.

I would specially thank dr. O.A. Soloviev for his constant availability towards directing my questions and giving me advice. Mainly, the patience and assistance throughout the ups and downs of writing this thesis. I am also thankful to dr. Paolo Pozzi for his help with all hardware help in the laboratory.

Finally, I express my deepest gratitude to my friends who have constantly supported and motivated me along the way. Being specially grateful to my sibling and my parents, who have provided me through moral and emotional support in my life.

Delft, University of Technology
November 17, 2017

R Gokul Nayar

"It does not matter how slowly you go as long as you do not stop.”

— *Confucius*

Chapter 1

Introduction

Uveal melanoma is a malignancy that threatens both eye sight and life. Affected cells grow into tumor masses and metastasize through the surrounding tissues. Today's eye preserving treatment modalities such as radiotherapy, local resection and transpupillary thermotherapy have the same survival rates as removal of the eye (enucleation), which was used in the past. Making it possible to preserve vision and quality of life. For larger sized tumors radiation treatment offers the highest chance to save the eye. Furthermore, Proton Therapy (PT) offers treatment with fewer side effects in comparison to the other forms of radiotherapy.

Current process involved in the Proton Therapy are complex and consists of invasive procedures for treatment delivery. Further, limitations in treatment planning and delivery increases the risk of severe side-effects such as irreversible destruction of the lens, neovascular glaucoma, radiation damage to the retina.

Goal of this thesis is to investigate an eye tracking technology by building a prototype for non invasive system of tumor localization used during diagnosis of Uveal melanoma and mainly during the treatment delivery stage.

1-1 Motivation

With the vision for a new proton therapy center, HollandPTC is funded as an independent polyclinical center for proton therapy and scientific research, based in Delft. Within which PROTONS4vision is a project for developing technologies that provide a more targeted and less invasive treatment. This thesis aims to aid the project research by building a prototype and investigating approach to the non invasive treatment for Uveal Melanoma. We begin with understanding our application, procedure and it's environment.

1-1-1 Proton Therapy

The advantage of a Proton therapy over traditional photon based radiation therapy lies in the main property of the protons, based on the release energy of the beam from the delivery system, it penetrates the human tissue with low dose and stop at a particular depth due to the nature of proton, releasing all it's energy at this depth. This peak dose is called a 'Bragg Peak' of ionization, with relative sparing of healthy tissues both proximal and distal to the tumor target. A proton when emitted with high energy, can destroy the DNA structure of the cancer cells, killing it's growth. And due to the positive charge of a proton, they can be steered in any direction and energized by a strong magnetic field. This is done in a Particle Accelerator. During the beam's first entry of human tissue, the initial energy transfers to molecular electron clouds on their path are of relatively of small amounts (low degree of ionization).

Using cyclotrons and synchrotrons the protons are accelerated to 60% of the speed of light (180,000 km/s, 250 MeV of kinetic energy), which can penetrate approximately 38cm into the body. Penetration depth and the direction are controlled by controlling the magnetic field. The patient sits in front of the delivery tube on a chair with electronically controlled servo motors with six degrees of freedom as shown by the layout in Figure 1-1.

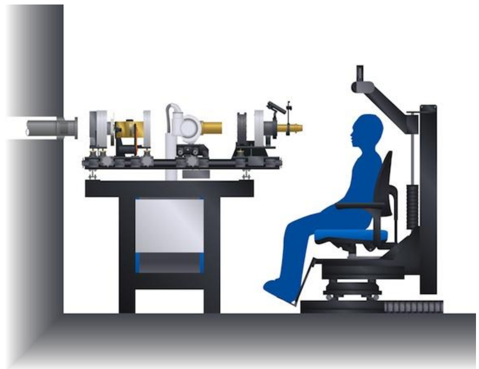

Figure 1-1: Treatment Layout: Depicting the proton beam delivery system in front of the patient's eye. Patient is seated on a treatment chair which immobilizes the patient but allows for 6Degree of Freedom movement and correction with the required angle of attack of tumor.

1-1-2 Uveal Melanoma Treatment

With respect to the conventional linac-based radiation therapy, proton therapy has superior dosimetric properties including an energy deposition that increases with depth reaching a maximum energy deposition, the Bragg peak, near the end of the proton beam range and no exit dose beyond the falloff. A typical treatment session for a uveal melanoma can lasts for couple of minutes, during which there is a possibility for the patient to inadvertently move the eye ball from the planned treatment direction, which need to is needed to measured and corrected for, so as to keep away from targeting the healthy tissues. Apart from some early modifications with immobilization of the patient and targeting the proton beam, the current methodology has essentially been unchanged since late 1970s. 3 to 5 Tantalum rings(radio-opaque) are sewn to the outside of the eyeball (sclera) at the edges of the tumor with relative distance measurements recorded. Once the markers are in position, we make use of a stereo x-ray system that are perpendicular to each others imaging direction. This imagining tool is used to reconstruct the location of the tantalum rings from which the tumor position can be relatively measured.

Reformation of Treatment: The imaging method of stereo x-rays requires surgery of the eye, to place the rings.This drastically increases the complexity of the system and possibilities of error. Hence, we look into non invasive methods that help us overcome the invasive preparation of the patients. Tumor must be targeted by recording the eye movements accurately. MRI provides 3D data along with the potential of a high resolution diagnosis and treatment planning. Though, significant artifacts are caused during the image reconstruction caused due to the eye movements during the MRI procedure. We can overcome this with a specific gaze tracking system, which is used in two stages, presented independently of the actual treatment/diagnosis system. First, to measure the exact orientation of the eye during the MRI examination and the next to register the eye orientation during proton therapy. This brings us to the problem formulation to our thesis.

Problem Formulation Even though proton therapy have been used earlier for the treatment of ocular tumors, the problem of eye movement during treatment still remains [1]. As any unmitigated eye movements turn the advantage of the Bragg peak into a major disadvantage in this regard. The required eye orientation is planned by the oncologist during diagnosis, but requires validation of stability during the treatment process.

This calls for an accurate and reliable techniques for eye tracking used for beam gating. The purpose of this thesis was to develop an flexible and automated eye-tracking system suitable for use in proton therapy. With an eye tracking system that uses a non invasive method, by which the tumor volume is localized during proton dose delivery by tracking of the anterior eye segment and registration in 6 Degrees of Freedom (translation, rotation, cyclotorsion). We approach this by setting objectives for this thesis.

1-2 Objectives

Our main objective for this thesis and study, to develop an automated eye tracking system that can be installed in a proton therapy clinical environment with flexibility to move the device for patient-specific requirements (which vary with location of tumor, patient disabilities, plan of attack for the specific tumor growth etc).

Most of proton therapy institutions have employed qualitative methods, whereby, eye position is monitored with an initial position marked on a monitor and the oncologist manually monitors the eye to account for gating the system due to large eye movements. Another purpose of this study is to develop an automated eye-tracking for the beam gating which are suitable for use in proton therapy clinics for uveal melanoma. Device outputs any changes in the orientation during the course of treatment (Eye orientation with cyclo-torsion). The overall objectives are as follows:

1. Flexibility: The system should work automatically to an extent where there is no additional load (restriction devices such as head masks) to the patient and least amount of operator intervention. As the beam delivery is in front of the eye, our system should not be rigid to hardware construction by accounting for the limited space in a clinic. We limit our equipment work space to 200 cm^3 with the flexibility to expand and introduce other essential clinical equipment.

2. Computational Speed: The gating system of the proton beam delivery currently has a limitation of 20ms or 50Hz. Hence, our device should be able to capture all the movements of the eye muscles by incorporating fast computational time meeting the specifications > 50Hz.

3. Non Invasive: The eye tracking method should be completely non invasive in nature, to reduce any possibilities for side effects. For example, using safe illumination sources, which maybe used for longer periods of time without damaging the eye. Working with low power sources for any harmful wavelength spectrum of the illumination source.

4. Ease of use: Have least amount of human intervention. We aim to produce an interactive Graphical User Interface (GUI) to keep the complexity to a minimal, for the use of this system by radiation oncologist and to be able to modify data for testing. This can be achieved by application compilers in most languages (C#, MATLAB).

5. Upgrade: The method should with low specification hardware and allows room for better on performance by improving the configuration of the system, such as, the resolution of the camera from (1280x1024), or by increasing the number of pixels per feature (such as, Pupil > 70pixels).

1-3 Outline

The body of this thesis is divided into four main chapters. With the first two explaining the study and the main theory behind our proposed gaze estimation method. The next one chapter explain the process of building the experimental prototype design. Finally, we discuss the sensitivity of our tracking device and results from the prototype. Each chapter is briefly explained as follows:

- In **Chapter 2** we show some of the existing eye tracking devices and how they work. Here, we pick a method for our device. We also discuss the anatomical eye and how it might restrict the accuracy of our eye tracking device.

- **Chapter 3** shows the optical eye model that is used in describing the features of the eye and the theory behind first order approximation of the camera model. Following this is the theory with mathematical model for the gaze estimation with different possible configurations for the setup.

- In **Chapter 4** the complete development of the experimental prototype is explained with performance analysis of the vision algorithms that have been used for detection.

- **Chapter 5** discusses the sensitivity of the model that we used and how they correspond with the results obtained from our setup.

- Finally, In **Chapter 6** we conclude the thesis with possible improvements for a future device.

1-4 Nomenclature

a	Scalars values: Lower or uppercase alphabets
$\mathbf{a}$	Vectors : Lowercase bold alphabets
$\mathbf{A}$	Matrices : Uppercase bold alphabets
i, j	Indexing: Subscripts using the alphabets
$\tilde{\mathbf{a}}$	Augmented vectors: diacritical mark(Tilde) placed over an alphabet
$\mathbf{A}^{-1}, \mathbf{A}^T$	Inverse and Transpose: Superscript using -1 and T respectively
$\bullet$	Scalar product or Dot product
$\times$	Cross product or Vector product
σ_i	i^{th} largest singular value of a matrix
$\kappa(\mathbf{A})$	Condition number of a matrix being the ratio of largest to smallest singular value
$\Vert\mathbf{a}\Vert$	Euclidean norm
σ^2	Variance or dispersion of a set of data
σ	Standard Deviation or Square root of Variance
$\sum_{i=1}^{n} x_j$	To express summation of many variables with i as the index and number succeeding as the starting point, stopping point as superscript and x_j as the j^{th} variable

While the subsections of topics entail their specific notations, we aim to have a unified notation throughout the thesis. As and when the unified notation could not be used, they have been explained in the introductory to that section.

Chapter 2

Eye Tacking Overview

An eye tracker is a general term for a device that is used for either location of eye among other feature and/or measuring of eye movement within eye ball socket itself. Our use for this term is the latter definition. We use this chapter to understand the existing technologies of eye tracking devices as they are increasingly becoming popular in applications such as Human-Computer Interaction (HCI) for Assistive technology for people with disabilities (some devices developed by [2], [3], [4]), assistant systems used in cars (as seen in [5], [6], [7]) and virtual reality, human behavior studies, etc. These technologies are used in the commercial market, but currently are very expensive and depending on the application, complex in terms of implementation.

2-1 Eye Tracking Techniques

Various eye tracking research have been presented through the history, some as early as 1737 [8]. However, with the constant development in technologies wider exploration has become possible.

Types of Eye Tracker: All the techniques currently used for tracking the motion of the eye can fundamentally be classified into these three types:

- *Eye-Attached Sensor Tracking :* With a sensor similar to a contact lens with an embedded mirror [9] or magnetic filed sensor to measure change in electromagnetic field [10]. We can read the measurement off the sensors with the assumption that it does not slip significantly with the eye movements. It allows for measuring movements in horizontal, vertical and torsion directions.
- *Electric Potential Measurement :* Electrodes placed in the vicinity of the eye measure any change in electric potential. Allowing to measure in total darkness and also when the eye lids are closed. Potential drifts and variable relations between the

Electrooculography (EOG) signal amplitudes and the saccade sizes, make it challenging to use Electrooculography for measuring slow eye movement and detecting gaze direction. The major disadvantage of EOG is its relatively poor gaze direction accuracy compared to a video tracker.

- *Optical / Vision Based Tracking :* These methods are non invasive and uses optical instruments or digital cameras for measuring the eye movements. Primarily the light reflected off by the eye is recorded and analyzed to get the gaze angle. Similar method of tracking is to image features from inside the eye, such as the retinal blood vessels, and follow these features as the eye rotates.

  1. Based on hardware design
     - *Head Mounted Systems*
       With the assumption that the head mounts are very stable, the measurement in these are independent to the head position.
     - *Remote Estimation Systems*
       Remote devices require patients to sit in front of the Field of view (FOV) of the vision system. Although, remote devices can only track the eyes within certain limits (the headbox), the freedom of movement is still sufficiently large so as to feel unrestricted.
  2. Based on software algorithms
     - Pattern Recognition: Use of template matching to compare the pixels to the match for patterns exhibited by a eye. This appraoaches have been used immesenly, examples in [3], [11] [12].
     - Purkinje Images: This method is one of the most affective method with lot of room for improvements. Vector from center of the pupil and the corneal reflections is used to compute the gaze direction. Though a calibration procedure of the particular eye is needed, examples for these are in [13], [14], [15], [16]. This method can be understood better with the explanation in Section 2-2.
     - Feature Shape : This method employs the use of change in pupil shape with different gaze directions ([17], [18]).

Feature and Model based approaches: The Video based eye tracking algorithms can be additionally be classified into feature based or model based approach.

Feature based approach typically detect and localize image features related to the position of the eye. As our study used this approach, algorithms for detection are discussed in detail along with our prototype in the later section. But a general criteria is to determine the presence of a feature with a specific threshold, which in most eye trackers rely on the user or try to calculate with the histogram. As this deeply rely on the intensity gradients of the image. After segmenting the feature of interest we can use Least-Squares or Circular Hough Transform to fit a circle or ellipse to detect these features. This is explained in detail in implementation part of the thesis.

Model based approaches find the best fitting model that is consistent with the image. Using integro-differential operator to get the best-fit circle or an ellipse for pupil contour [19].

These provide a better estimate but can be stuck in local minimum as it is searching a complex parameter space. This reduces the possibility of real time performance due to cost in computational speed.

To Summarize: The Eye-Attached sensor tracking is most intrusive and invasive method. Restricting the length of measurements to around 30 minutes. The EOG method involves a large electrodes around the eye, which is cumbersome and uncomfortable to the patient and creates limitations to a clinical environment and does not offer flexibility. The Optical Tracking requires no direct physical contact to the eye creating a non-invasive measurement device with it's accuracy depending on the resolution of images and the accuracy of feature detection itself. But with increasing vision algorithms and computational power constant development in this method is taking place and still has possibility of great improvements which we investigate in the thesis.

2-2 Video Based - Corneal Reflection Method

Within the feature based method, an addition to the feature is by looking into the reflections cause by the eye surface. This is currently one of the most used method for eye tracking applications due the possibility of improving the accuracy with varied algorithms and better resolution cameras. But this greatly varies in term of implementation. Fundamentally, in this method, the system uses the reflection of an illumination source, either active(controlled illumination) or passive(reflection of surroundings) to gain information on the visual axis of the eye. In this thesis, only the active illumination is considered since this is a controlled method and reliable with environmental changes. In this section let us understand these reflections and then discuss how we can use them to gain information for point of gaze. The reflections that occur on the surface of the eye is called as the Purkinje Image.

Purkinje Images: Purkinje images are simply reflections from the visible structure of the eye. They are termed as Purkinje reflexes or Purkinje images. At least four Purkinje images are usually visible, but they vary greatly in their intensity of their illumination, which governed by Eq. (2-2).

- The first Purkinje image (P1) is the reflection from the outer surface of the cornea.
- The second Purkinje image (P2) is the reflection from the inner surface of the cornea.
- The third Purkinje image (P3) is the reflection from the outer (anterior) surface of the lens.
- The fourth Purkinje image (P4) is the reflection from the inner (posterior) surface of the lens. Unlike the others, P4 is an inverted image.

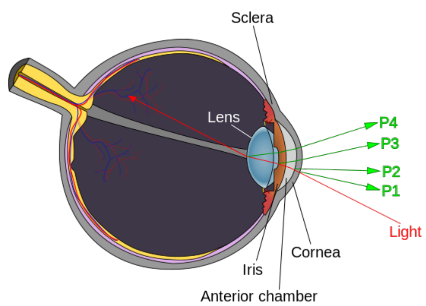

Figure 2-1: Diagram of light being reflected to form the four Purkinje images(P1, P2, P3 and P4). With the highest intensity for the reflection P1 (Credits:Wikipedia).

The first and fourth Purkinje images are used by some
eye trackers, devices to measure the position of an eye.
But the feature strength is varied, hence would require different algorithms to detect them. The brightness of the Purkinje images is determined using Fresnel's equation:

$$I = \frac{(n' - n)^2}{(n' + n)^2}, \tag{2-1}$$

where, I is the intensity value, n and n' are the refractive indices before and after the reflecting surface, respectively.

Purkinje image P1 will be the brightest of all the four, following with, P3 and P4 (with almost same brightness), finally P2. The Purkinje images can further be used to assess the curvatures and separations of the surfaces within the eye's anatomy. With P1 producing the highest intensity, our prototype tracker will focus on the cornea reflection (P1), generally also known as glint. Purkinje image can be triggered by both active light and passive or ambient lighting conditions.

Use of active illumination for Eye Tracking: Using the active illumination to create a Purkinje Image, is a method employed by various eye trackers. This method works by adding an active illumination source in a known position and controlling the illumination using a digital or analog signal. Reflection of this light source from the cornea can be modeled accurately with few fundamental assumptions. First assumption is that cornea has a perfect sphere shape, causing the glint to stay in the same position for any direction of gaze while the pupil moves, unless for head movements. Hence, the feature vector (connecting pupil center and the corneal reflections) is used to compute the gaze direction. Although, a simple calibration procedure(asking patient to look into fixation targets) of the specific patient is required by the tracker to set the parameters before using the eye tracker for absolute gaze estimation.

2-3 Current Challenges

Study of proton therapy environment and the available eye trackers gives us insight on the challenges of using an eye tracker, which can be broadly classified into technical and anatomical challenges. With various eye tracking technologies available on the market today, the challenges are more focused towards our specific application in proton clinic environment.

Technical Challenges: Main challenges lie in the balance between speed and resolution. As the image resolution goes higher, our search space is increased, requiring more computation time. Some other methods used, underlie with certain requirements for the tracker to work, creating a limitation in space and flexibility. For example, *the bright and dark pupil method* requires mechanical attachments, which can be challenging in an proton clinical environment. Illumination changes in the environment also cause disturbance to the vision algorithm. Challenges in the procedure is accounted in the calibration for visual axis from the optical axis, this is challenging as there is no way to verify if the measurements are accurate, most trackers approximate over large measurements. Hardware challenges also play a role, such as the frame rate of the digital camera being slower than algorithm, slowing down the detection frequency.

Patient Aspect: The eye trackers use calibration techniques in which we rely on the patient to look into a fixation target, which can not be verified as the patient may/may not be looking exactly at the target during the frame acquisition. This bring in a certain uncertainty. On the other hand, as explained in the section below, the human eye is limited to 1° of visual angle, being a major contributing factor to the low accuracy of the eye tracker which use calibration with fixation targets. The patient during the course of treatment will also resort to blinking, which either needs to be mechanically constrained during treatment with an additional attachment or detected with the program, to be able to effectively gate the proton beam.

2-4 The Eye and Movements

Let us examine what causes the motions of the human eye, accuracy in the anatomy of the eye and how we can optically model the eye to be able to use it further for the gaze detection with our eye tracker.

2-4-1 Anatomy

Shape of an eye varies from person based on DNA and age. But they all have similarities. It is a fusion of two units, composed of the anterior segment and the posterior segment rather than being a perfect sphere. The anterior segment consists of the parts called cornea, iris and lens, which are our main concern in this thesis. Six muscles that are around the ball and that control the movement of the eye with one extra muscle that controls eye lid elevation. The muscles are attached antagonistically in pairs giving 3 Degree of Freedom (DOF) to the eye. The pair of human eyes itself act similar to a stereoscopic camera with each eye having rod and cone cells in the retina allowing conscious light perception and vision including color differentiation and the perception of depth. Iris behaves as the aperture setting of the eye, opening and closing the pupil.

2-4-2 Accuracy in Fovea

In this section we try to analyze the accuracy that can be achieved anatomically by the human eye. The photo receptors of the eye are unevenly distributed over the retina. The distribution is visualized in the Figure 2-2. For an eye to focus on a target, it needs a stable projection on the retina. This is achieved by the eye muscles compensating the movements of the head by inverting any head movements. However, if the target is in a stable position, the eye gaze initially jumps abruptly with movements called saccades. Which slowly moves into rest position called fixation after the micro-saccades. These saccades occur due to the 1° visual angle size of the fovea, to get the target projected onto the fovea, saccades help align the projection. In this thesis, calculate the optic axis, which is independent of the fixation target and would allow us for an accurate estimation. With a low frame rate camera, gaze drifting and micro-saccades can cause the eye tracker accuracy to be limited to $\pm 0.5°$ while locating for the absolute gaze.

