A Non-Contact Video-oculograph For Tracking Gaze in a Human Computer Interface

by

Borna Noureddin
B.Eng., University of Victoria, 1994

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
Master of Applied Science

in
THE FACULTY OF GRADUATE STUDIES

(Department of Electrical and Computer Engineering)

We accept this thesis as conforming to the required standard

The University of British Columbia
March 2003
© Borna Noureddin, 2003
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature)

Department of **Electrical and Computer Engineering**

The University of British Columbia
Vancouver, Canada

Date **23 April 2003**
Abstract

Video-based eye tracking devices that can detect where a person is looking without requiring the user to wear anything can be effective components of a human computer interface. However, issues such as speed, accuracy, cost and ease of use have so far limited the widespread, practical application of existing devices. No pre-existing device that does not require user contact is able to accurately track the eye (and hence where a person is looking) in real time in the presence of large head movements.

This thesis attempts to overcome these limitations by presenting a novel design for a video-oculograph. It is a non-contact (i.e., it does not require the user to wear anything) human computer interface device, and uses two cameras to maintain accurate, real-time tracking of a person’s eye in the presence of significant head motion. Image analysis techniques are used to obtain very accurate locations of the pupil and corneal reflection. All the computations are performed in software and the device only requires simple, compact optics and electronics attached to the user’s computer via a serial port and IEEE 1394 interface, making the device very cost effective.

An implementation of the design to track a user’s fixations on a computer monitor is also evaluated in this thesis. Two methods of estimating the user’s point of gaze on a computer monitor were evaluated. Using functional approximation, the gaze was estimated to within 5.2% of the monitor’s width and 9.6% of the monitor’s height. Using a direct, analytical approach, the gaze was estimated to within 22.0% horizontally and 27.8% vertically. Evidence was found to support further investigation into effective
calibration procedures that would significantly improve the ability of the system to estimate the user's point of gaze.

The implemented system – called GTD (Gaze Tracking Device) – is capable of reliably tracking the user's eye in real-time (nine frames per second) in the presence of natural head movements as fast as 100°/s horizontally and 77°/s vertically. It is able to track the location of the eye to within 0.758 pixels horizontally and 0.492 pixels vertically, and is robust to changes in eye colour and shape, ambient lighting and the use of eyeglasses.
Contents

Abstract .......................................................................................................................... ii

Contents ......................................................................................................................... iv

List of Tables .................................................................................................................. vii

List of Figures .................................................................................................................. viii

Acknowledgments ......................................................................................................... xi

1 Background ................................................................................................................ 1
  1.1 Introduction ............................................................................................................ 1
  1.2 Physiology of eye movements ............................................................................. 3
  1.3 Video-oculography ............................................................................................... 10
  1.4 Overview of GTD .................................................................................................. 15

2 Previous Work ............................................................................................................ 18
  2.1 Non-video methods ............................................................................................ 19
  2.2 VOG-based systems requiring user contact ....................................................... 20
  2.3 Feature detection and parameterization ............................................................. 22
  2.4 Large head movements and high resolution ....................................................... 25
  2.5 Complete VOG-based devices ........................................................................... 29
      2.5.1 Camera calibration ....................................................................................... 34

3 System Design ........................................................................................................... 37
  3.1 Assumptions and overall objectives .................................................................... 37
  3.2 Tracking approach ............................................................................................... 38
      3.2.1 Image capture ............................................................................................... 39
      3.2.2 Head movement compensation .................................................................... 40
  3.3 Eye parameterization ............................................................................................ 41
      3.3.1 Pupil and glint detection .............................................................................. 41
      3.3.2 Pupil and glint centre calculation ............................................................... 44
  3.4 Gaze calculation .................................................................................................... 45
      3.4.1 Calculation of angle of gaze ....................................................................... 46
      3.4.2 Calculation of point of gaze ....................................................................... 55
  3.5 Calibration ............................................................................................................. 59
4 System Implementation ........................................................................................................ 65
  4.1 Hardware overview ........................................................................................................ 66
  4.2 Software overview ........................................................................................................ 69
  4.3 Mechanics ....................................................................................................................... 71
    4.3.1 Pan/tilt motors ......................................................................................................... 72
    4.3.2 Angular Position sensors ........................................................................................ 73
  4.4 Optics .............................................................................................................................. 74
    4.4.1 Cameras .................................................................................................................... 74
    4.4.2 Lenses and filters .................................................................................................... 76
  4.5 Electronics ...................................................................................................................... 78
    4.5.1 Infrared LED circuit ............................................................................................... 78
    4.5.2 Control and interface circuit .................................................................................. 80
  4.6 Pupil and glint tracking ................................................................................................. 82
    4.6.1 Pre-processing and image differencing .................................................................... 83
    4.6.2 Thresholding ........................................................................................................... 86
    4.6.3 Selecting the pupil blob .......................................................................................... 89
    4.6.4 Finding the pupil center .......................................................................................... 90
    4.6.5 Spatio-temporal filtering ........................................................................................ 90
    4.6.6 Finding the glint ...................................................................................................... 93
  4.7 Gaze calculation .............................................................................................................. 94
    4.7.1 Description of parameters ..................................................................................... 96
    4.7.2 Direct versus approximated approaches ................................................................. 98
  4.8 Camera calibration .......................................................................................................... 99

5 Results .................................................................................................................................. 105
  5.1 Hardware ........................................................................................................................ 105
  5.2 Detection and parameterization ..................................................................................... 107
  5.3 Gaze calculation .............................................................................................................. 117
  5.4 Calibration ...................................................................................................................... 129
    5.4.1 Camera calibration .................................................................................................. 129
    5.4.2 Complete calibration .............................................................................................. 131

6 Conclusions and Future Work ........................................................................................... 134
  6.1 Analysis of results .......................................................................................................... 134
  6.2 Future work and applications ......................................................................................... 137

References ................................................................................................................................ 141

Appendix A - Electronic Circuits .......................................................................................... 147
  Infrared LED circuit ............................................................................................................ 147
List of Tables

Table 5.1 Comparison of performance under different conditions for various subjects
(Pupil in NA camera) ........................................................................................................... 115
Table 5.2 Comparison of performance under different conditions for various subjects
(Pupil in WA camera) ........................................................................................................... 115
Table 5.3 Comparison of performance under different conditions for various subjects
(Glint in NA camera) ........................................................................................................... 115
Table 5.4 Datasets used to train and test SPORE approximation ....................................... 121
Table 5.5 Results of training SPORE approximation with a single dataset ......................... 125
Table 5.6 Results of training SPORE approximation with two and three datasets .......... 125
Table 5.7 Comparison of training SPORE with different grid sizes .................................... 126
Table 5