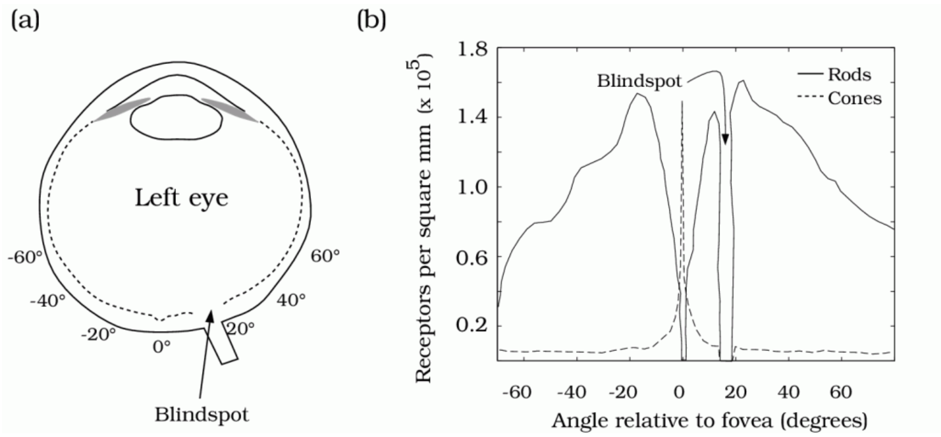

Figure 2-2: Distribution of photo receptors on the retina with highest concetration on the fovea and with no receptors over the blind spot of the eye. Credits: [20]

2-4-3 Optical Model

There are various optical eye models that are used for approximating a complex human eye. Details on the other models are described in the Appendix B. Reduced Models B-1-1 are anatomically incorrect since there is no crystalline lens and being compensated by a powerful cornea having a short length. They fail to demonstrate accommodation, yet they can be functionally accurate, with the cardinal points near to correct positions. The Four Refracting Surface Models require parameters that are patient specific which might be useful in the future, where the data from the diagnosis stage might be used to optimize this model. Also, the cornea of eye has an aspherical shape which is not yet accommodated for. This is taken as a future development as the complexity is high.

The Gullstrand-LeGrands schematic eye is a convenient and accurate model for optical tracing and to model the features that can be detected with a Video-oculography (VOG) device. We will be using this eye model for our gaze estimation model.

Axis Alignment : It is important to note that the eye has two distinct axes, namely, the optical axis and the visual axis as shown by the Figure 2-3. The optic axis is a direct axis through the center of cornea to the pupil, the lens, and the retina. Since the intersect is below the fovea, this is not the most light and color sensitive. The visual axis is connected from the center of the pupil to the fovea. This axis gives the best color vision, but, because it doesn't intersect the cornea and lens at their exact centers, yet is not as optically clear as light passing on the optical axis. The offset of both these axes in 3D object space is shown by the three angles α, κ and λ.

Summary: Visual axis and optical axis have an offset which is patient specific. Relevance of this angular offset in model based gaze estimation is explained in the later section as we

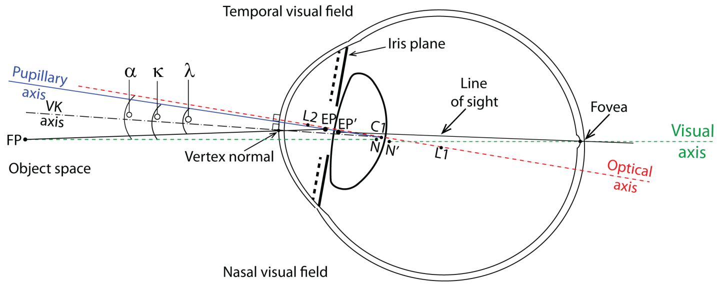

Figure 2-3: Schematic sketch of ocular axes [21]. The axes are indicated by the following lines; solid black (line of sight), solid blue (pupillary axis), dashed green (visual axis), dashed red (optical axis), and dashed black (videokerato-scope axis). EP entrance pupil,EP' exit pupil center, FP Fixation Point are shown. The angle from the PA to the line of sight is λ, optical axis to the visual axis is α, pupillary axis and visual axis is κ

are able to detect optical axis accurately which works accurately as we only need to record the eye ball rotation and not the absolute gaze itself. To record the patient's visual axis, we additionally need to be calibrated with patient specific parameters. We apply the corneal reflection method along with the explained Gullstrand-LeGrands schematic eye for our eye tracking device.

Chapter 3

Gaze Estimation

Before starting to explain the main built of the prototype, first it is necessary to introduce the theory behind the gaze estimation. We will follow the reasoning from the paper of E.D Guestrin [22]. We being our chapter with the theory behind estimating a real camera (with complex lens structure and sensor array) into a simple pin-hole model. Later using this camera model and our optical eye model to find the equations governing the point of gaze.

3-1 Camera Calibration

The procedure to calibrate the camera is termed as the calibration procedure in this thesis (*Not be confused with the patient parameter calibration done by patient looking at target points*). We estimate the real world camera to a pin hole model and finally estimate the parameters in relation to a stereo camera setup with determining the translation and rotation of each camera with respect to each other.

Linear pinhole camera model approximation: Imagining systems vary from complexity based on the manufacturer and requirements. But for every camera, as more light goes through the aperture, the better the sensor can work with it's fixed quantum efficiency. To model these higher order systems, we approximate the complex system into a first order liner pinhole model with only the consideration of Gaussian optics.

3-1-1 Camera Model Geometry

The pinhole camera model defines the geometric relationship between a 3D point and its 2D corresponding projection onto the image plane. When using a pinhole camera model, this geometric mapping from 3D to 2D is called a perspective projection. The geometry related to the mapping of a pinhole camera is illustrated in Figure 3-1. Nodal point(camera aperture) of the camera is located at $\mathbf{o}$. Also shown is the 3D orthogonal coordinate system with its origin at the nodal point with the three axes $(X1, X2, X3)$. With $X3$ (*optical axis*, *principal axis*, or *principal ray*) in the viewing direction. Image plane is parallel to axes $X1$ and $X2$ and is located at distance f or focal length from nodal point **o** in negative $X3$. The focal length is measured in pixels. We split the focal length in each direction into f_x, f_y. For an ideal camera $f_x = f_y$, but in practice they vary for the reasons:

- Manufacturing flaws of the sensor
- Non uniform scaling while processing the sensor output
- Changes cause by any lens distortion
- Errors in camera calibration

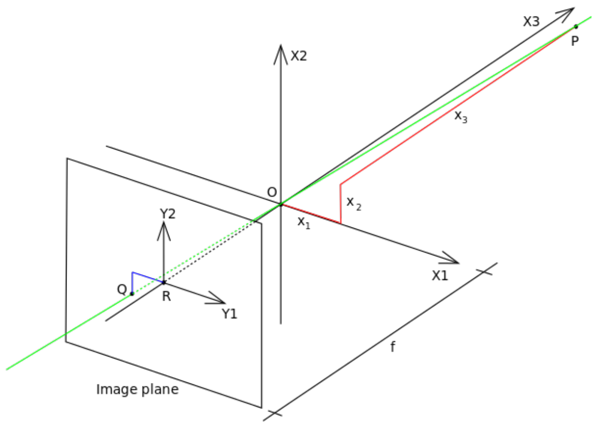

Figure 3-1: Projection for an ideal linear pinhole camera model. A point P is on the model's field of view at $(x_1, x_2, x_3)^T$ from **o**. The green line (projection line) shows its pin hole projection on the image plane marked with point Q on image plane. Let us have the coordinates of Q on the image plane coordinate $Y1, Y2$ as $Q = (u, v)^T$. Image from [23]

3-1-2 Camera Parameters

Perspective Projection or Camera Homography

A homography is an isomorphism of projective spaces as explained in detail in the Appendix C. For a pin-hole model, any point in 3D coordinates $(x_1, x_2, x_3)^T$ can be projected onto an image plane with representation as a 2D point seen in Figure 3-1.

Using the homogeneous coordinates, projection of a 3D point$(x_1, x_2, x_3)^T$ onto the image plane $(u, v)^T$ can be illustrated as:

$$u = \frac{x_1}{x_3} f_x \text{ and } v = \frac{x_2}{x_3} f_y, \tag{3-1}$$

where, f_x and f_y represent the focal length in pixels in each direction. This relation can be reformulated using the projective geometry framework, as

$$(\beta u, \beta v, \beta)^T = (x_1 f_x, x_2 f_y, x_3)^T, \tag{3-2}$$

where, $\beta = \frac{1}{\lambda} = x_3$ is the scalar homogeneous scaling factor. To show this in the matrix notation we get Eq. (3-3).

$$\beta \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix}. \tag{3-3}$$

As the focal length is measured in number of pixels. We introduced the physical horizontal and vertical spacing of the sensor elements as p_x, p_y, we get,

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} \frac{f_x}{p_x} & 0 & 0 & 0 \\ 0 & \frac{f_y}{p_y} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix}. \tag{3-4}$$

This 12 element matrix represents the main structure of the camera matrix. Camera matrix fundamentally maps between elements of two different projective spaces, it too can be regarded as a projective element. It has only 11 degrees of freedom since any multiplication by a non-zero scalar results in an equivalent camera matrix. We now bring in other parameters that define the camera.

Principal Point Offset

In the ideal model the principal point would be the center of the image plane. However, in most cameras a slight offset occurs. In the image pixel coordinate system, we define the principal point as $(c_x, c_y)^T$. This makes the projection matrix into:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} \frac{f_x}{p_x} & 0 & c_x & 0 \\ 0 & \frac{f_y}{p_y} & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix}. \tag{3-5}$$

Axis Skew

We implicitly assumed that the pixels of the image sensor have no skewing in pixels. However, this assumptions may not always be true in some cases. To account for shear distortion in the projected image, we introduce the skew coefficient s. Updating the camera matrix as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} \frac{f_x}{p_x} & s & c_x & 0 \\ 0 & \frac{f_y}{p_y} & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix}. \tag{3-6}$$

Conclusion: The parameter $\frac{1}{\beta} = \lambda$ must be adjusted such that the third element in the left-hand side of equation is unity. This procedure reflects the loss of information that takes place in imaging: an infinity of 3D points are imaged on a single 2D point. We finally yield:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} \frac{f_x}{p_x} & s & c_x \\ 0 & \frac{f_y}{p_y} & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \tag{3-7}$$

represented as,

$$\vec{u} = \lambda K \vec{x}. \tag{3-8}$$

where K is the intrinsic calibration matrix. K is a property of a camera. In general, the sensor size of a camera is supplied by the manufacturer, but the focal length to pixel dimensions ratios, $\frac{f_x}{p_x}$ and $\frac{f_y}{p_y}$, along with the principal point $(c_x, c_y)^T$ must be computed through a calibration procedure.

3-1-3 Calibration Procedure

There are several approaches for the calibration procedure. We follow one of the most robust and well cited procedure by Zhang [24]. We summarize this method in this following section:

Homography between the calibration plane and it's image: With the assumption of a world coordinate system with the origin on the calibration plane. With Z=0, without any loss of generality. From the homography equation we have,

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \tag{3-9}$$

For this derivation, the 3D point is defined with $[X, Y]^T$ since Z is always 0. Therefore, the homography H a 3x3 matrix can be defined upto a scale factor:

$$s\tilde{m} = H\tilde{M} \quad \text{with} \quad H = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}. \tag{3-10}$$

Constraints on the intrinsic parameters: With the known calibration plane and it's image we, an homography can be estimated. The Eq. (3-10) can be written as:

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \lambda A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}, \tag{3-11}$$

where λ is an arbitrary scalar. The vectors r_1 and r_2 in the rotation matrix are orthonormal, hence, we have:

$$h_1^T A^{-T} A^{-1} h_2 = 0, \tag{3-12}$$

$$h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2. \tag{3-13}$$

Given one homography with 8DOF of which 6 are extrinsic parameters, these are the two constraints on the intrinsic parameters.

Analytic Solution: Introducing a matrix B such that,

$$B = A^{-T}A_{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{c}{\alpha^2\beta} & \frac{cv_0-u_0\beta}{\alpha^2\beta} \\ -\frac{c}{\alpha^2\beta} & \frac{c^2}{\alpha^2\beta^2}+\frac{1}{\beta^2} & -\frac{c(cv_0-u_0\beta)}{\alpha^2\beta^2}-\frac{v_0}{\beta^2} \\ \frac{cv_0-u_0\beta}{\alpha^2\beta} & -\frac{c(cv_0-u_0\beta)}{\alpha^2\beta^2}-\frac{v_0}{\beta^2} & \frac{(cv_0-u_0\beta)^2}{\alpha^2\beta^2}+\frac{v_0^2}{\beta^2}+1 \end{bmatrix}. \tag{3-14}$$

As the matrix is symmetrical, it can be defined by a 6D vector

$$b = \begin{bmatrix} B_{11} & B_{12} & B_{22} & B_{13} & B_{23} & B_{33} \end{bmatrix}^T, \tag{3-15}$$

with the i^{th} column vector of H be h_i, we have:

$$h_i^T B h_j = v_{ij}^T b, \tag{3-16}$$

with 6D vector

$$v_{ij} = \begin{bmatrix} h_{i1}h_{j1} & h_{i1}h_{j2}+h_{i2}h_{j1} & h_{i2}h_{j2} & h_{i3}h_{j1}+h_{i1}h_{j3} & h_{i3}h_{j2}+h_{i2}h_{j3} & h_{i3}h_{j3} \end{bmatrix}^T.$$

The two equations Eq. (3-12) and Eq. (3-13) can be written as two homogeneous equations:

$$\begin{bmatrix} v_{12}^T \\ (v_{11}-v_{22})^T \end{bmatrix} b = 0, \tag{3-17}$$

if the number of images taken are n then stacking all equations we get:

$$Vb = 0, \tag{3-18}$$

with V as a $2n\times$ 6 matrix. We have a solution for b if $n \geq 3$, defined upto a scale factor. If n = 2, we can introduce another constraint by taking $c = 0$ ($[0,1,0,0,0,0]\,b = 0$). The solution is then known as V^TV associated with the smallest singular value.With the value of b, the matrix A can be computed.

With known value of A, we have,

$$r_1 = \lambda A^{-1}h_1, r_2 = \lambda A^{-1}h_2, r_3 = r_1 \times r_2, t = \lambda A^{-1}h_3,$$

with $\lambda = 1/\left\|A^{-1}h_1\right\| = 1/\left\|A^{-1}h_2\right\|$. Due to the presence of noise, the rotation matrix is an approximation and does not satisfy the properties of a rotation matrix.

Maximum Likelihood Estimation: As the above solution optimizes an algebraic distance which is not physically meaningful. This is refined with the maximum likelihood estimation. With n images and m points in the plane, the maximum likelihood estimate can be obtained by minimizing the following functional:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \left\|m_{ij} - \hat{m}(A, R_i, t_i, M_j)\right\|^2, \tag{3-19}$$

where $m_{ij} - \hat{m}(A, R_i, t_i, M_j)$ is the projection of the j^{th} 3D point on the i^{th} image. R is parametrized by a vector of 3 parameters denoted by r. R and r are related by the Rodrigues formula [25]. Minimizing Eq. (3-19) is a nonlinear minimization problem solved with Levenberg-Marquardt using the initial guess of A from the above solution.

Lens Distortion Estimation: We follow the step from [26]. The model behind the estimation is shown as:

$$x' = x \cdot (1 + k_1 * r^2 + k_2 * r^4), \tag{3-20}$$

$$y' = y \cdot (1 + k_1 * r^2 + k_2 * r^4), \tag{3-21}$$

where, x,y are the un-distorted pixel coordinates in normalized image coordinates and x',y' are it's respective distorted coordinates. k_1, k_2 are the radial distortion coefficients of the lens and $r^2 = x^2 + y^2$. Whereas, for the tangential distortion we have the model:

$$x' = x + \left[2p_1xy + p_2(r^2 + 2x^2)\right], \tag{3-22}$$

$$y' = y + \left[p_1 * (r^2 + 2y^2) + 2p_2xy\right], \tag{3-23}$$

with p_1 and p_2 as the tangential distortion coefficients.

The coefficients k_1, k_2, p_1 and p_2 are estimated to account for the lens distortion, images are un-distorted with this model before processing on them.

3-1-4 Stereo Camera Calibration

The base of calibration is the same calibration procedure as we show above for each individual camera. Further on, we calibrate the stereo setup with the steps taken as follows:

1. Images of the calibration target are taken simultaneously by both the cameras.
2. The single camera calibration routine is performed. independently for both the cameras for each point that is detected.
3. Common points used in both the camera calibration procedures are separated.
4. Initial estimate is made for the Extrinsic matrix of the second camera using the median of extrinsic used in the calibration.
5. The estimates are refined by using the non-linear least squares minimization.

3-2 Gaze Estimation - Mathematical Model

In this section, we present the general and our specific model for the remote Point of Gaze (PoG) estimation which utilizes number of corneal reflections or glints extracted from video images from one or more video cameras. The model covers general configurations as seen in the Figure 3-2.

Representation : All points are represented as three dimensional column vectors (bold font) in a right-handed World Coordinate System (WCS). i, j represent the number light sources and cameras, respectively. $\mathbf{l}_i$ represents the i^{th} light source, $\mathbf{q}_{ij}$ is the reflection of the i^{th} light source as seen on j^{th} camera. $\mathbf{o}_j$ is the nodal point of the j^{th} camera, $\mathbf{u}_{ij}$ is the image of the i^{th} glint (first Purkinje image) on j^{th} camera image sensor plane, $\mathbf{v}_j$ is the image of the pupil center on j^{th} camera image sensor plane, $\mathbf{p}$ is the pupil center, $\mathbf{r}_j$ point of refraction of pupil center ray as seen by j^{th} camera, $\mathbf{c}$ is the corneal curvature, $\mathbf{d}$ is the center of rotation of the eye.

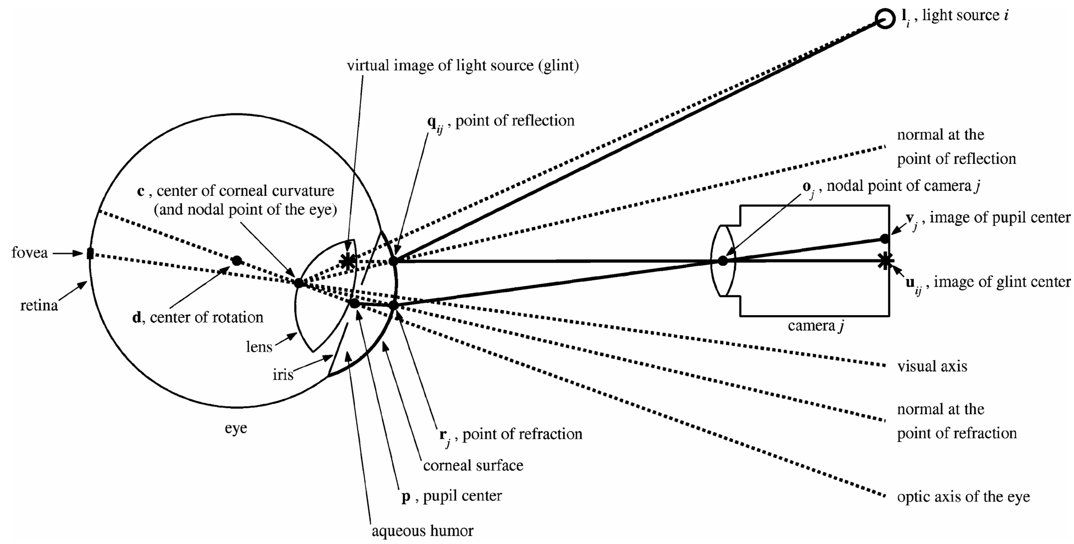

Figure 3-2: Optical Ray-tracing diagram(not to scale) representing the eye, camera and light source. We see they light ray from the light source reflects off the cornea and the image of the pupil originating from pupil center is refracted off the cornea surface and is detected by the camera sensor.This figure is taken from the scheme in [22]

Assumption: Light sources behave as point sources and the video cameras are modeled as pinhole cameras.

Model Equations: Two main features in this system that are seen or recorded with the video camera are the pupil center and the glint. These are explained as follows:

1. A light ray coming from the light source reflects on the surface of the cornea reflecting and passing through the nodal point of the camera and into the sensor plane.
2. Visual ray originating from the pupil center refracts while exiting the surface of the cornea passing through the nodal point of the camera and into the sensor plane.

For a ray from light source $\mathbf{l}_i$ with reflection at a point $\mathbf{q}_{ij}$ on the corneal surface passing through the nodal point of camera $j, \mathbf{o}_j$ and intersecting the camera image plane at a point $\mathbf{u}_{ij}$, can be expressed in parametric form as Eq. (3-24):

$$\mathbf{q}_{ij} = \mathbf{o}_j + k_{q,ij}(\mathbf{o}_j - \mathbf{u}_{ij}), \text{ for some scalar } k_{q,ij}, \tag{3-24}$$

for the eye model we use discussed in Chapter B, the corneal surface is modeled as a convex spherical mirror of radius R, the condition for $\mathbf{q}_{ij}$ can be shown in Eq. (3-25)

$$\left\|\mathbf{q}_{ij} - \mathbf{c}\right\| = R. \tag{3-25}$$

Based on the fundamental law of reflection, incident ray, reflected ray and surface normal at the point of reflection are co-planar implying that $\mathbf{l}_i, \mathbf{q}_{ij}, \mathbf{c}$ and $\mathbf{o}_j$ are co-planar. As

$\mathbf{a}_1 \times \mathbf{a}_2 \bullet \mathbf{a}_3 = 0$ is true for any three co-planar vectors $\mathbf{a}_1, \mathbf{a}_2$ and $\mathbf{a}_3$. We can write it as Eq. (3-26)

$$(\mathbf{l}_i - \mathbf{o}_j) \times (\mathbf{q}_{ij} - \mathbf{o}_j) \bullet (\mathbf{c} - \mathbf{o}_j) = 0. \tag{3-26}$$

Law of reflection also states that the angles of incidence and reflection are equal, the angle between two vectors can be obtained from scalar product, $\mathbf{a} \bullet \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos\theta$ for any two vectors $\mathbf{a}$ and $\mathbf{b}$ This condition is expressed in Eq. (3-27)

$$(\mathbf{l}_i - \mathbf{q}_{ij}) \bullet (\mathbf{q}_{ij} - \mathbf{c}) \cdot \left\|\mathbf{o}_j - \mathbf{q}_{ij}\right\| = (\mathbf{o}_j - \mathbf{q}_{ij}) \bullet (\mathbf{q}_{ij} - \mathbf{c}) \cdot \left\|\mathbf{l}_i - \mathbf{q}_{ij}\right\|. \tag{3-27}$$

Next, the condition for the ray coming from pupil center $\mathbf{p}$ to refract on corneal surface$\mathbf{r}_j$ continue to pass through the nodal point of camera $\mathbf{o}_j$ intersecting the image sensor plane at $\mathbf{v}_j$ can be expressed as Eq. (3-28)

$$\mathbf{r}_j = \mathbf{o}_j + k_{r,j}(\mathbf{o}_j - \mathbf{v}_j) \text{ for some } k_{r,j}. \tag{3-28}$$

Similar to Eq. (3-25), the refraction point on corneal surface can be written as Eq. (3-29)

$$\|\mathbf{r}_j - \mathbf{c}\| = R. \tag{3-29}$$

With the fundamental law of refraction, we know incident ray, reflected ray and normal at the point of refraction are co-planar. With the vector $(\mathbf{r}_j - \mathbf{c})$ normal to surface at $\mathbf{r}_j$, we imply $\mathbf{p}, \mathbf{r}_j, \mathbf{c}$ and $\mathbf{o}_j$ are co-planar, written as Eq. (3-30)

$$(\mathbf{r}_j - \mathbf{o}_j) \times (\mathbf{c} - \mathbf{o}_j) \bullet (\mathbf{p} - \mathbf{o}_j) = 0. \tag{3-30}$$

Using Snell's law and the angle between the two vectors from cross product we can express (3-31)

$$n_1 \cdot \|(\mathbf{r}_j - \mathbf{c}) \times (\mathbf{p} - \mathbf{r}_j)\| \cdot \|\mathbf{o}_j - \mathbf{r}_j\| = n_2 \cdot \|(\mathbf{r}_j - \mathbf{c}) \times (\mathbf{o}_j - \mathbf{r}_j)\| \cdot \|\mathbf{p} - \mathbf{r}_j\|, \tag{3-31}$$

where, n_1 represents the effective index of refraction of the aqueous humor and cornea combined and n_2 is the index of refraction of air approximated to 1. The second refraction at aqueous humorcornea interface is neglected since the difference is minimal to cornea air interface [22]. But this may also be estimated or averaged for patient specific eye.

Considering K to be the distance between the pupil center and the center of corneal curvature we get Eq. (3-32) :

$$\|\mathbf{p} - \mathbf{c}\| = K. \tag{3-32}$$

Summarize: The set of equations from Eq. (3-24) to Eq. (3-32) explain the eye model. These are re-wriiten for clear picture:

$$\mathbf{q}_{ij} = \mathbf{o}_j + k_{q,ij}(\mathbf{o}_j - \mathbf{u}_{ij}), \text{ for some scalar } k_{q,ij},$$

$$\left\|\mathbf{q}_{ij} - \mathbf{c}\right\| = R,$$

$$(\mathbf{l}_i - \mathbf{o}_j) \times (\mathbf{q}_{ij} - \mathbf{o}_j) \bullet (\mathbf{c} - \mathbf{o}_j) = 0,$$

$$(\mathbf{l}_i - \mathbf{q}_{ij}) \bullet (\mathbf{q}_{ij} - \mathbf{c}) \cdot \left\|\mathbf{o}_j - \mathbf{q}_{ij}\right\| = (\mathbf{o}_j - \mathbf{q}_{ij}) \bullet (\mathbf{q}_{ij} - \mathbf{c}) \cdot \left\|\mathbf{l}_i - \mathbf{q}_{ij}\right\|,$$

$$\mathbf{r}_j = \mathbf{o}_j + k_{r,j}(\mathbf{o}_j - \mathbf{v}_j) \text{ for some } k_{r,j},$$

$$\|\mathbf{r}_j - \mathbf{c}\| = R,$$

$$(\mathbf{r}_j - \mathbf{o}_j) \times (\mathbf{c} - \mathbf{o}_j) \bullet (\mathbf{p} - \mathbf{o}_j) = 0,$$

$$n_1 \cdot \|(\mathbf{r}_j - \mathbf{c}) \times (\mathbf{p} - \mathbf{r}_j)\| \cdot \|\mathbf{o}_j - \mathbf{r}_j\| = n_2 \cdot \|(\mathbf{r}_j - \mathbf{c}) \times (\mathbf{o}_j - \mathbf{r}_j)\| \cdot \|\mathbf{p} - \mathbf{r}_j\|,$$

$$\|\mathbf{p} - \mathbf{c}\| = K.$$

The above set of equations govern p and c, with which we can reconstruct the optic axis. Further, we tackle two different system configuration for the reconstruction:

3-2-1 Single Camera Configuration

Single camera with a single light source is the simplest system configuration. In this case, we need to know the subject specific eye parameters such as, R, K, and n_1. Using the system of equations Eq. (3-24) to Eq. (3-32) with $i = 1$ and $j = 1$, is equivalent to 13 scalar equations with 14 scalar unknowns. Which would require one additional constraint to be able to solve, we introduce Eq. (3-33):

$$\|\mathbf{o}_1 - \mathbf{c}\| = known. \tag{3-33}$$

This constraint can be satisfied if the head is fixed relative to the system or if the distance between the eye and the camera is estimated (with, magnetic head tracker, ultrasonic transducer, auto-focus system, etc.)

Whereas, in case of multiple light sources, the system of equations Eq. (3-24) to Eq. (3-32) with $i = 1, \cdots N$ and $j = 1$, if the eye parameters (R, K and n_1) are obtained through a calibration procedure. In this case, it is advantageous to substitute Eq. (3-24) into Eq. (3-26) to obtain Eq. (3-34)

$$\underbrace{(\mathbf{l}_i - \mathbf{o}_j) \times (\mathbf{u}_{ij} - \mathbf{o}_j)}_{\text{normal to the plane defined by } \mathbf{l}_i, \mathbf{o}_j, \mathbf{u}_{ij}} \bullet (\mathbf{c} - \mathbf{o}_j) = 0. \tag{3-34}$$

Since $j = 1$, the subscript is dropped and, by noting that $\mathbf{a} \bullet \mathbf{b} = \mathbf{a}^T\mathbf{b}$ for Eq. (3-34) , we get (3-35):

$$\underbrace{\begin{bmatrix} [(\mathbf{l}_1 - \mathbf{o}) \times (\mathbf{u}_1 - \mathbf{o})]^T \\ [(\mathbf{l}_2 - \mathbf{o}) \times (\mathbf{u}_2 - \mathbf{o})]^T \\ \vdots \\ [(\mathbf{l}_N - \mathbf{o}) \times (\mathbf{u}_N - \mathbf{o})]^T \end{bmatrix}}_{\mathbf{M}} (\mathbf{c} - \mathbf{o}) = 0. \tag{3-35}$$

Eq. (3-34) explains the $\mathbf{c}$ belonging to each plane defined by the $\mathbf{o}_j$, $\mathbf{l}_i$, and $\mathbf{u}_{ij}$. This interpretation follows that matrix $\mathbf{M}$ has, at most, rank 2. If $\mathbf{M}$ has rank 2, the solution to Eq. (3-35) is given by an equation of the form

$$\mathbf{c} - \mathbf{o} = k_{c,b}\mathbf{b}_{norm}, \tag{3-36}$$

which defines the vector $(\mathbf{c} - \mathbf{o})$ up to a scale factor. From this reasoning, it follows that Eq. (3-24), Eq. (3-25), Eq. (3-27), $i = 1, \cdots N, j = 1$ and Eq. (3-36) are equivalent to $(5N+3)$ scalar equations with $(4N + 4)$ scalar unknowns.

In particular, when $N = 2, \mathbf{b}_{norm}$ is a unit vector in the direction of the line of intersection of the planes whose normals are given by $[(\mathbf{l}_1 - \mathbf{o}) \times (\mathbf{u}_1 - \mathbf{o})]$ and $[(\mathbf{l}_2 - \mathbf{o}) \times (\mathbf{u}_2 - \mathbf{o})]$ thus with $\mathbf{b}_{norm} = \frac{\mathbf{b}}{||\mathbf{b}||}$. We have (3-37)

$$\mathbf{b} = [(\mathbf{l}_1 - \mathbf{o}) \times (\mathbf{u}_1 - \mathbf{o})] \times [(\mathbf{l}_2 - \mathbf{o}) \times (\mathbf{u}_2 - \mathbf{o})], \tag{3-37}$$

and Eq. (3-24), Eq. (3-25), Eq. (3-27), $i = 1, 2$ and are equivalent to 13 scalar equations with 12 scalar unknowns.

With the multiple light source we have enough equations to solve for the corneal center $\mathbf{c}$. Along with Eq. (3-28) and Eq. (3-29) we can compute the point of refraction $\mathbf{r} = \mathbf{r}_1$ (4 scalar unknowns and 4 scalar equations). Knowing $\mathbf{c}$ and $\mathbf{r}$, 3-30 to 3-32 we can solve for $\mathbf{p}$ (3 scalar unknowns and 3 scalar equations).

With $\mathbf{c}$ and $\mathbf{p}$, the optic axis of the eye can be reconstructed as the line defined by these two points.

Conclusion: In this configuration the subject specific eye parameters R, K and n_1 must be known in order to reconstruct the optic axis of the eye and thus be able to estimate the PoG. The above discussion shows that one camera and two light sources is the simplest configuration that allows for the reconstruction of the optic axis of the eye from the centers of pupil and glints while allowing for free head movements. The above analysis also shows that knowing (the center of corneal curvature), the calculation of (the pupil center) is independent of the number of light sources (7 scalar equations and 7 scalar unknowns regardless of the number of light sources). In the next section, we tackle system configuration that allows for the reconstruction of the optic axis without the need for a subject-specific parameter calibration.

3-2-2 Stereo Camera Configuration

Using system configuration with two cameras and multiple light sources, Eq. (3-34), $i = 1, ..., N$ and $j = 1, 2$, together can be written in matrix form as (3-38)

$$\underbrace{\begin{bmatrix} [(\mathbf{l}_1 - \mathbf{o}_1) \times (\mathbf{u}_{11} - \mathbf{o}_1)]^T \\ [(\mathbf{l}_1 - \mathbf{o}_2) \times (\mathbf{u}_{12} - \mathbf{o}_2)]^T \\ \vdots \\ [(\mathbf{l}_N - \mathbf{o}_1) \times (\mathbf{u}_{N1} - \mathbf{o}_1)]^T \\ [(\mathbf{l}_N - \mathbf{o}_2) \times (\mathbf{u}_{N2} - \mathbf{o}_2)]^T \end{bmatrix}}_{\mathbf{M}_2} \mathbf{c} = \underbrace{\begin{bmatrix} (\mathbf{l}_1 - \mathbf{o}_1) \times (\mathbf{u}_{11} - \mathbf{o}_1) \bullet \mathbf{o}_1 \\ (\mathbf{l}_1 - \mathbf{o}_2) \times (\mathbf{u}_{12} - \mathbf{o}_2) \bullet \mathbf{o}_2 \\ \vdots \\ (\mathbf{l}_N - \mathbf{o}_1) \times (\mathbf{u}_{N1} - \mathbf{o}_1 \bullet \mathbf{o}_1) \\ (\mathbf{l}_N - \mathbf{o}_2) \times (\mathbf{u}_{N2} - \mathbf{o}_2 \bullet \mathbf{o}_2) \end{bmatrix}}_{\mathbf{h}}, \tag{3-38}$$

after applying the distributive property for the dot product, rearranging terms and noting that $\mathbf{a} \bullet \mathbf{b} = \mathbf{a}^T \mathbf{b}$. If $\mathbf{M}_2$ has rank 3, $\mathbf{c}$ can be obtained from Eq. (3-38) by using the left pseudo-inverse of $\mathbf{M}_2$ as

$$\mathbf{c} = (\mathbf{M}_2^T \mathbf{M}_2)^{-1} \mathbf{M}_2^T \mathbf{h} \tag{3-39}$$

If only 3 linearly independent rows of $\mathbf{M}_2$ and the corresponding rows of $\mathbf{h}$ are considered in Eq. (3-38), then Eq. (3-39) reduces to $\mathbf{c} = (\mathbf{M}_2)^{-1}\mathbf{h}$

Note that when Eq. (3-28) and Eq. (3-30) are combined, they correspond to the physical condition that for each camera $\mathbf{p}$, $\mathbf{r}_j$, $\mathbf{o}_j$, $\mathbf{v}_j$, and $\mathbf{c}$ are co-planar. For this system configuration with two cameras, it is convenient to represent this physical condition as

$$\begin{aligned} (\mathbf{o}_1 - \mathbf{v}_1) \times (\mathbf{c} - \mathbf{o}_1) \bullet (\mathbf{p} - \mathbf{c}) = 0 \\ (\mathbf{o}_2 - \mathbf{v}_2) \times (\mathbf{c} - \mathbf{o}_2) \bullet (\mathbf{p} - \mathbf{c}) = 0 \end{aligned}. \tag{3-40}$$

Since the optic axis of the eye is defined by points $\mathbf{c}$ and $\mathbf{p}$, these equations mean that the optic axis of the eye belongs to the plane defined by points $\mathbf{c}, \mathbf{o}_1$ and $\mathbf{v}_1$. Therefore, the optic axis of the eye is the line of the intersection of those two planes and its direction is given by

$$s = [(\mathbf{o}_1 - \mathbf{v}_1) \times (\mathbf{c} - \mathbf{o}_1)] \times [(\mathbf{o}_2 - \mathbf{v}_2) \times (\mathbf{c} - \mathbf{o}_2)], \tag{3-41}$$

where $\mathbf{s}_{norm} = \frac{\mathbf{s}}{\|\mathbf{s}\|}$.
if $\mathbf{s} \neq 0$, the solution to Eq. (3-40) can be expressed as

$$\mathbf{p} - \mathbf{c} = k_{pc} \mathbf{s}_{norm}, \tag{3-42}$$

which defines vector $(\mathbf{p} - \mathbf{c})$ up to a scale factor (notice that $|k_{pc}| = K$, the distance between the pupil center and the center of corneal curvature). In this way, knowing $\mathbf{c}$ and the direction of vector $\mathbf{p} - \mathbf{c}$, i.e., the direction of vector$\mathbf{s}$, the optic axis of the eye can be reconstructed without knowing the eye parameters.

Special Condition: Equation Eq. (3-42) is only valid when $\mathbf{s} \neq 0$. The following discussion considers the conditions that result in $\mathbf{s} = 0$ .To have $\mathbf{s} = 0$, it is sufficient that$(\mathbf{o}_1 - \mathbf{v}_1) \times (\mathbf{c} - \mathbf{o}_1) = 0$ or $(\mathbf{o}_2 - \mathbf{v}_2) \times (\mathbf{c} - \mathbf{o}_2) = 0$, or that $(\mathbf{o}_1 - \mathbf{v}_1) \times (\mathbf{c} - \mathbf{o}_1)$ and$(\mathbf{o}_2 - \mathbf{v}_2) \times (\mathbf{c} - \mathbf{o}_2)$ are parallel. The condition that$(\mathbf{o}_j - \mathbf{v}_j) \times (\mathbf{c} - \mathbf{o}_j) = 0$ implies that the center of corneal curvature, the nodal point of the camera, and the image of the pupil center are collinear. Since any line passing through point $\mathbf{c}$ is normal to the spherical corneal surface, the line defined by points $\mathbf{o}_j$ and $\mathbf{v}_j$ is normal to the corneal surface. Since points $\mathbf{o}_j$ and $\mathbf{v}_j$ are on the refracted ray coming from the pupil center, the refracted ray is normal to the corneal surface (i.e., there is no refraction) and, therefore, point $\mathbf{p}$ is collinear with points$\mathbf{c}$ and $\mathbf{p}$. Since the optic axis of the eye is defined by points $\mathbf{c}$ and $\mathbf{p}$, it implies that the optic axis goes through the nodal point of the camera, . In summary, if the optic axis of the eye passes through the nodal point of camera, then $(\mathbf{o}_j - \mathbf{v}_j) \times (\mathbf{c} - \mathbf{o}_j) = 0$ and hence $\mathbf{s} = 0$.

The condition that $(\mathbf{o}_1 - \mathbf{v}_1) \times (\mathbf{c} - \mathbf{o}_1)$ and$(\mathbf{o}_2 - \mathbf{v}_2) \times (\mathbf{c} - \mathbf{o}_2)$ are parallel implies that points $\mathbf{c}, \mathbf{o}_1, \mathbf{v}_1, \mathbf{o}_2, \mathbf{v}_2$ are all on a single plane. Equation 3-40 implies that the optic axis of the eye, defined by the center of corneal curvature and the pupil center is also in that plane. Consequently, this situation occurs when the optic axis of the eye is co-planar with the line defined by the nodal points of the cameras, $\mathbf{o}_1$ and $\mathbf{o}_2$. Since the condition that the optic axis of the eye passes through the nodal point of a camera is a particular case of the condition that the optic axis of the eye is co-planar with the line connecting the nodal points of the cameras, and since in practice the condition that the optic axis of the eye is parallel to the line connecting the nodal points of the cameras is unrealistic, the above discussion can be summarized by saying that $\mathbf{s} \neq 0$ and, hence, 3-42 is valid as long as the optic axis of the eye does not intersect the line defined by the nodal points of the cameras.

Conclusion: From the discussion in this section, it follows that the simplest configuration that allows for the reconstruction of the optic axis of the eye, without knowledge of the values of the subject-specific eye parameters, consists of two cameras and two light sources. In order to reconstruct the visual axis of the eye from the optical axis, to be able to estimate the actual PoG, the eye parameters α_{eye}, β_{eye} (angle between visual axis and optical axis) still need to be estimated. These two parameters, which are subject-specific, can be estimated through a simple calibration procedure in which the subject is required to fixate on a single point. As in this thesis we do not tackle that the derivation for this can be found within [22].

Chapter 4

Proposed System Design

In this chapter, we explain our process towards building an experimental prototype setup for the gaze estimation using corneal reflections as theorized in Chapter 3. The decision factors towards building this setup were high accuracy and system flexibility, which contribute towards use in proton therapy clinics. We divide this into multiple sections which are explained as follows:

1. Hardware configuration used for the development and their geometric placement in the setup.

2. Discussion and performance of the algorithms used for gaze estimation procedure.

3. Applying other detection algorithms and how they compare in terms of time and performance.

4. Using vision algorithms that allow us to detect saccades and blink in real time.

5. Construct a Graphical User Interface (GUI) for ease of use and to package the software.

6. Summarizing with the overall structure of the program with our software flowchart.

4-1 Hardware Setup

Our gaze tracking system is composed of pair of stereo cameras and two IR LEDs as shown in the Figure 4-1. The LEDs are used as the point source illumination for locating the cornea center. The location of LEDs and the baseline of the stereo camera used in the system are not fixed but can be varied. If the distances are varied, the calibration of the setup must be ensured before the gaze tracking. The exact parts used are listed in the Table **??**.

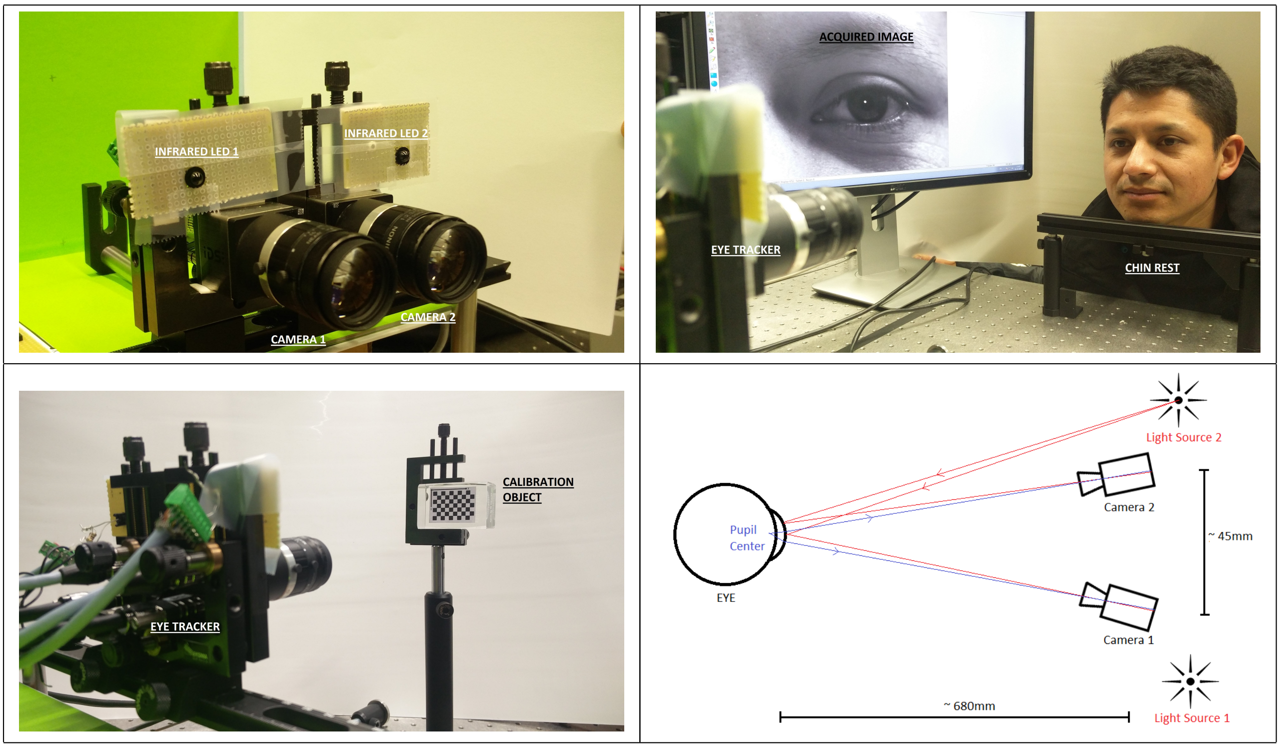

Figure 4-1: *Top Left:* Eye Tracking setup consisting of the two cameras and two infrared LED placed on a rail to adjust baseline, *Top Right:* Measurement of a subject during where the subject places his chin on the head rest, *Bottom Left:* Calibration procedure by showing the calibration object with different orientations to the setup, *Bottom Right:* Ray diagram of the top view of our setup geometry (when one light source is used)

Assembly Parts			Comments
Part Name	***Part Number***	***Quantity***	
Monochrome Camera	UI-3140CP-M-GL Rev.2	2	1280x1024 resolution with 224 fps(peak)
Infrared LED	SFH 4554	2	860nm, Half Angle:10degree
C-Mount Lens	Fujinon HF50HA-1B	2	50mm focal length with 1.5Megapixels
Flat Mirror	N.A	1	In-house stock, mounted on a pole
Calibration Object	N.A	1	Printed with squares of 3.5mm width
GPIO Cable	HR25	2	Modified HR25 connection for our need

Table 4-1: Hardware components used in the eye tracking setup and specifications

4-1-1 Light Source

Near-Infrared (NIR) Imaging: We can acquire the images with ambient lighting reflected by the eye (Passive) or with an infrared illumination (Active). Disadvantage of using ambient light is the noise(reflection) and varying ambient light which occludes the detection of the pupil as seen in Figure 4-2. Using the NIR Sources eliminates this problem and also as this is invisible to the eye, varying the intensity does not affect the pupil dilation nor distract the user from any gaze fixation. With this, we worked in the NIR range with the glint and illumination source as the same.

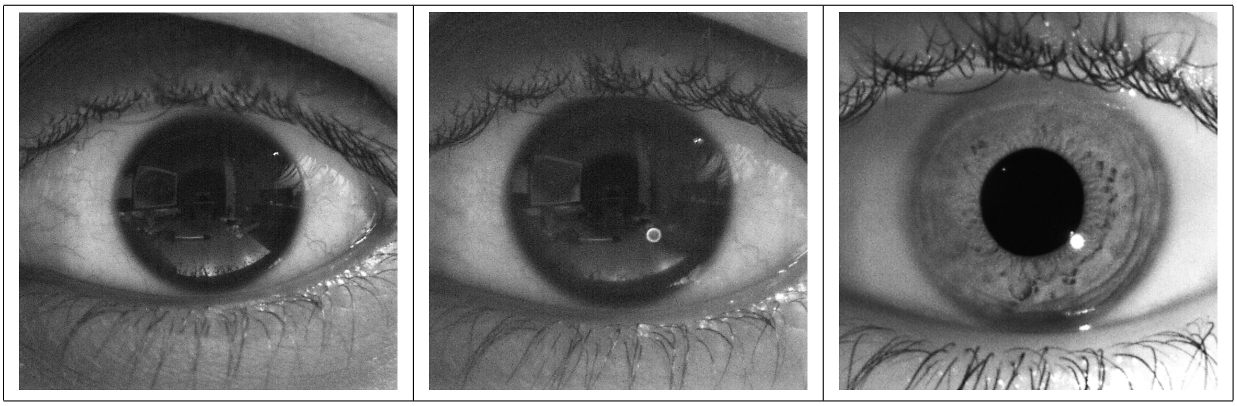

Figure 4-2: *Left:* In visible light, we see less iris pigment details with more reflections. *Center:* In visible light along with infrared light, we still see lot of reflection and focus is restricted to either infrared or visible light. *Right:* Eyes reveal rich iris texture in the NIR band, and most corneal reflections are blocked giving us sharp glint.

Infrared Hazard: The peak wavelength of our LED is 860nm and maximum power consumption is 180mw. Radiation within this wavelength can cause a thermal retina hazard and thermal injury risk of the cornea and possible delayed effects on the lens of the eye (cataractogenesis). The standard IEC (EN DIN) 60825-1 'SAFETY OF LASER PRODUCTS' is far more restrictive for any thermal retinal hazard and the limits to the cornea/lens are given in IEC 62471. Conditions can also be found only in IEC 62471. We studied the safety note from our LED manufacture [27], for conforming to the standards. The exact equations used are found in [27]. For our hardware arrangement, we have the specifications: 2× SFH4554, DC Operation I_{max}=1 A, $I_{e,max}$= 550mW/sr.

For a working distance of: $d = 0.2m$:

- Cornea Hazard: The values from the data sheet conform to the Emission Limt (EL) based on [27] as seen in Eq. (4-1-1).

$$E_{e,max} = \frac{I_e}{d^2} = \frac{0.55 \times 2}{0.2^2} = \mathbf{27.5W/m^2} < \mathbf{E_{IR}} = \mathbf{100W/m^2} \tag{4-1}$$

- Retinal Hazard: Calculation of angular subtenseα at a viewing distance d for a mean source extension Z:

$$\alpha = \frac{Z}{d} = \frac{0.3mm}{200mm} = 0.0015rad \tag{4-2}$$

where, $Z = \frac{(l+w)}{2} = \frac{0.3mm+0.3mm}{2} = 0.3mm$ according to the value in the standard. $\alpha_{\text{eff}} = 0.01$ for $(t > 1000s)$. Hence, the Emission Limt (EL) is calculated as $\mathbf{L_{IR}} = \frac{6000}{\alpha_{\text{eff}}}$. With, $R(\lambda = 860nm) = 0.40$, we calculate the actual $\mathbf{L_{IR}}$ value as shown in 4-1-1

$$\mathbf{L_{IR}} = I_e \cdot \frac{R(\lambda)}{(\frac{(l+w)}{2})^2} = 550 \cdot \frac{0.4}{(0.3)^2} = 2444.45\text{mW}/\text{mm}^2/\text{sr} \ll \mathbf{2.94 \cdot 10^5}\text{mW}/\text{mm}^2/\text{sr (EL)} \tag{4-3}$$

Limits of $\mathbf{L_{IR}}$ and $\mathbf{E_{IR}}$ are below the allowed limits. No risk.

Illumination Control: The digital cameras we use have General Purpose Input-Output (GPIO) pins as shown in the pin assignment for our camera in Figure 4-3. These GPIO pins are used to control the LED illumination. LED is connected to the GPIO pin in series with a resistor. We output a Pulse Width Modulation(PWM) signal which provides the ability to simulate varying levels of power by oscillating the output. Changing the duty cycle allows for chnage in illumination.

Figure 4-3: LED connection to the camera, with the output signal controlling illumination. Image taken from [28]

4-1-2 Camera

As explained above we work in NIR range with our LED at 860nm peak illumination. We try to optimize on the cost of the prototype hence, as seen in Figure 4-4 the quantum efficiency of the camera at this wavelength is only little below 20%. But as we are testing in a controlled environment, it is more that sufficient to test our algorithms.

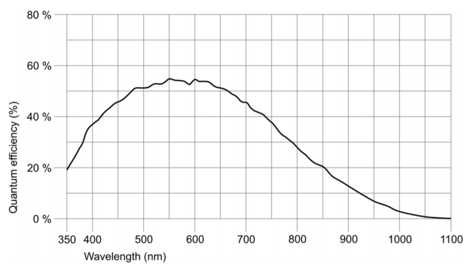

Figure 4-4: Quantum efficiency - UI3140CP

Lens: The camera is connected to the lens HF50HA-1B with C-mount. The focusing of the lens can be varied based on change in baseline of the cameras and required patient position.

Acquisition: The IDS camera (without modifications) can be connected only with a USB interface (2.0 or 3.0). With the highest bandwidth of 5 Gbit/s, i.e. images can be transmitted with a bandwidth of 400 Mbytes/s. In addition to the drivers, IDS provides us with the uEye API (The *uEye.h* header file contains all the definitions and constants needed for the uEye API) programming interface. As we use MATLAB to run our detection and estimation codes, we integrate MATLAB with the API, with the use of the uEye .NET framework.

We use the Direct3D interface as seen on the Appendix A-1-1 and the process flow is outlined is as follows:

1. Make uEye .NET assembly visible to MATLAB
2. Create an object for each of the camera using the CameraID to initialize the left and right cameras correctly.
3. Set the display mode to Direct3D and define the camera parameters
4. Allocate camera memory for copying data from RAM of graphics card to PC.
5. Direct3D acquisition is started and runs in the background independently of the MATLAB status.
6. For each loop, Image Frame is transferred from the graphics card to the PC and converted from .NET array to MATLAB array for further processing.

4-2 Software - Algorithm for Gaze Estimation

We will subdivide the software into three parts. In the first part, we would discuss the algorithm that can estimate the gaze in all conditions without looking into computational speed and focus on the estimation itself. In the second part, we discuss other algorithms and analyze how they improve performance or computational time. Third, we talk about the algorithms for the fast run for Saccade and Blink detection.

For this part, each algorithm is briefly analyzed and concluded. We start with the bottoms up approach i.e., by looking at requirements of the gaze estimation model parameters.

Requirements: The gaze estimation model as discussed in Chapter 3 requires the following parameters to be first calculated: $\mathbf{o}_1, \mathbf{o}_2, \mathbf{l}_1, \mathbf{l}_2, \mathbf{u}_{11}, \mathbf{u}_{21}, \mathbf{u}_{12}, \mathbf{u}_{22}, \mathbf{v}_1, \mathbf{v}_2$. Algorithms to determine these are explained as follows:

4-2-1 Camera Nodal Points $\mathbf{o}_1, \mathbf{o}_2$

For $\mathbf{o}_1, \mathbf{o}_2$, we start with reasoning that all coordinates in WCS are estimated with respect to the first nodal point $\mathbf{o}_1$. Hence, $\mathbf{o}_1$ is taken as the origin, $\mathbf{o}_1 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$. For the determination of the second camera nodal point we use the calibration procedure that was explained in the Section 3-1-3. This is done in MATLAB using the in-built calibration tool. This is done in few steps as explained:

Camera Calibration using the calibration object

The camera calibration procedure is as follows:

1. We select a printable calibration target of checkered pattern with 10×7 squares and print with 600dpi.
2. Verify the width of each square with the use of a vernier calipers. In our case, this came to be 3.5mm.
3. Mount the pattern onto a rigid flat surface.
4. Acquire Images of the target at different orientations and distances as seen in Figure 4-2-1.
5. Detect calibration target and compute parameters (MATLAB toolbox).
6. Save the Parameters object until the hardware state has been changed.

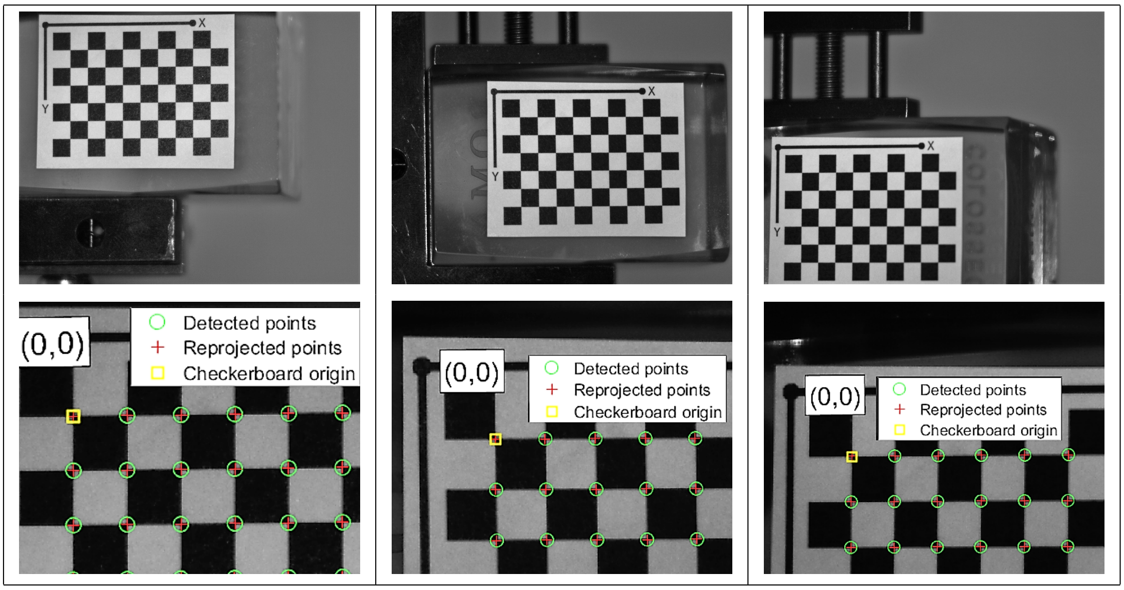

Figure 4-5: Calibration Pattern Detection - Second row shows the resulting detection of three calibration images(first row). Results are shown and zoomed in for better visual.

Summary: Imperfections on the surface may affect the accuracy of the calibration. Camera settings (focus, aperture, etc) are kept fixed through out the calibration procedure. 10 to 20 images are captured with different orientations of the pattern and to cover as much of the image frame as possible as lens distortion increases radially from the center of the image and sometimes is not uniform across the image frame. To capture this lens distortion, the pattern is shown close to the edges of the image frame as shown in Figure 4-6. The tool then detects the points on the pattern running the estimation to get the camera parameters as explained in Section 3-1-3. This gives us the translation and rotation of the second camera, hence the nodal point with re-projection error of less that half a pixel as seen from the results in Figure 4-6.

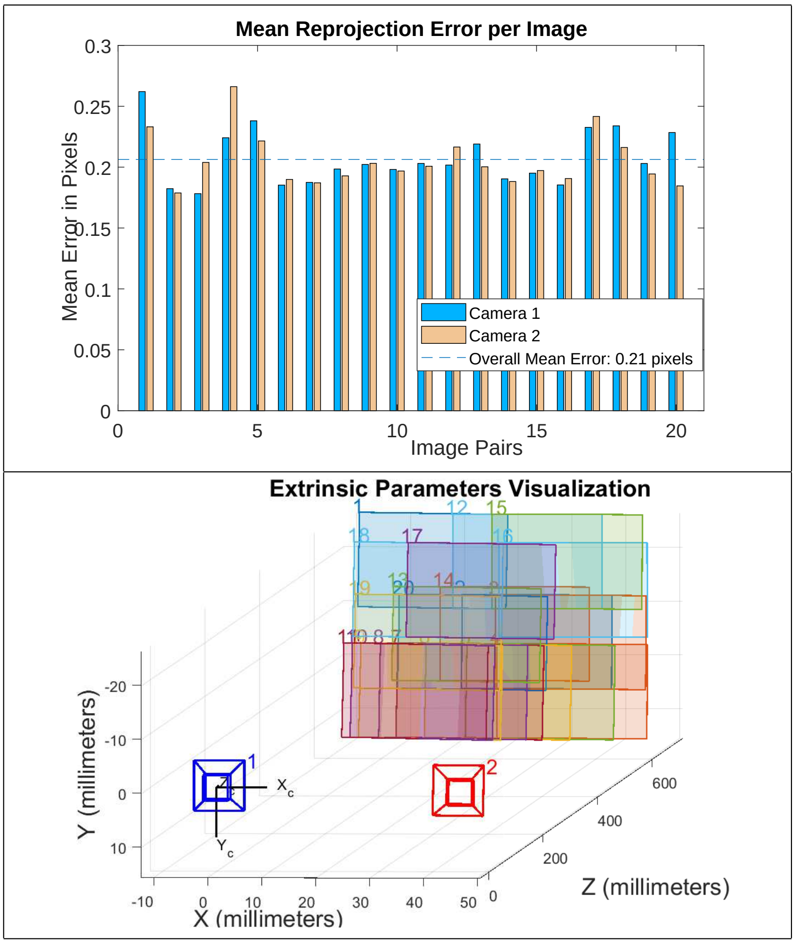

Figure 4-6: *Left:* The camera matrix is optimized for each point in the calibration and then re-projected on the plane. These Re-projection Errors are plotted. *Right:* The different patch seen here are the calibration plane that and been re-projected using the estimated points and it's Extrinsic matrix.

4-2-2 Absolute Orientation - LED Position $\mathbf{l}_1, \mathbf{l}_2$

For $\mathbf{l}_1, \mathbf{l}_2$, the distance with respect to the camera can be measured with a precisely machined LED holder for the setup and distances be found with Computer Aided Design (CAD) model and/or with various metrological instruments such as a vernier calipers. This introduced a certain rigidity to the system as the eye tracker can not be adjusted based on the patient's specific tumor location. In this section we introduce a new method that gives us flexible and more accurate measurement of the LED positions.

Experimental Setup: Measurement from this method allows us to quickly re-calibrate the LED positions and allow the change in LED position based on the treatment diagnosis and treatment planning. First we introduce a flat mirror in between the eye and the camera's field of view as shown in the setup layout in Figure 4-7. This is done in a way, so that the field of view of both the cameras contain the virtual images of the camera contour and the LED as shown in the Figure 4-8. Additionally, a calibration target which in this case is a

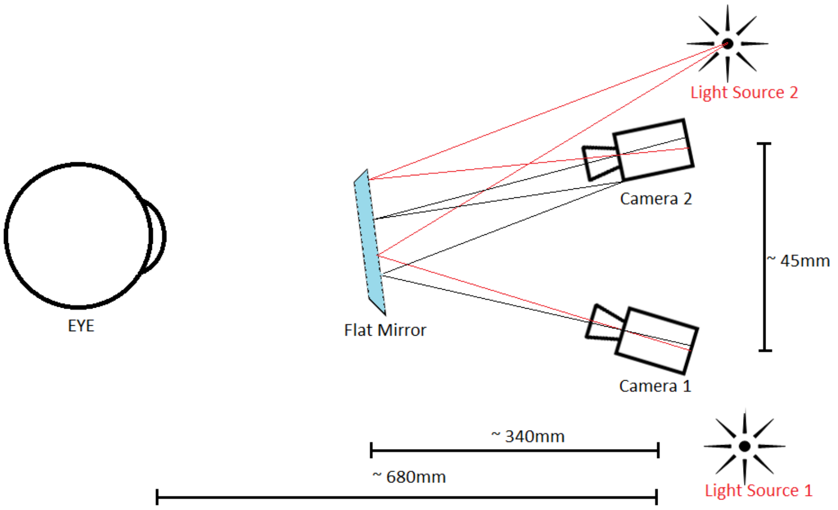

Figure 4-7: Experimental Setup: *Left:* We add a small checkered board target onto the camera's surface. *Right:* Detection results for the checkered board

checkered-board pattern printed on with roughly an area of $6mm^2$. Containing squares each of $0.5mm$ width which is shown on the zoomed image in Figure 4-8. Once the images frames are acquired as shown in the Figure 4-8, we run the checkered board detection algorithm. A corner detection and registration technique from [29] .

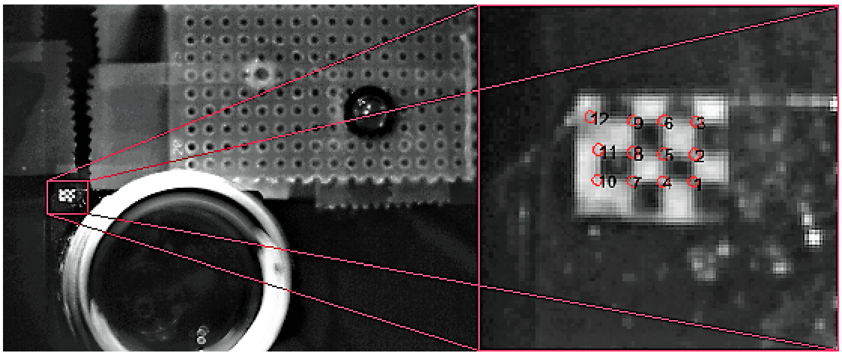

Figure 4-8: Detection of checkered pattern on the Camera. *Left:* we see the image taken by the stereo camera setup when it looks into the mirror, showing the virtual image of the camera along with the the placed checkered board. *Right:* We get the detection of the checkered board by applying corner detection.

Overall Algorithm Flow:

1. Acquire new image frame with camera and the LED with in the field of view.
2. Correct image for lens distortion with the distortion coefficient found in the camera calibration procedure.
3. Flip the image to have the same type of coordinate system (RHS Coordinate System)
4. Detect checkerboard pattern in Left and Right image (Average algorithm elapsed time is 0.7305 seconds)

5. The detected image points of the virtual image from first camera are matched with second camera and triangulated to get the 3D world coordinates.
6. Solve absolute orientation problem, by solving for unknown rotation matrix and unknown translation vector in the problem formulation (Eq. (4-4)). This implementation is sped up by using a closed form solution using unit quaternions [30]. With average algorithm time of 0.0013 seconds.

$$min \sum_i \|RV_i + t - K_i\|^2 \tag{4-4}$$

   where, R is the rotation matrix, t is translation vector, V_i is the i^{th} triangulated point in the virtual scene. K_i is the i^{th} corresponding known point.
7. Finally, the virtual LED is triangulated with the image coordinates for both left and right cameras and then the actual coordinates are found with extrinsic matrix $[R|t]$

Verification: To make sure that we have estimated the transformation matrix correctly, we divide the known points into training data and validation data, after the estimation from the training data, we use that on our validation data to get the results shown in Table 4-2. By computing the error for 15 points, we get the standard deviations to be $\sigma = 0.1135$. Which can be accounted for in precision for measuring the Z Coordinates of the known checkerboard points. Hence, the method works with accuracy depending on the detection accuracy, allowing for a flexible 3D measurement of the LEDs.

Table 4-2: Verification for Absolute Orientation Method *All Dimensions are in mm*

Points	Coordinates	Estimated Data	Validation Data	Error
1	x	13.3427	13.5190	-0.1763
	y	-17.8796	-17.7160	-0.1636
	z	-64.9119	-64.9500	0.0381
2	x	39.5696	39.5609	0.0087
	y	-12.6197	-12.5504	-0.0693
	z	-97.5750	-97.6470	0.0720
3	x	39.9323	40.0589	-0.1266
	y	-12.6921	-12.5421	-0.1500
	z	-97.7324	-97.6031	-0.1293

4-2-3 Pupil Detection $\mathbf{v}_1, \mathbf{v}_2$

Robust and accurate detection of the pupil position is essential for an accurate Point of Gaze (PoG) estimation. Here, the algorithm used, is the Circular Hough Transform (CHT).

Our focus here is to increase the robustness with varying illumination and detection in case of an occluded eye. Although, we only use the IR range, we see changes in the image when there is a change in the camera gain, aperture, LED luminescence or if exposure time is changed.

Circular hough transform: This algorithm identifies the three parameters of a circle which are the center coordinates x_c, y_c and radius of the circle r, represented in the parametric

Eq. (4-5). The CHT is not a specified algorithm and has a number of different approaches for it's implementation. We use the Atherton and Kerbyson's phase coding method [31]. This is described as follows. Starting with the parametric equation for a circle:

$$\begin{aligned} x &= x_c + r \times \cos\theta \\ y &= y_c + r \times \sin\theta, \end{aligned} \tag{4-5}$$

where, as angle θ sweeps through 2π, the points x, y complete a circle. If the approximate pupil size in our image is known (which is a range, as is the head movements have to be within the focus range), then the search space is reduced. Steps taken are:

- CHT accumulation matrix: Pixels with a high gradient are designated to be checked. Probability is accumulated around each of these pixels, and append into the accumulator. This is done for the complete image.
- Circle Center: The peaks in the accumulator matrix is then measured, giving us the center of the circle.
- Circle Radius: In case of multiple circles we compute the the radius in another step. But in case of our pupil, we can take the value from the same accumulator array.

The results from these steps are seen in the Figure 4-2-3 for different situations (with open eye, half open and closed eye lids)

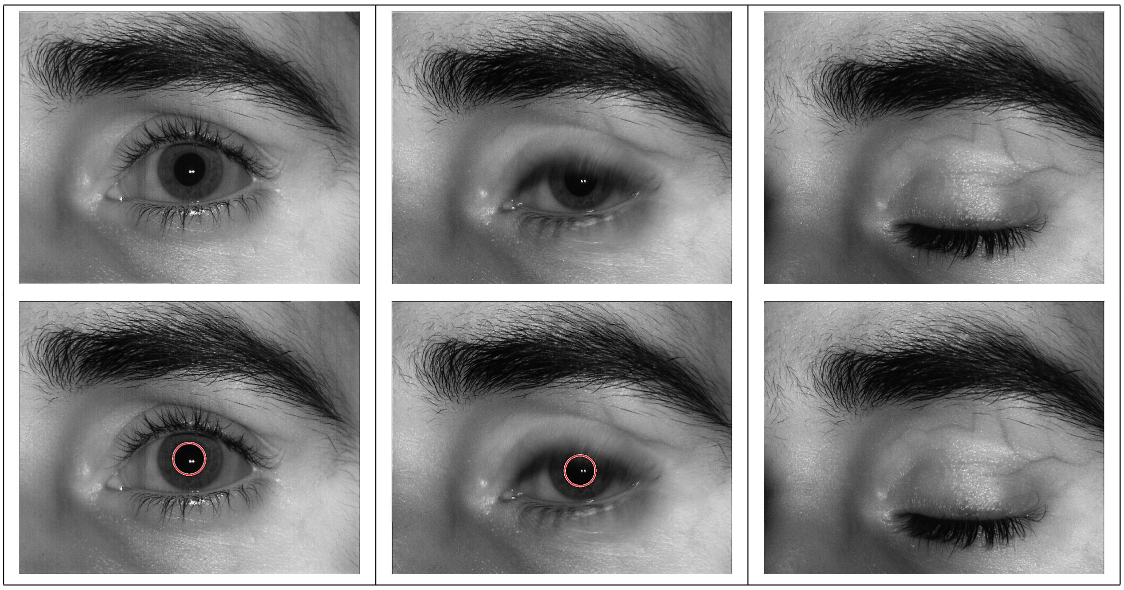

Figure 4-9: Cicular Hough Transform - *Left Column:* Detection for an open eye, *Center Column:* Detection for an occluded eye, *Right Column:* Detection fails only for an closed eye

Coordinate Transformation: Once we have the pupil detected in image coordinates, we must transform these to Camera Coordinate System (CCS) using our parameters which was acquired by the calibration tool box. We apply the following transformation:

$$\begin{bmatrix} x_{\text{ccs}} \\ y_{\text{ccs}} \\ z_{\text{ccs}} \end{bmatrix} = \begin{bmatrix} p_x \cdot (c - c_x) \\ p_y \cdot (r - c_y) \\ -\frac{(\frac{f_x}{p_x} + \frac{f_y}{p_y})}{2} \end{bmatrix}, \tag{4-6}$$

where, the LHS are the coordinates after transformation in CCS and the RHS is same as defined in Section 3-1-2.

Results: This approach is very useful because of it's robustness in the presence of noise, occlusion and varying illumination. This can also be used to detect any shape in cases of huge offset to the camera position causing the pupil shape to look like an ellipse or can be defined to any other parameterized shape.

4-2-4 Glint Detection $\mathbf{u}_{11}, \mathbf{u}_{21}, \mathbf{u}_{12}, \mathbf{u}_{22}$

For glint detection we first make use of the information from the pupil detection, using the pupil center and radius that was calculated earlier. We crop the image reducing our search area to 1.5 times the size of the pupil radius approximately being 2% of the original image. Which means reduced search space and increases the speed proportionally . The flow and the result of an detection is shown in the Figure 4-2-4.

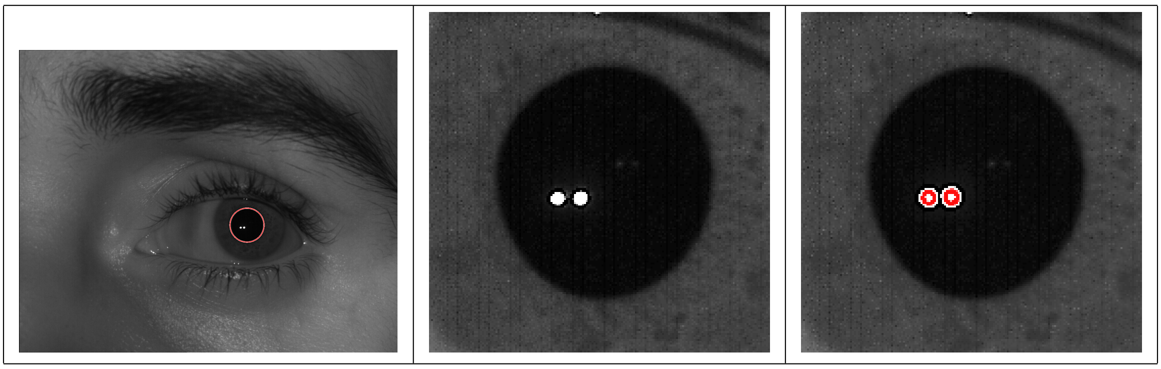

Figure 4-10: Cicular Hough Transform - Glint Detection (Elapsed time is 0.068730 seconds). *Left:* Shows the Pupil center and radius detection method, *Center:* Shows the cropped image, the pupil center is used to crop the probed image into 3x Pupil radii, *Right:* CHT is used to locate the glints with very low computational time due to low search space.

4-3 Software - Other detection algorithms

To be able to make the software better we would need to explore methods that take lesser computation power. We know that the images from our setup from each camera is of the resolution of 1280x1024.

4-3-1 Region of Interest:

One of the methods could be to be able to search for the features in a smaller resolution image i.e without losing the resolution near the features itself. This can be done in three ways,

1. By tweaking the hardware, i.e using a different lens system to zoom into the eye and acquire the images with binning, lowering the resolution with having the features within the image. This would mean, we can also switch the binning mode to get higher accuracy while losing on time. Would cause a trade-off. The main drawback of this is that we reduce the field of view of our system, thus restricting the possibility of large head movements. Thus, we did not investigate this method further

2. The other method is, using fast algorithms to get a feature and cut off the region around it for further processing as with the normal method.

3. The third method is modification to the second method, we first estimate the feature and process the cropped image, but during the run-time of the software as the pupil center move into different quadrants of the image, we constantly center and update the cropped image with pupil center. Thus, discarding the need to search again.

Hence, to investigate the second and third methods we analyzed few algorithms in our setup to detect the feature and match the computational speed with each other.

4-3-2 Viola-Jones algorithm

One of the algorithms investigated to detect the *eye* feature is the popular Viola-Jones algorithm [32]. We ran the algorithm the image and resulted in the area as shown in Figure 4-3-2. But this is sensitive to illumination and not robust for all images as we show in image with different illumination in Figure 4-3-2. Hence, would not be a good method as the types of eye and the skin around it vary a lot from patient to patient.

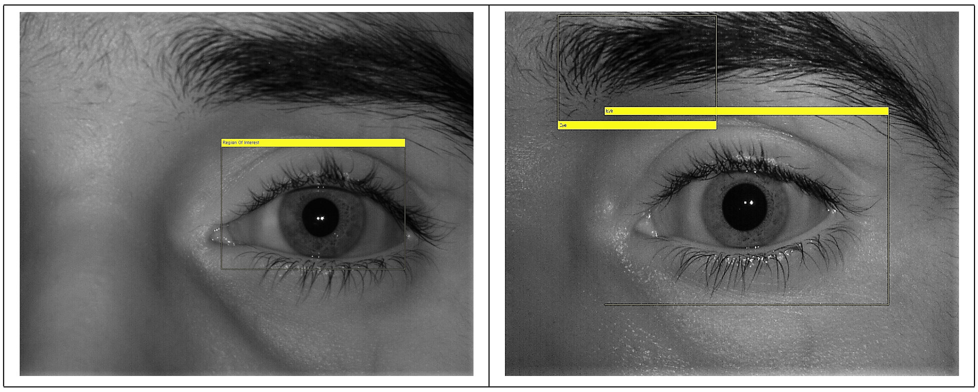

Figure 4-11: Viola-Jones algorithm with average elapsed time 0.03 Seconds. On the left we see a good detection by the algorithm. On the right we see a wrong detection due to change in illumination of the environment.

Image	BRISK	SURF	FAST	HARRIS	MinEigen	MSER	KAZE
1	0.2430	0.0942	0.0049	0.1869	0.1911	0.1348	0.7200
2	0.2450	0.0877	0.0048	0.1837	0.1894	0.1338	0.7138
3	0.2453	0.0917	0.0048	0.1849	0.1903	0.1413	0.7151

Table 4-3: Results: Average of computational time(in seconds) of 20 iterations of each image, done for 3 different images of varying intensities

4-3-3 Local Feature Extraction

As the corneal reflections in our image are almost like a point spread function around the pupil. They exhibit strong local feature with respect to the rest of the image. We use this information to scan for the strong features with the these algorithms and compare their speeds in running for about 20 measurements. The results for the detection is shown in Figure 4-3-3

BRISK Features [33] : Running this algorithm gave us the good results with strong features at the glint as shown in Figure 4-3-3

Speeded Up Robust Features (SURF) [34] : This algorithm is a patented local feature detector and descriptor. inspired by the scale-invariant feature transform (SIFT) descriptor, which we use later. This proved to faster and was able to successfully locate the feature points as shown in Figure 4-3-3

Features from Accelerated Segment Test (FAST) [35] : This is a corner detection algorithm but works to detect the glint as it acts like a strong corner point with pupil in it's background. This method was proposed for real-time application point of view which have limited computational resources.

Maximally Stable Extremal Regions (MSER) Features [36] : This is used as a method of blob detection in images. Initially proposed for stereo camera calibration to find correspondences between image elements from two images with different viewpoints. This can be used to find the features in our images. The results of this are shown in Figure 4-3-3.

Results: The results for the most prominent algorithms can be seen in Figure 4-3-3. As seen from the results These methods allow us to incorporate a high speed detection algorithm to either crop the image for the robust detection to work with higher speeds or can be worked more to use these methods directly for glint detection, but would not apply to pupil detection without larger modifications (such as complement of images) which might add to the computational time.

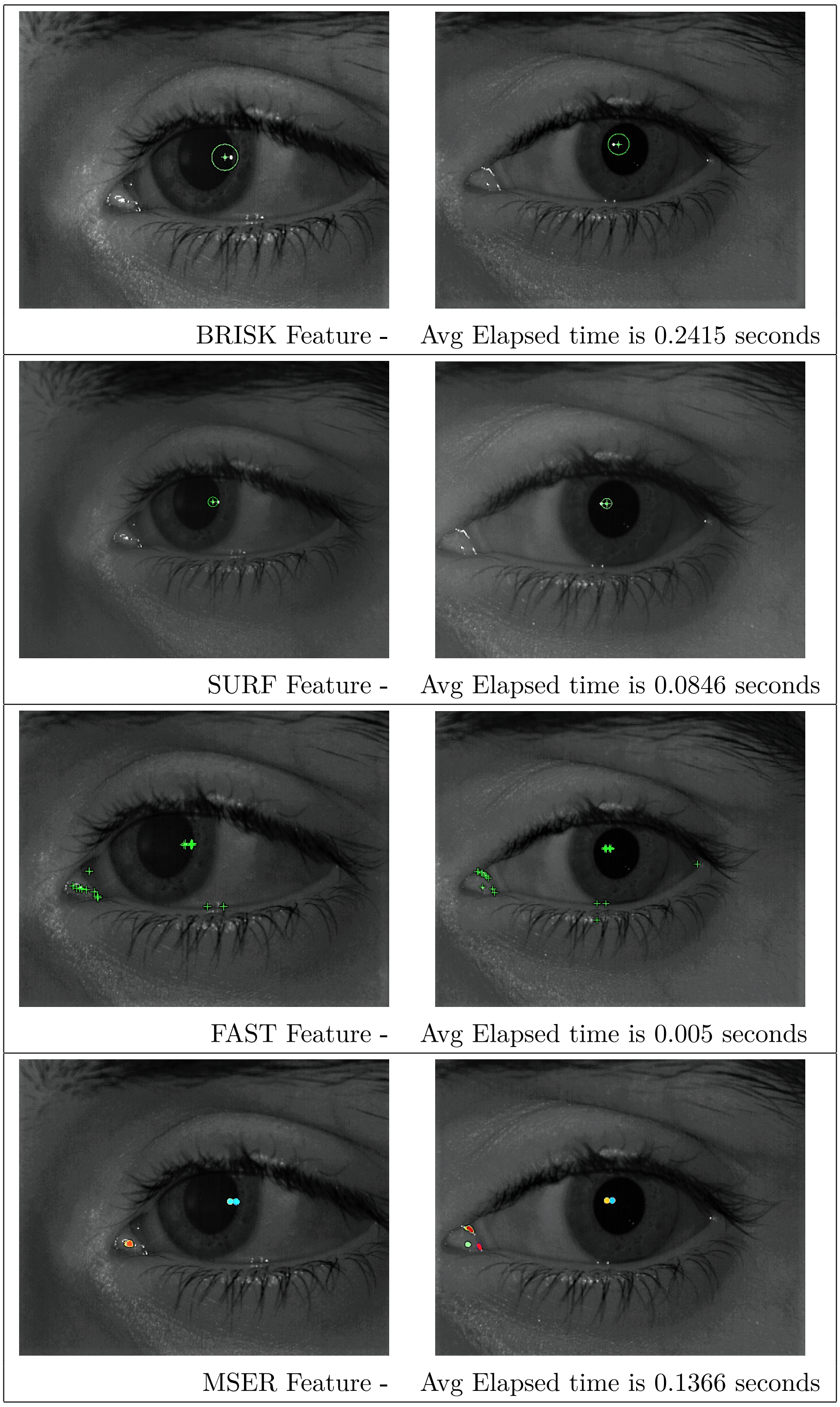

Figure 4-12: Feature Detection Algorithm Performances - *Images were tested on high resolution image (similar to Figure 4-3-2), but for the sake of visibility of features, here the images have been cropped. All detected features are within the eye, no outliers.* This method does not detect for a particular feature but for the strongest one, which are the ones with high gradient.

4-4 Saccades and Blink Detection

For our system to have a fast response to changes the main run of the software is to detect for changes with the initial frame of reference and triggers in case it exceeded a certain threshold. This is can be achieved by methods as follows:

2D Fast Normalized Cross Correlation

We use the correlation to detect the change from the first image. A probe (small part of the image with features) is extracted and then we run the cross correlation to detect the position/change in the position or if the eye lid moves for blinking. Hence, the probe must have a small part of the iris feature with a small part of the eye lid. This is implemented is visualized with the MATLAB function *normxcorr2* which follows the equation :

$$\gamma(u,v) = \frac{\sum_{x,y}\left[f(x,y) - \bar{f}_{u,v}\right]\left[t(x-u,y-v) - \bar{t}\right]}{\left\{\sum_{x,y}\left[f(x,y) - \bar{f}_{u,v}\right]^2 \sum_{x,y}\left[t(x-u,y-v) - \bar{t}\right]^2\right\}^{0.5}}, \tag{4-7}$$

where, f is the image, $\bar{t}$ is the mean of the template, $\bar{f}_{u,v}$ is the mean of $f(x,y)$ in the region under the template.

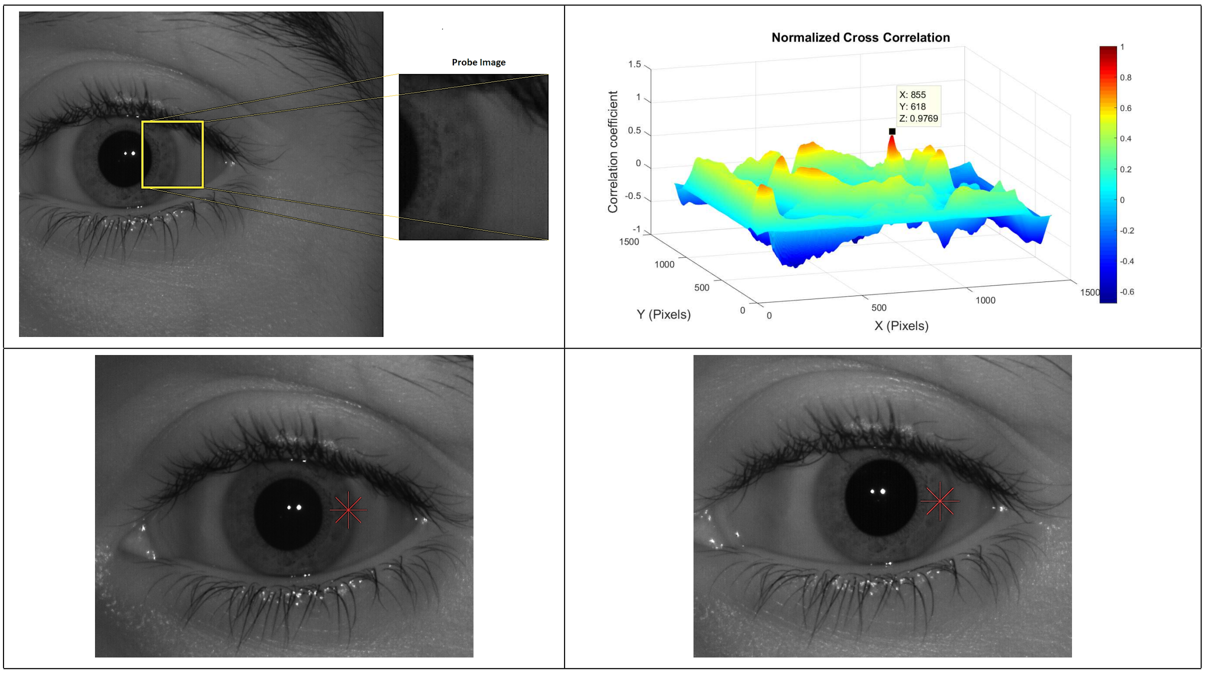

Figure 4-13: *Top-Left*: Cross Correlation: Display of the image and it's probe side-by-side.
Top-Right: Normalized Correlation Coefficient value for detection (Elapsed time is 0.388209 seconds with full image and 0.133167 with the cropped image).
Bottom: Cross Correlation applied on two different frames with the same probe image. We see the detection with a red star, giving a successful detection*Images cropped for better visual*

SIFT (Scale-invariant feature transform) Motion Detection

Patented by University of British Columbia, D.Lowe, in his paper *Distinctive Image Features from Scale-Invariant Key points* proposes robustly identify objects even among clutter and under partial occlusion, because the SIFT feature descriptor is invariant to uniform scaling, orientation, illumination changes, and partially invariant to affine distortion. We summarize the original SIFT algorithm as follows:

- Scale-space Detection : Key points are found by approximating the Laplacian of Gaussian at different scales and locating the extremes.
- Localization: Key points that are located are refined with Taylor Series Expansion
- Orientation Assignment: For rotation in-variance each point is assigned an orientation using the gradient of the neighborhood around.
- Descriptor Assignment: A 16x16 neighborhood around the key point is taken. It is divided into 16 sub-blocks of 4x4 size. For each sub-block, 8 bin orientation histogram is created. It is represented as a 128 vector to form key point descriptor.
- Key points between two images are matched by identifying their nearest neighbors.

We have modified this method to scan the iris region of the eye and compute the SIFT on limited pixels, increasing the speed and allowing us to locate small changes in each frame. The disadvantage with this is that we will not be able to account for large changes in Z dimension, but this may not be necessary in our application as the patient is seated with a head rest.

Figure 4-14: Optical flow with HornSchunck method. We are able to capture the divergence value for small movements. We captured a video of opening eyelid and extracted two frames from it. On the frame from left to center are single frame of the action of opening an eye lid. We apply the optical flow to this and overlay the detected vector field. We clearly see the blinking being detected. Value of divergence based on the specific parts of picture helps us detect either blinks or saccades.

Optical Flow

We implemented the optical flow to capture the blinking and possible saccades. This is done by optimizing a functional based on residuals from the brightness constancy constraint, and a particular regularization term expressing the expected smoothness of the flow field. The results are shown in Figure 4-14. The full resolution image (1280x1024) runs at 5.58Hz while a cropped image (160x120) had the frequency of 394.47Hz. If we localize the image crop area to just the eye and eye lid, we get (222x278) which helps preserve our features and we get a frequency of 156.0793Hz, which is over 3 times above our objective of 50Hz.

4-5 Software - Graphical User Interface

In order to make the whole process faster and more intuitive for the user, a Graphical User Interface (GUI) was created. The GUI was created by keeping in mind for the testing of different situations (Calibration, LED Measurement and Eye Images) and with varying parameters could be visualized in an easy and faster manner.

Figure 4-15 illustrates the first canvas of the GUI. The top half of the application is concerned with the Eye Tracker Calibration Procedure. Which include the options:

1. Capture Stereo Images
2. Calibrator App
3. Import Parameters
4. Estimate LED Position

The second half of the application is concerned with the real calculation towards gaze estimation. These include the options of:

1. Feature Detection
2. Saccade Detection
3. Calculate Point of Gaze

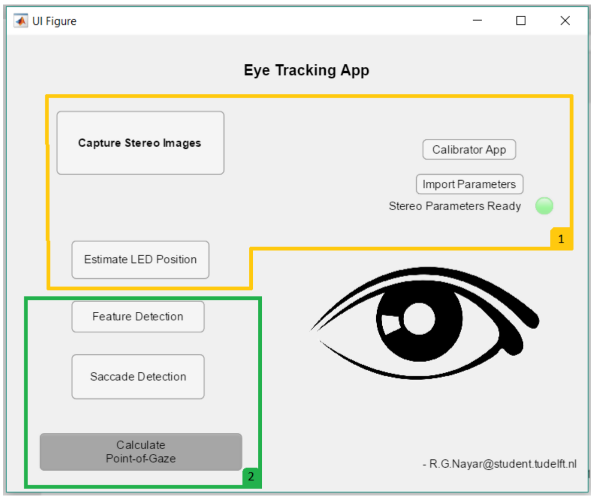

Figure 4-15: The opening screen panel of the Graphical User Interface (GUI) from where we control execution of each algorithm.

Capture Stereo Images: The screen-shot of this canvas is shown in **Figure 4-16** illustrating where different numbers can be found, which are explained as:

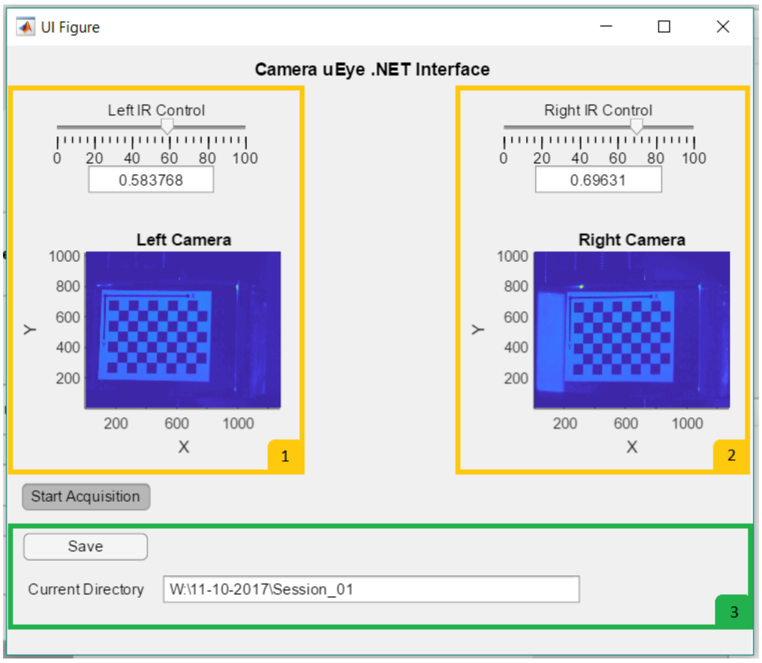

Figure 4-16: Button from the main panel opens this interface which connects to both the cameras via the .NET interface and is used to control the LED intensities and capture stereo images.

1. This can be seen as the left camera control panel where the image from the first camera is displayed in real time and the slider on top of the image allows us to set the duty cycle of the Pulse Width Modulation signals to control the intensity of the first LED
2. This works the same function as the first panel, but for the second camera.
3. Here we can start/stop the camera acquisition but a click of the button and save the files in pair corresponding folders for further processing. The text bar display the current active folder, which can be changed by typing in the path destination that is required.

This canvas helps us to use the .NET interface to capture images simultaneously from both the cameras and save the output in the directory that is automatically specified with the date and experiment number. As the LED are connected to their closest camera's GPIO pins, we are able to control the intensity of the LED with the two slider bars on the top end of the canvas.

MATLAB Calibrator Application: As the calibrator tool within MATLAB is an intensive and intuitive application, we make use of this in built tool. Clicking on the button "Calibrator App" in Figure 4-15 opens this tool. Images need to imported and the application runs the calibration routine and outputs the re projection errors. If the error are less that a pixel for over 15 calibration images, the *stereoParameter* object is exported to the MATLAB workspace, which is again imported by our application. If this is successful, we see the green bulb on the Figure 4-15 light up, confirming that we have successfully completed the calibration process.

Calculate Point of Gaze: The screen-shot of this canvas is shown in **Figure 4-17** illustrating where different numbers can be found, which are explained as:

1. This panel displays the feature points that have been calculated in the previous step and they are displayed. These can be edited to be able to analyze the model and correct for any errors, if need be.
2. This runs the Least square estimation based on the stereo configuration model that is explained in earlier chapter and estimates the Point of gaze by estimating the values p and c.
3. The values that are estimated can then be visualized for a clearer understanding by clicking on the "*Visualize*" button. The visualization is illustrated with the image on the right of Figure 4-17

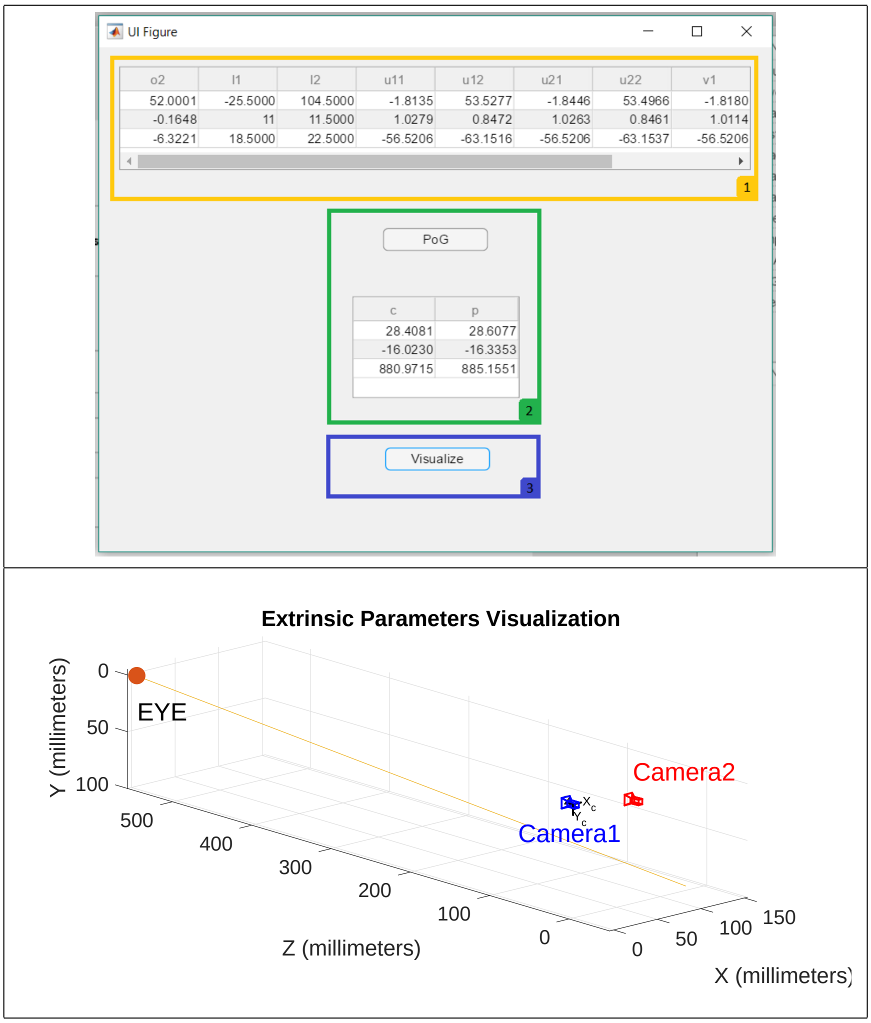

Figure 4-17: *Top:* PoG Estimation Panel. *Bottom:* Visualization for patient gazing towards a point (Position 5 in Figure 5-2).

4-6 Software Flowchart

To complete the software run for our system, we explain the complete program with a flowchart as seen in Figure 4-6

- Setup Initialization Stages
    1. Starts with establishing communication with the device using the .NET framework
    2. Each camera acquires image frames for the calibration object, with different orientations and these images are used to calibrate the stereo camera setup.
    3. Another image is taken after introducing a flat mirror, and the LED positions are estimated by the program.
    4. A reference image for the start of treatment is created capturing the initial position and gaze.
- Main Program
    1. Starts with acquiring live stream of image frames at the high fps.
    2. The frames are passed through a filter, that checks for any blurry images and/or occluded eye i.e. images without feature points (Glint or Pupil)
    3. The saccade are detected in each frame and compared to the threshold valu set for the highest divergence. The saccade algorithm also accounts for the eye blinks.
    4. If the threshold has exceeded the threshold or completed certain number of runs. We turn on the PWM signals to the LED
    5. The LED creates the Glint and this is detected by the feature detection algorithm along with the pupil.
    6. The change in pupil size is found with the reference image taken at the start
    7. All the detected points are converted to WCS and the Least Squares estimate gives us the Gaze Direction.
    8. These values are output to the Proton delivery system for gating and the program repeats.

This the completes development of the prototype, we now gather measurements to check the accuracy and run analysis on the performance of our system along with it's sensitivity to measurements.

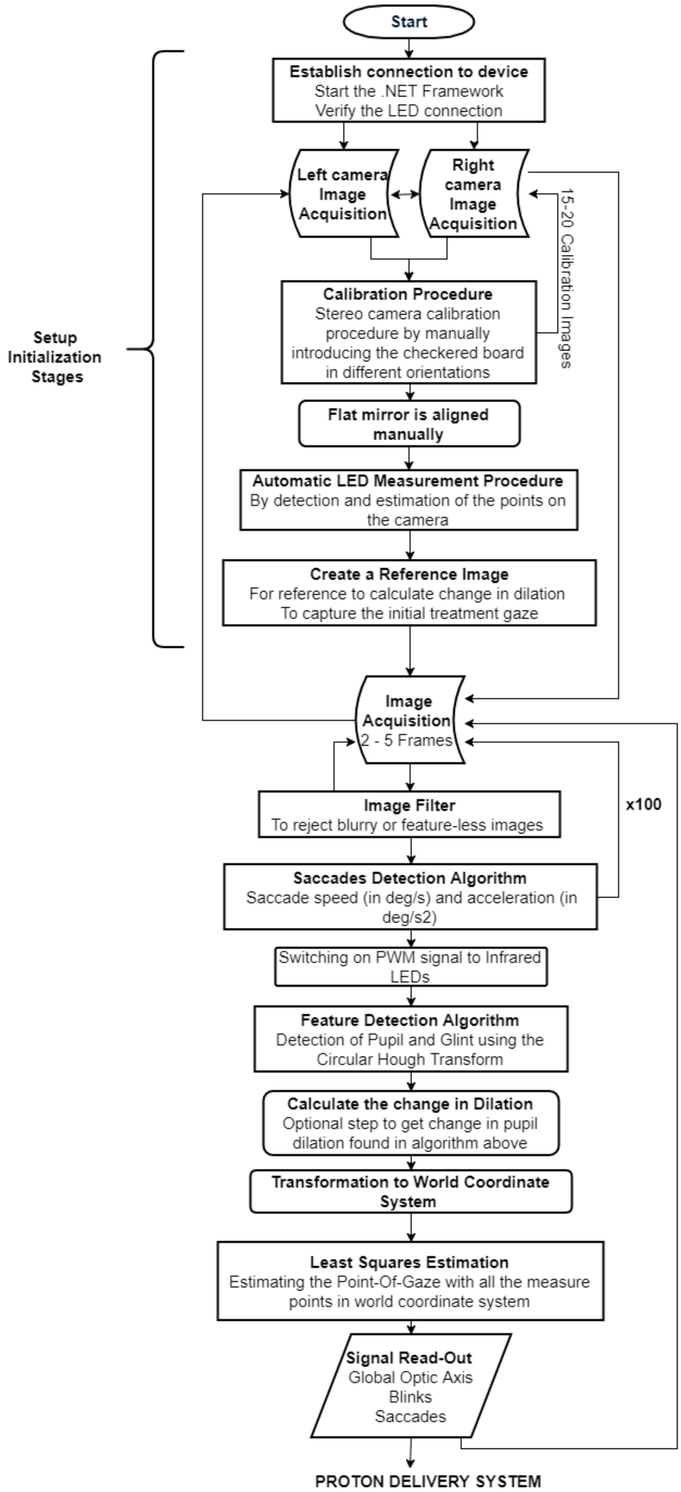

Figure 4-18: Software Flowchart

Chapter 5

Model Sensitivity and Results

In this chapter, we first study of how uncertainty in the output of a model can be attributed to different sources of uncertainty in our model input.

5-1 Sensitivity Analysis

Testing the sensitivity to the eye model to small changes in the values of measured parameters. The final model is expressed in the matrix form as:

$$\underbrace{\begin{bmatrix} [(l_1 - o_1) \times (u_{11} - o_1)]^T \\ [(l_1 - o_2) \times (u_{12} - o_2)]^T \\ [(l_2 - o_1) \times (u_{21} - o_1)]^T \\ [(l_2 - o_2) \times (u_{22} - o_2)]^T \end{bmatrix}}_{M_2} c = \underbrace{\begin{bmatrix} (l_1 - o_1) \times (u_{11} - o_1) \bullet o_1 \\ (l_1 - o_2) \times (u_{12} - o_2) \bullet o_2 \\ (l_2 - o_1) \times (u_{21} - o_1) \bullet o_1 \\ (l_2 - o_2) \times (u_{22} - o_2) \bullet o_2 \end{bmatrix}}_{h}, \tag{5-1}$$

with M_2 having the rank of 3, c is the coefficient vector of the least-squares hyperplane expressed as (i.e left pseudoinverse):

$$c = (M_2^T M_2)^{-1} M_2^T h. \tag{5-2}$$

Without knowing the subject parameters we can now determine the direction of $(p - c)$ and hence the optical axis with the equations:

$$s = [(o_1 - v_1) \times (c - o_1)] \times [(o_2 - v_2) \times (c - o_2)], \tag{5-3}$$

$$p - c = k_{pc} s_{norm}, \tag{5-4}$$

where

$$s_{norm} = \frac{s}{\|s\|}.$$

5-1-1 Least Square Analysis

We first look into the least squares problem defined in Eq. (5-1) and Eq. (5-2). Starting with the analysis on the value of c. Where,

$$M_2^T M_2 c = M^T h.$$

The first-order perturbation is

$$\delta M_2^T M_2 c + M_2^T \delta M_2 c + M_2^T M_2 c = \delta M_2^T h = M_2^T \delta h,$$

rearrange to get,

$$\delta c = (M_2^T M_2)^{-1} \delta M_2^T (h - M_2 c) + (M_2^T M_2)^{-1} M_2^T (\delta h - \delta M_2 c).$$

If we define $r = h - M_2 c$ and let $M_2 = U \sum V^T$ with singular values $\sigma_1, \cdots, \sigma_n$, we have ,

$$\|\delta c\| \leq \left\|(M_2^T M_2)^{-1}\right\| \|\delta M_2\| \|r\| + \left\|(M_2^T M_2)^{-1} M_2^T\right\| (\|\delta h\| + \|\delta M_2\| \|c\|)$$

$$= \frac{\|\delta M_2\|}{\sigma_n^2} \|r\| + \frac{1}{\sigma_n} (\|\delta h\| + \|\delta M_2\| \|c\|).$$

Dividing the equation with $\|x\|$ and re-arranging the terms for ratios, we get:

$$\frac{\delta c}{c} \leq \kappa(M_2)^2 \frac{r}{\|M_2\| \|x\|} \frac{\|\delta M_2\|}{\|M_2\|} + \kappa(M_2) \left(\frac{\|h\|}{\|M_2\| \|c\|} \frac{\delta h}{h} + \frac{\|\delta M_2\|}{\|M_2\|} \right).$$

We can draw a right triangle who's sides are $\|M_2 c\| \leq \|M_2\| \|c\|, \|r\| and \|b\|$. In terms of this we have:

$$\frac{\|r\|}{\|M_2\| \|c\|} \leq \frac{r}{\|M_2 c\|} = \tan(\theta),$$

$$\frac{\|h\|}{\|M_2\| \|c\|} \leq \frac{h}{\|M_2 c\|} = \sec(\theta).$$

Combining the equations,

$$\frac{\|\delta c\|}{\|c\|} \leq \left(\kappa(M_2)^2 \tan(\theta) + \kappa(M_2) \right) \frac{\|\delta M_2\|}{\|M_2\|} + \kappa(M_2) \sec(\theta) \frac{\|\delta h\|}{\|h\|}. \tag{5-5}$$

When the angle is small, the relative changes to M_2 are amplified by the condition number. Whereas, when angle is large, the amplification factor is squared condition number.

Results in Practice: We took measurements in 9 different gaze angles, we see similar results also shown in the Figure 5-3. Picking first 3 measurements of each gaze direction, we analyze the values from our setup and the matrix M_2 with θ and it's condition number in the Table 5-1.

Table 5-1: Results from Setup for LSE Sensitivity

Gaze Direction	$\kappa(M_2)$	θ(Degrees)	Amplification Factor M2	Amplification Factor h
1 - RightTop	68.1482	1.2643	170.6470	68.1648
	68.3021	1.3242	176.1421	68.3203
	68.1168	1.0110	149.9984	68.1275
2-Top	68.5821	1.3410	178.6877	68.6009
	68.4254	1.2180	167.9763	68.4408
	68.7439	1.1668	164.9947	68.7581
3-LeftTop	68.3082	1.2168	167.4173	68.3236
	68.4649	1.2972	174.6120	68.4825
	68.1586	1.1440	160.9295	68.1722
4-Left	68.9661	1.3342	179.7449	68.9848
	69.0893	1.3002	177.4340	69.1071
	68.9501	1.3193	178.4440	68.9684
5-LeftDown	68.0358	1.2412	168.3340	68.0518
	67.9665	1.2660	170.0562	67.9831
	68.0164	1.2609	169.8427	68.0329
6-Down	69.4147	1.1133	163.0560	69.4278
	69.5208	1.1826	169.3003	69.5356
	69.5379	1.2566	175.6081	69.5546
7-RightDown	69.5290	1.3644	184.6787	69.5487
	69.6147	1.2078	171.7898	69.6302
	69.7274	1.2934	179.4999	69.7451
8-Right	67.7155	1.2601	168.5846	67.7319
	67.6996	1.2742	169.6469	67.7164
	67.6529	1.2713	169.2241	67.6696
9-Center	68.3698	1.2876	173.4419	68.3871
	68.3017	1.3334	176.8963	68.3202
	68.3429	1.2751	172.3102	68.3598

Conclusion: We see that the maximum values of the amplification factor is large, which in practice does not give our system a good flexibility but this can be slightly accounted with the amplification factor of the second term, as we know the vectors used for the formation of matrix M_2 is also used in the formation of h. Hence, the changes in the vector h can be said to be dependent to the changes in the matrix M_2. This would either increase the error or negate the error based on the dependency or the matrix. To have a better understanding, in the next section we will correspond this by running local sensitivity analysis of the Least Squares and analyze the results.

5-1-2 Local Sensitivity

Our M_2 matrix and h vector used for the Least Square Estimation comprises of algebraic operations between of the measured 3D world coordinates, which are measure upto an accuracy. To analyze the sensitivity of these measurements to our final gaze direction we perturb our measurements one at a time. By adding the perturbation on each coordinates of LED1, LED2 and camera 2 distance we study the change in euclidean distance of the corneal center estimation we can get an idea on the error sensitivity. The equation shown below is by adding offset to the x coordinate of LED1, where the perturbations value is shown with δ:

$$\underbrace{\begin{bmatrix} \left[(l_1 + \begin{bmatrix}\delta & 0 & 0\end{bmatrix}^T - o_1) \times (u_{11} - o_1)\right]^T \\ \left[(l_1 + \begin{bmatrix}\delta & 0 & 0\end{bmatrix}^T - o_2) \times (u_{12} - o_2)\right]^T \\ [(l_2 - o_1) \times (u_{21} - o_1)]^T \\ [(l_2 - o_2) \times (u_{22} - o_2)]^T \end{bmatrix}}_{M_2} c = \underbrace{\begin{bmatrix} (l_1 + \begin{bmatrix}\delta & 0 & 0\end{bmatrix}^T - o_1) \times (u_{11} - o_1) \bullet o_1 \\ (l_1 + \begin{bmatrix}\delta & 0 & 0\end{bmatrix}^T - o_2) \times (u_{12} - o_2) \bullet o_2 \\ (l_2 - o_1) \times (u_{21} - o_1) \bullet o_1 \\ (l_2 - o_2) \times (u_{22} - o_2) \bullet o_2 \end{bmatrix}}_{h} \tag{5-6}$$

Experiment: For the error propagation analysis we take use each measurement to get our matrices for LSE for the value of c . Since, c is a 3D vector, we measure the change in the magnitude of the vector i.e. $\|c\|$. First, we add a random signal with specified variance $\sigma^2_{\text{purtabation}}$ and measure the output variance σ^2_{output}. Further, plot the change in the standard deviation $\sigma(\|c\|)$ with different standard deviation noise signals.

Results of the Experiment: The results from the experiment are illustrated in Figure 5-1-2. We can clearly deduce that the sensitivity towards each coordinate is varied. The practical results show better results being less sensitive to perturbation. With the amplification factor being less that the perturbation amplitude in all cases. With perturbations in Z Coordinate not being affect as we would expect.

5-2 Gaze Estimation with Fixation Points

In order to quantify the accuracy of the gaze tracking system, a subject without glasses was asked to participate. The subject was asked to look at 9 different gaze points which was

Table 5-2: Results in Angular deviation (in Degrees) of the eye for each gaze angle. We take 40 measurements in each gaze direction and compute the deviation in each direction.

Gaze Direction	Angle to y-z plane	Angle to z-x plane	Angle to x-y plane
1	0.4186	0.5287	0.5735
2	0.3149	0.2722	0.3150
3	0.3587	0.2652	0.2070
4	0.6917	0.5557	0.4224
5	0.7885	0.7489	0.6279
6	0.5348	0.5015	0.2680
7	0.8499	0.5433	0.6087
8	0.8038	0.4280	0.4057
9	0.3532	0.3447	0.3936
Mean :	0.5682	0.4653	0.4247

made in the form of a grid, as shown in Figure 5-2. 40 measurements for each gaze direction was recorded, with allowing for natural head movements. and allowed to take breaks, since resuming does not require re-calibrating the system.

5-3 Results

Results of the experiments are clearly visualized in the Figure 5-3. The plot shows us the deviations that occur with the patient looking in one direction for 40 image frames. Accuracy of our system is better measured with angles of the gaze estimation rather than the spatial gaze fixation. We plot the Table 5-2 with standard deviation of the angles made by the points that intersect the fixation plane. The mean deviation of the angle with y-z plane is $\mathbf{0.5682}^{\circ}$, with z-x plane is $\mathbf{0.4653}^{\circ}$ and with x-y plane is $\mathbf{0.4247}^{\circ}$. Based on overall angle(addition of all three) at a particular measurement, we can conclude that our eye tracking system gives us the accuracy $< \mathbf{1}^{\circ}$ which is the resolution of fovea, in regards with the eye fixation points.

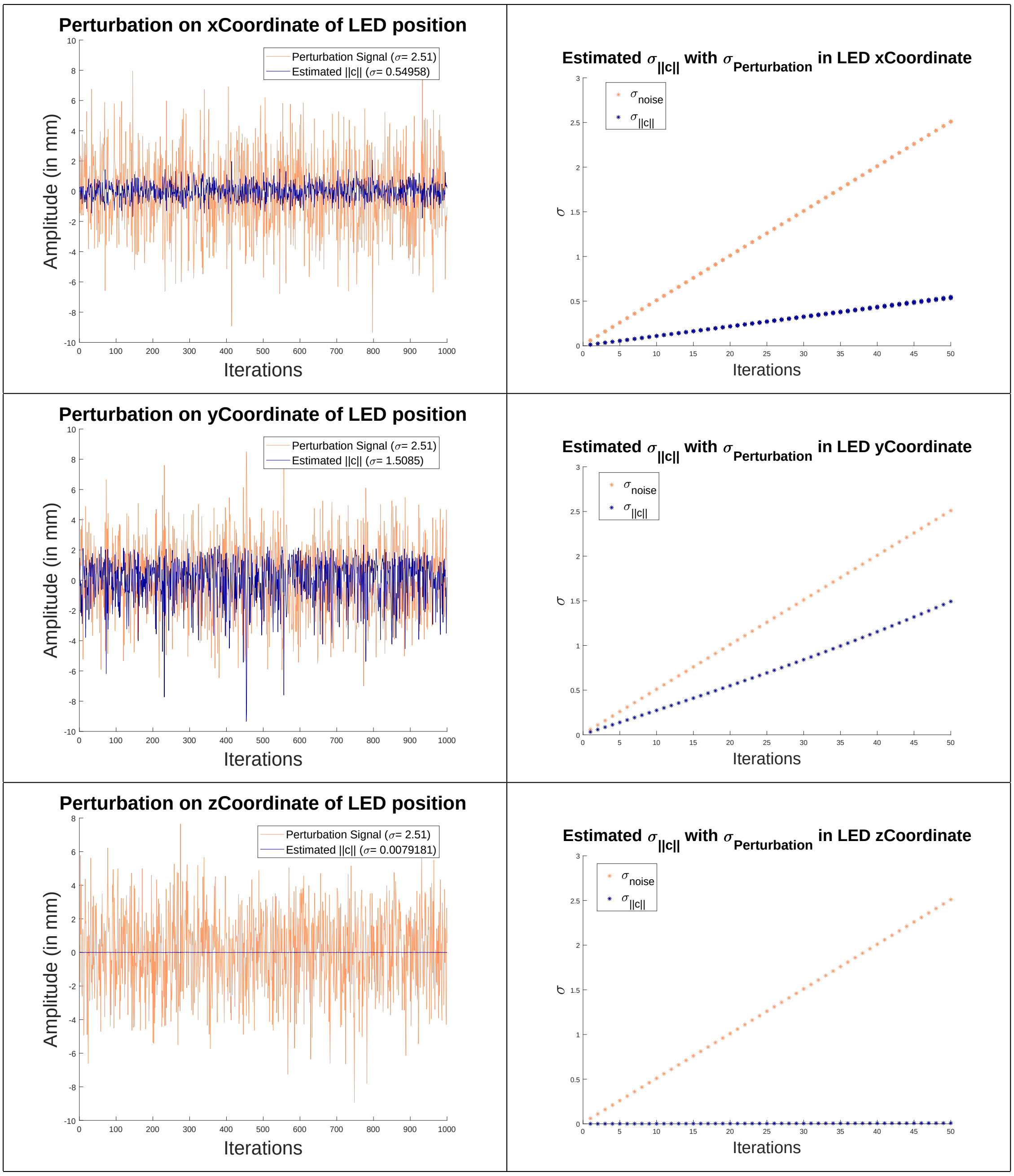

Figure 5-1: Local Perturbation analysis on coordinates of the Left LED (The Matrix values for this measurement can be seen on the first row of Table 5-1). Based on our calculations in perturbation, the values are as expected. Since, the maximum amplification factor was very high, we do meet the expectation, but not conclusively.

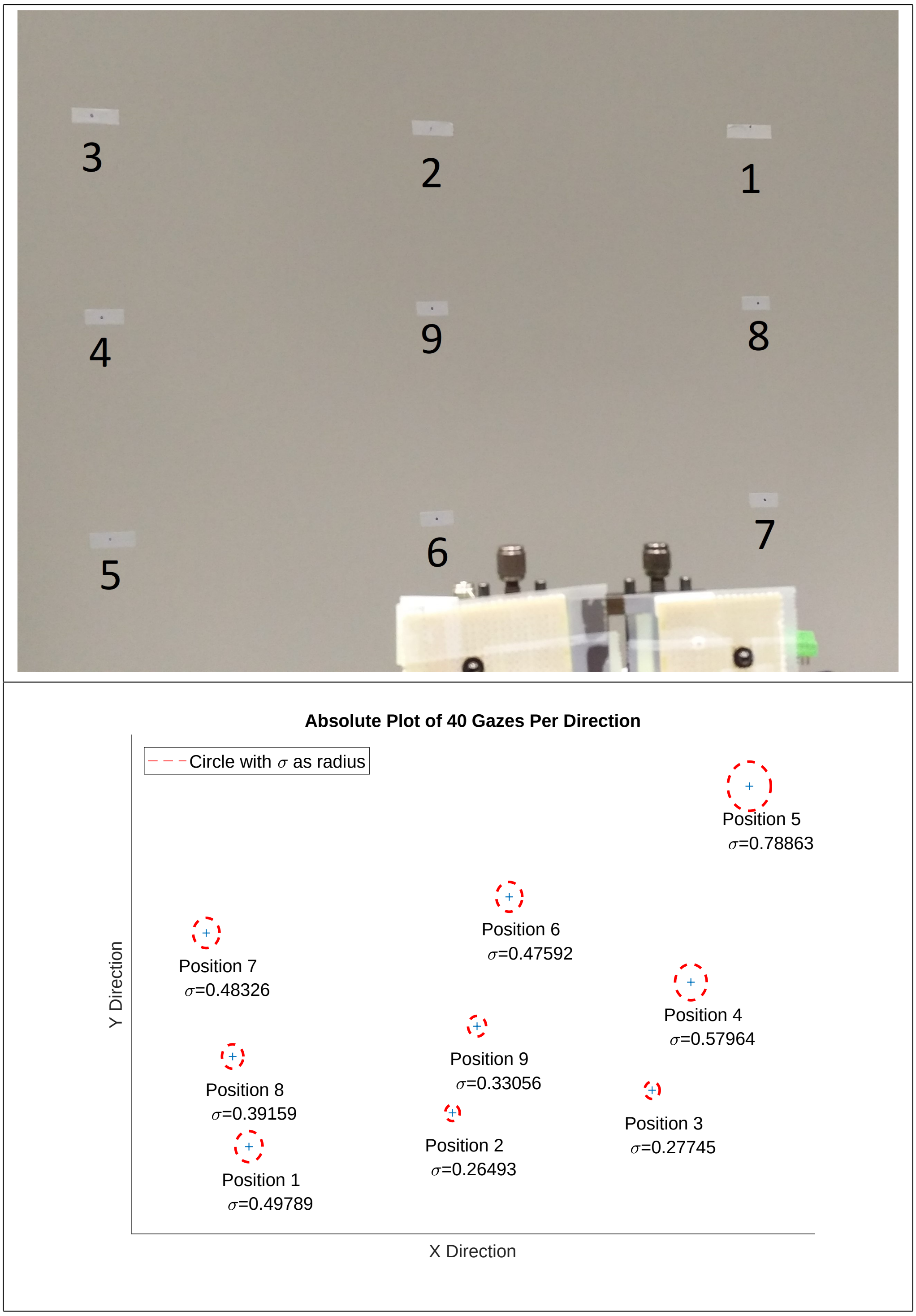

Figure 5-2: *Top:* Fixation Targets placement with the setup (as viewed by the patient) *Bottom:* Absolute Gaze Estimates from the Measurements without subject specific parameters

Figure 5-3: Estimation for 9 Gaze fixations for 40 measurements. Each direction is bounded with 1mm radius around it's mean. Distances between each direction is not to scale.

Chapter 6

Conclusion

The main goal of this thesis is to develop a prototype eye tracking device which will improve the traditional eye tracking procedure during proton therapy of uveal melonama. Our stereo camera configuration has shown to work with meeting our objectives that were stated at the introduction.

- Flexibility: Our minimum device volume is around $125cm^3$, with flexibility to move parts for flexibility.
- Speed: main loop running at 150Hz, we have three times the requirement.
- Non Invasive: Our low power illumination source at $860nm$ do not damage the eye.
- Ease of Use: Our GUI makes it easy to use the program.
- Upgrade: Using the AOI mode of camera we can upgrade the system upto 1000fps.

6-1 Contributions

Contributions of the thesis is summarized as follows:

1. Built an experimental eye tracking setup to be used for treatment of uveal melanoma with less than 1° accuracy.
2. A new method with sub-pixel accuracy measurement for LED position with sub-pixel accuracy has been proposed and verified.
3. A program flowchart is introduced with the fastest loop at around 400Hz, that reduces the computational time, meeting the our requirement of 50Hz gating speeds of the proton beam.

4. Sensitivity analysis of the proposed method which can be used to compare with different eye optical models.
5. Created a Graphical User Interface (GUI) for easy use of the program and implementation.
6. Time and computational performance comparison for different vision algorithms.

Although, the accuracy achieved by our system is less than 1°, which conforms with the fovea resolution. Yet, we do not conclude the repeat ability of the system, as during our measurements patient might not have been able to focus steadily. We propose a method to validate our system in the recommendations

6-2 Recommendations

Accuracy and Precision: For a precise specification, a validation tool maybe employed which would be a prosthetic eye ball on 6DOF movable stage with a motion resolution in micro meters. This would remove any human uncertainty and we can validate the accuracy and precision of the system to micro meters.

Computational Performance: Though each algorithm ran with very low computational time, when running the program as a whole, we do not achieve real time performance with MATLAB. Hence, the speed is another aspect to consider for the next device - processing in MATLAB will make copies of the data during it's processing time. Whereas, using a C# interface, the programmer can allocate and manipulate the data directly and ensure maximum performance. This would be of a concern when analyzing with live stream data in real time. This should be employed for the next version of the device. Addition to development of C code, we can introduce multi-core programming and/or use parallel computing by using the capabilities of a Graphical Processing Unit (GPU)

Eye Modeling: The simplicity of the model does not account for patient specific features. Eye's have aspherical cornea and multiple refractive indices which can be taken into account for even more accurate device.

6-3 Future Aspects

Eye tracking is a promising technology for safe and non invasive surgery, but this technology can be accustomed to be used in other sectors due it's flexibility to environmental changes. More importantly they may be used for people with disability to have easier interactions to digital world.

Appendix A

A-1 Codes

A-1-1 Camera Acquisition with Direct3D in .NET framework

```
%% Camera Acquisition
clear all;

% Check the specific location of library
asm = System.AppDomain.CurrentDomain.GetAssemblies;
if ~any(arrayfun(@(n) strncmpi(char(asm.Get(n-1).FullName), ...
        'uEyeDotNet', length('uEyeDotNet')), 1:asm.Length))
    NET.addAssembly(...
        'C:\Program Files\IDS\uEye\Develop\DotNet\signed\uEyeDotNet.dll')
            ;
end

% Create the camera object and specify the cameraID for the specific
    camera
camL = uEye.Camera; camL.Init(1);
camR = uEye.Camera; camR.Init(2);

% Enable the Direct3D Mode
camL.Display.Mode.Set(uEye.Defines.DisplayMode.Direct3D);
camR.Display.Mode.Set(uEye.Defines.DisplayMode.Direct3D);
err = camL.PixelFormat.Set(uEye.Defines.ColorMode.Mono8)
err = camR.PixelFormat.Set(uEye.Defines.ColorMode.Mono8)

% Allocate Camera Memory
[err, IDL] = camL.Memory.Allocate(true);
[err, WidthL, HeightL] = camL.Memory.GetSize(IDL);
camL.DirectRenderer.SetStealFormat(uEye.Defines.ColorMode.Mono8);

[err, IDR] = camR.Memory.Allocate(true);
[err, WidthR, HeightR] = camR.Memory.GetSize(IDR);
camR.DirectRenderer.SetStealFormat(uEye.Defines.ColorMode.Mono8);
```

```
% Set up Working Directory
ExpNum=1;
FolderName=strcat(pwd,datestr(now, 'mm-dd-yyyy'),...
    sprintf('\\Exp_%02d',ExpNum));
while 7==exist(FolderName,'dir')
    ExpNum=ExpNum+1;
    FolderName=strcat(pwd,datestr(now, 'mm-dd-yyyy'),...
        sprintf('\\Exp_%02d',ExpNum));
end
mkdir(FolderName);
mkdir(strcat(FolderName,'\\L'));
mkdir(strcat(FolderName,'\\R'));

% Set up Matlab figure with Stop and Save buttons
clf
hStp = uicontrol('Style', 'ToggleButton', 'String', 'Stop', ...
    'ForegroundColor', 'r', 'FontWeight', 'Bold', 'FontSize', 20);
hStp.Position(3:4) = [100 50];
hSave = uicontrol('Style', 'pushbutton', 'String', 'Save', ...
    'ForegroundColor', 'b', 'FontWeight', 'Bold', 'FontSize', 20,'
        UserData', 0,...
    'CallBack', {@SavePushButton,camL,camR,FolderName});
hSave.Position = [200 20 100 50];
subplot(1,2,1)
hImg_L = imagesc;
axis(hImg_L.Parent, 'image');
axis(hImg_L.Parent, 'tight');
hx = line(0, 0, 'Color', 'r', 'LineWidth', 2);
hy = line(0, 0, 'Color', 'r', 'LineWidth', 2);
subplot(1,2,2)
hImg_R = imagesc;
axis(hImg_R.Parent, 'image');
axis(hImg_R.Parent, 'tight');
hx = line(0, 0, 'Color', 'r', 'LineWidth', 2);
hy = line(0, 0, 'Color', 'r', 'LineWidth', 2);


% Start capturing
camL.Acquisition.Capture;
camR.Acquisition.Capture;

%Loop until stop button is pressed
T = zeros(10, 1);
tic
while ~hStp.Value || strcmp(err, uEye.Defines.Status.NO_SUCCESS)
    % Wait until image is copied from the Graphics card
    camL.DirectRenderer.StealNextFrame(uEye.Defines.DeviceParameter.Wait)
        ;
    camR.DirectRenderer.StealNextFrame(uEye.Defines.DeviceParameter.Wait)
        ;

    % Transfer and reshape the image into a Matlab array
```

```
    [err, I_L] = camL.Memory.CopyToArray(IDL);
    [err, I_R] = camR.Memory.CopyToArray(IDR);
    I_L = reshape(uint8(I_L), WidthL, HeightL).';
    I_R = reshape(uint8(I_R), WidthR, HeightR).';

    % Plotting the data
    Ix_L = sum(uint64(I_L));
    Iy_L = sum(uint64(I_L), 2);
    Ix_R = sum(uint64(I_R));
    Iy_R = sum(uint64(I_R), 2);
    subplot(1,2,1)
    hImg_L.CData = flipud(I_L);

    % Plot the fps rate
    T = [T(2:end); toc];
    title(sprintf('FPS: %.1fs %s', 10/diff(T([1 end])), ...
        datestr(now, 'HH:MM:SS.FFF dd/mm/yyyy')));
    drawnow;
    subplot(1,2,2)
    hImg_R.CData = flipud(I_R);
    T = [T(2:end); toc];
    title(sprintf('FPS: %.1fs %s', 10/diff(T([1 end])), ...
        datestr(now, 'HH:MM:SS.FFF dd/mm/yyyy')));
    drawnow;
end

% Stop Acquisition and Clear Memory for next run
hStp.Value = false;
camL.Acquisition.Stop; camR.Acquisition.Stop;
camL.Memory.Free(IDL); camR.Memory.Free(IDR);
camL.Exit;camR.Exit;
close all
```

A-1-2 Control the GPIO Pins in in .NET framework

```
% Create camera object
camL = uEye.Camera;
camR = uEye.Camera;

% Initialize the camera
camL.Init(1);
camR.Init(2);

%Initialize the PIN 1 to PWM output with GPIO Mode
camL.IO.Pwm.SetMode(uEye.Defines.IO.PwmMode.Gpio1);
camR.IO.Pwm.SetMode(uEye.Defines.IO.PwmMode.Gpio1);

%Set the Frequecy and Duty Cycle of the PWM Signal
camL.IO.Pwm.SetParams(2000, 0.5);
camR.IO.Pwm.SetParams(2000, 0.5);

%Release the camera objects
camL.Exit;
```

```
camR.Exit;
```

A-1-3 Pupil Detection - Circular Hough Transform

```
function [ center, radii ] = pupildetect(image)

% Initialize
I=image;
inc=0.005;

% Filter and binarize the the image for smooth detection
T = adaptthresh(I,0.9);
I=wiener2(I,[10 10]);
BW=imbinarize(I,T);

% Circular Hough Tranformation
[center, radii] = imfindcircles(BW,[40 95],'ObjectPolarity','dark');

% If Pupil not found, change the threshold value and iterate this 4 times
while isempty(center)&&inc<0.009
    BW=imbinarize(I,T+inc);
    [center, radii] = imfindcircles(BW,[40 95],'ObjectPolarity','dark');
    inc=inc+0.001;
end
fprintf('\n Done %02d ',i);

% Output the result of detection
if isempty(center)
    pradii(i)=[1];
    disp('NOT FOUND :(')
elseif size(radii,1)>1
    [temp,indx]=max(radii);
    pradii(i)=radii(indx);
    pcenter(:,:,i)=center(indx,:);
    disp('Found More than one pupil, Largest picked. Check the result!')
else
    pradii(i)=radii;
    pcenter(:,:,i)=center;
end
if pradii(i)==1
    NotFound=[NotFound i];
end

end
```

A-1-4 Glint Detection - Circular Hough Transform

```
function [center,radii] = glintdetect(image,startingThreshold,range,
    NumberOfIncr)

% Initialize
ThresholdIncrements=0.01;
```

```

% Check the number of times the function is called
if NumberOfIncr==0;
    center=1;
    radii=1;
    disp('_______NONE Found :( __________')
    return;
end

% Resize to increse the Glint resolution and binarize
bin=imresize(image,2);
bin=imbinarize(bin, startingThreshold);

% Circular Hough Tranformation
[center, radii] = imfindcircles(bin,range);

% If Glint not found, change the threshold value and iterate
if size(center,1)<2
    [ center, radii,NumberOfIncr]=pupildetect(image , startingThreshold - ThresholdIncrements ,range,NumberOfIncr-1);
elseif size(center,1)>2
    while size(center,1)>2
        [ center, radii,NumberOfIncr] = pupildetect(image , startingThreshold + 0.01, feature,range,NumberOfIncr-1);
        if NumberOfIncr==0
            radii=[1;1];
            center=[1,1;1,1];
        end
    end
    disp('Detection ERROR: More than one GLINT detected :O. REDUCE Threshold')
end

end
```

A-1-5 Image Coordinates to World Coordinate System

```
% Initialize the Camera Calibrated Values
t = stereoParams.TranslationOfCamera2;
R = stereoParams.RotationOfCamera2;

% Coordinate Transformation for Second Camera's Position
o2= ((bsxfun(@minus, [0 0 0], t)) * R');

% Estimate the coordinates of First Camera Image Plane
u11 = ImagePlane2World(Glint1_Center,stereoParams.CameraParameters1);
u21 = ImagePlane2World(Glint2_Center,stereoParams.CameraParameters1);
v1  = ImagePlane2World(Pupil_Center,stereoParams.CameraParameters1);

% Estimate the coordinates of Second Camera Image Plane
u12 = ImagePlane2World(Glint1_Center,stereoParams.CameraParameters2);
u12 = ((bsxfun(@minus, u12 , t)) * R');
u22 = ImagePlane2World(Glint2_Center,stereoParams.CameraParameters2);
```

```
u22 = ((bsxfun(@minus, u22 , t)) * R');
v2  = ImagePlane2World(Pupil_Center,stereoParams.CameraParameters2);
v2  = ((bsxfun(@minus, v2 , t)) * R');


function [ u ] = ImagePlane2World(ImagePoint,CameraParameters)
cppx=CameraParameters.PrincipalPoint(1)
cppy=CameraParameters.PrincipalPoint(2)
u=[(cppx+1-ImagePoint(1))*(4.8e-3),...
    (cppy+1-ImagePoint(2))*(4.8e-3),...
    -1*mean(CameraParameters.FocalLength)*(4.8e-3)];
end
```

A-1-6 Two Camera Configuration

```
function [ M2,h,c,s,p ] = TwoCamModel( o1,o2,l1,l2,u11,u21,u12,u22,v1,v2,
    K )
%TWOCAMMODEL Defines the Two camera Two Glint Geometery to solve for p
    and c

M2=[(cross((l1-o1),(u11-o1)));(cross((l1-o2),(u12-o2)));(cross((l2-o1),(
    u21-o1))); (cross((l2-o2),(u22-o2)))];
h=[dot(cross((l1-o1),(u11-o1)),o1);dot(cross((l1-o2),(u12-o2)),o2);dot(
    cross((l2-o1),(u21-o1)),o1);dot(cross((l2-o1),(u22-o2)),o2)];
c=(pinv(M2)*h)';
s=cross( cross((o1-v1),(c-o1)) ,  cross((o2-v2),(c-o2))  );
p= K * (s/norm(s)) + c;

end
```

A-1-7 LED Position - Absolute Orientation

```
function [ LED_Position, Validate_err ] = ledm( Left_frame, Right_frame,
    V_Led, stereoParams )
% Input arguments are the image frames from both camera, triangulated
% virtual LED position in world coordinates and the camera parameters
% Function outputs the estimated LED position and accuracy through
% validation.

% Undistort the image
Left_frame = undistortImage(Left_frame,stereoParams.CameraParameters1);
Right_frame = undistortImage(Right_frame,stereoParams.CameraParameters2);

% Flip for RHS coordinate system
Left_frame=fliplr(Left_frame);
Right_frame=fliplr(Right_frame);

% Pattern Detection
[imagePoints1,boardSize1,imagesUsed1] = detectCheckerboardPoints(
    Left_frame);
[imagePoints2,boardSize2,imagesUsed2] = detectCheckerboardPoints(
    Right_frame);
```

```
% Only take the Points matched in both frames, can be user input
m=[1:9];

% Triangulate the 3D Positions for each point
for i=1:size(m,2)
    worldPoints(i,:) = triangulate(imagePoints1(m(i),:),imagePoints2(m(i)
        ,:),stereoParams);
end

% NOTE: For this case, 222 to 773 pixels width of camera.
% Technical sheet says that width is 29.5mm. Each pixel is
% 29.5/(773-222) mm i.e 0.0535mm/px.
% To find the real points, we see that point 1 is 263.7,611.
% camera border is at  222 and 587 on height.
%263.7-222=41.7 or 2.2310mm and 611-587=24 or 1.2840mm

% Get the known points from the techincal datasheet
top_right=[14.75,-13.5,-(stereoParams.CameraParameters1.FocalLength(1)
    *4.8e-3)];
WCS_checkeredPoints(1,:) = top_right + [-2.2310,1.2840,0];
WCS_checkeredPoints = [WCS_checkeredPoints(1,:);
    WCS_checkeredPoints(1,:) + [0,-0.5,0];
    WCS_checkeredPoints(1,:) + [0,-1,0];
    WCS_checkeredPoints(1,:) + [0.5,0,0];
    WCS_checkeredPoints(1,:) + [0.5,-0.5,0];
    WCS_checkeredPoints(1,:) + [0.5,-1,0];
    WCS_checkeredPoints(1,:) + [1,0,0];
    WCS_checkeredPoints(1,:) + [1,-0.5,0];
    WCS_checkeredPoints(1,:) + [1,-1,0]  ];

% Absolute Orientation using the first four points
Estimated_Matrix= absor([worldPoints(1,:)',worldPoints(2,:)',worldPoints
    (3,:)',worldPoints(4,:)'],...
    [WCS_checkeredPoints(1,:)',WCS_checkeredPoints(2,:)',
        WCS_checkeredPoints(3,:)',WCS_checkeredPoints(4,:)']);

% Estimate the LED Position
LED_Position = Estimated_Matrix.R*V_Led' + Estimated_Matrix.t;

% Validate with unused points
Validate_err = norm(Estimated_Matrix.R*worldPoints(5,:)' +
    Estimated_Matrix.t) ...
    - norm(WCS_checkeredPoints(5,:)');



end
```

Appendix B

Optical Eye Model

Structure

The eye is not a perfect sphere and various from person to person. It is rather a fused two piece unit, composed of the anterior segment and the posterior segment. The anterior segment is made up of the cornea, iris and lens. The posterior segment is composed of the vitreous, retina, choroid and the outer white shell called the sclera.

Movements

Eye movements are typically divided into fixations and saccades. When the eye moves from one fixation and saccades to another, resulting series of fixations and saccades is called a scanpath.The movements of the eye are controlled by six muscles attached to each eye, and allow the eye to elevate, depress, converge, diverge and roll. These muscles are both controlled voluntarily and involuntarily to track objects and correct for simultaneous head movements. **Saccades** are quick, simultaneous movements of both eyes in the same direction controlled by the frontal lobe of the brain. Some irregular drifts, movements, smaller than a saccade and larger than a microsaccade, subtend up to one tenth of a degree. Following an object moving at constant speed is relatively easy, though the eyes will often make saccadic jerks to keep up. The smooth pursuit movement can move the eye at up to 100deg/s in adult humans. It is more difficult to visually estimate speed in low light conditions or while moving, unless there is another point of reference for determining speed.

B-1 Optical Model

There are two *principal points* for the eye. For ray-tracing and vergence equations, we can often relate everything on the object side of the system to the anterior principal point P and everything on the image side of the posterior principal point P'. There are two focal points of the eye. The posterior focal point of the eye F' is found by ray-tracing into the eye from infinity. The anterior focal point F of the eye is found by ray-tracing out of the eye, as if from infinity. There are two nodal points of the eye. A ray directed to the anterior nodal point will pass through to the retina at the same angle but as if it came from the posterior nodal point. Gaussian theory confirms that they exist even in eyes with astigmatic refracting surfaces that may or may not have collinear centres of curvature.[37]

B-1-1 Reduced Eye

The reduced eye or the single refractive surface optical model eyes is an idealized model of the optics of the human eye. The reduced eye model replaces the several refracting bodies of the eye (the cornea, lens, aqueous humor, and vitreous humor) are replaced by an ideal air/water interface surface that is located 20 mm from a model retina. The converts a system with six cardinal points (two focal points, two principal points and two nodal points) into one with three cardinal points (two focal points and one nodal point). Differences in few of the proposed models are shown in Table B-1

Table B-1: Reduced Eye Models

	Emsely's	Listing's	Donder's
Principal Point P	+0.0mm	+1.5mm	+2.0mm
Nodal Point N	+5.55mm	+7.2mm	+7.0mm
First focal point F	-16.67mm	-15.7mm	-13.0mm
Second focal point F'	+22.22mm	+22.9mm	+22.0mm
Refractive Index	4/3	4/3	4/3
Radius of Equivalent Surface	5.55mm	5.73mm	5.00mm

B-1-2 Simplified Eye

A better model to the reduced model is the simplified model, with three refracting surfaces, one for the cornea and two for the lens. As aperture stop is placed in an anatomically correct position at the front of the lens, accommodated forms can be provided and the cardinal points can be accurately placed. An example for this is the modified Gullstrand called the Gullstrand-LeGrand Eye.

B-1-3 Four Refracting Surfaces

These models have two corneal and two lens refracting surfaces. A good example is Le Grand's full theoretical eye, which comes in relaxed and 7.1 D accommodated forms. From such models, 'adaptive' optical model eyes have been developed, with equations showing how parameters vary with accommodation and age.

Appendix C

Geometric Transformations

The combination types of transformations (translation, reflection, rotation and dilation) is explained as follows:

Isometric Transformation : Transformation of this type preserves Euclidean distance. With 3 Degree of Freedom. Isometery transformation for any point $\mathbf{p}$ can be written as:

$$\mathbf{p}' = \begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix} \mathbf{p} \tag{C-1}$$

where, R is rotation matrix, t is translation matrix and 0^T is a row of two zeros

Similarity Transformation : Similarity Transformation is an isometric transformation in addition with isotropic scaling. Which implies that the scaling is invariant with respect to direction, adding a degree of freedom to a total of 4 degrees of freedom. Distances do not remain the same but the ration between the distances is constant. Similarity transformation for a point $\mathbf{p}$ can be expressed as:

$$\mathbf{p}' = \begin{pmatrix} sR & t \\ 0^T & 1 \end{pmatrix} \mathbf{p} \tag{C-2}$$

where, s is the isotropic scaling factor

Affine Transformation : An affine transformation is like a similarity transform but instead of a single rotation and isotropic scaling it is a composition of two rotations and two non-isotropic scalings. It contains two more degrees of freedom than the similarity transformation.Unlike the similarity transformation, an affine transformation does not preserve the distance ratios or the angles between lines. There still are some invariants though, as parallel lines in one image remain parallel in the mapped image, and the ratios of lengths of parallel line segments and areas are preserved. An affine transformation can be written as:

$$\mathbf{p}' = \begin{pmatrix} A & t \\ 0^T & 1 \end{pmatrix} \mathbf{p} \tag{C-3}$$

where, A is non-singular matrix and decomposed as:

$$A = R(\theta)R(-\phi)DR(\phi)$$

with

$$D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

$R(\cdot)$ as the Rotation matrix and λ_1 and λ_2 as the two scaling values. The matrix A is the concatenation of a rotation by ϕ, scaling with λ_1 and λ_2 in x and y direction respectively, a rotation back by $-\phi$ and then another rotation by θ

Projective Transformation or Homography : Terminology for this transformation include *homography*, *collineation*, *projectivity* and *planar projective transformation*. The projective transformation is a non-singular linear transformation of homogeneous coordinates. This transformation would be non-linear with in-homogeneous coordinates and this is what makes the use of homogeneous coordinates so valuable. Projective transformations contain two more degrees of freedom than affine transformations as now the matrix has nine elements with only their ratio significant. A projective transformation can be written as:

$$\mathbf{p}' = \begin{pmatrix} A & t \\ \nu^T & \upsilon \end{pmatrix} \mathbf{p} \tag{C-4}$$

where, $\nu = (\upsilon_1, \upsilon_2)^T$ The vector ν is responsible for the non-linear effects of the projectivity. For affnities, the scalings from A are the same everywhere in the plane, while for projectivities scaling varies with the position in the image. Projective Transformation can be decomposed into:

$$H = H_S H_A H_P = \begin{pmatrix} sR & t \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} U & 0 \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} I & 0 \\ \nu^T & \upsilon \end{pmatrix} = \begin{pmatrix} A & t \\ \nu^T & \upsilon \end{pmatrix}$$

H_S represents a similarity transformation, H_A represents an affinity and H_P represents a projectivity. $A = sRU + t\nu^T$and U is an upper triangular matrix normalized as $det(U) = 1$. For this decomposition to be valid, ν cannot equal 0.

Bibliography

[1] E. S. Gragoudas, J. M. Seddon, K. Egan, R. Glynn, J. Munzenrider, M. Austin-Seymour, M. Goitein, L. Verhey, M. Urie, and A. Koehler, "Long-term results of proton beam irradiated uveal melanomas," *Ophthalmology*, vol. 94, no. 4, pp. 349 – 353, 1987.

[2] A. Królak and P. Strumiłło, "Eye-blink detection system for human–computer interaction," *Universal Access in the Information Society*, vol. 11, pp. 409–419, Nov 2012.

[3] V. Raudonis, R. Simutis, and G. Narvydas, "Discrete eye tracking for medical applications," in *2009 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, pp. 1–6, Nov 2009.

[4] R. G. Lupu, R. G. Bozomitu, F. Ungureanu, and V. Cehan, "Eye tracking based communication system for patient with major neoro-locomotor disabilites," in *15th International Conference on System Theory, Control and Computing*, pp. 1–5, Oct 2011.

[5] H. Huang, Y.-S. Zhou, F. Zhang, and F.-C. Liu, "An optimized eye locating and tracking system for driver fatigue monitoring," in *2007 International Conference on Wavelet Analysis and Pattern Recognition*, vol. 3, pp. 1144–1149, Nov 2007.

[6] W.-B. Horng, C.-Y. Chen, Y. Chang, and C.-H. Fan, "Driver fatigue detection based on eye tracking and dynamk, template matching," in *IEEE International Conference on Networking, Sensing and Control, 2004*, vol. 1, pp. 7–12, March 2004.

[7] R. C. Coetzer and G. P. Hancke, "Eye detection for a real-time vehicle driver fatigue monitoring system," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 66–71, June 2011.

[8] W. Porterfield, "An essay concerning the motions of our eyes. part i. of their external motions," *Edinburgh Medical Essays and Observations*, vol. 3, pp. 160–263, 1737. cited By 30.

[9] A. L. Yarbus, *Eye Movements During Perception of Complex Objects*, pp. 171–211. Boston, MA: Springer US, 1967.

[10] R. V. Kenyon, "A soft contact lens search coil for measuring eye movements," *Vision Research*, vol. 25, no. 11, pp. 1629 – 1633, 1985.

[11] J. Tang and J. Zhang, "Eye tracking based on grey prediction," in *2009 First International Workshop on Education Technology and Computer Science*, vol. 2, pp. 861–864, March 2009.

[12] H. Liu and Q. Liu, "Robust real-time eye detection and tracking for rotated facial images under complex conditions," in *2010 Sixth International Conference on Natural Computation*, vol. 4, pp. 2028–2034, Aug 2010.

[13] J. Gips, P. DiMattia, F. X. Curran, and P. Olivieri, "Using eagleeyes—an electrodes based device for controlling the computer with your eyes—to help people with special needs," in *Proceedings of the 5th International Conference on Computers Helping People with Special Needs. Part I*, ICCHP '96, (Munich, Germany, Germany), pp. 77–83, R. Oldenbourg Verlag GmbH, 1996.

[14] Y. L. Kuo, J. S. Lee, and S. T. Kao, "Eye tracking in visible environment," in *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 114–117, Sept 2009.

[15] B. Fu and R. Yang, "Display control based on eye gaze estimation," in *2011 4th International Congress on Image and Signal Processing*, vol. 1, pp. 399–403, Oct 2011.

[16] X. Li and W. G. Wee, "An efficient method for eye tracking and eye-gazed fov estimation," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 2597–2600, Nov 2009.

[17] Y. W. Chen and K. Kubo, "A robust eye detection and tracking technique using gabor filters," in *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, vol. 1, pp. 109–112, Nov 2007.

[18] T. Kocejko, A. Bujnowski, and J. Wtorek, "Eye mouse for disabled," in *2008 Conference on Human System Interactions*, pp. 199–202, May 2008.

[19] A. Doshi and M. Trivedi, "Investigating the relationships between gaze patterns, dynamic vehicle surround analysis, and driver intentions," in *2009 IEEE Intelligent Vehicles Symposium*, pp. 887–892, June 2009.

[20] B. Wandell, "Chapter 3: The photoreceptor mosaic."

[21] M. Nowakowski, M. Sheehan, D. Neal, and A. V. Goncharov, "Investigation of the isoplanatic patch and wavefront aberration along the pupillary axis compared to the line of sight in the eye," *Biomed. Opt. Express*, vol. 3, pp. 240–258, Feb 2012.

[22] E. D. Guestrin and M. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," *IEEE Transactions on Biomedical Engineering*, vol. 53, pp. 1124–1133, June 2006.

[23] Wikipedia, "Pinhole camera model — wikipedia, the free encyclopedia," 2017. [Online; accessed 31-October-2017].

[24] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientation," 02 1999.

[25] O. Faugeras, *Three-dimensional Computer Vision: A Geometric Viewpoint.* Cambridge, MA, USA: MIT Press, 1993.

[26] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 1330–1334, Nov. 2000.

[27] OSRAM, "Eye safety of ireds used in lamp applications."

[28] D.Kandepet, "Deconstructing a flashlight."

[29] A. Geiger, F. Moosmann, . Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *2012 IEEE International Conference on Robotics and Automation*, pp. 3936–3943, May 2012.

[30] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Am. A*, vol. 4, pp. 629–642, Apr 1987.

[31] T. J. Atherton and D. J. Kerbyson, "Size invariant circle detection," 1999.

[32] 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. CVPR 2001*, vol. 1, pp. I–511–I–518 vol.1, 2001.

[33] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International Conference on Computer Vision*, pp. 2548–2555, Nov 2011.

[34] H. Bay, T. Tuytelaars, and L. Van Gool, *SURF: Speeded Up Robust Features*, pp. 404–417. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[35] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2*, ICCV '05, (Washington, DC, USA), pp. 1508–1515, IEEE Computer Society, 2005.

[36] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761 – 767, 2004. British Machine Vision Computing 2002.

[37] D. A. Atchison and L. N. Thibos, "Optical models of the human eye," *Clinical and Experimental Optometry*, vol. 99, no. 2, pp. 99–106, 2016. Atchison15-161.

Glossary

List of Acronyms

PT	Proton Therapy
PoG	Point of Gaze
DNA	Deoxyribonucleic acid
DOF	Degree of Freedom
VOG	Video-oculography
FOV	Field of view
GUI	Graphical User Interface
CAD	Computer Aided Design
CCS	Camera Coordinate System
MRI	Magnetic Resonance Imaging
LED	Light Emitting Diode
LHS	Left-Hand Side
RHS	Right-Hand Side
GPIO	General Purpose Input-Output
GPU	Graphical Processing Unit
HCI	Human-Computer Interaction
WCS	World Coordinate System
CHT	Circular Hough Transform
NIR	Near-Infrared
EOG	Electrooculography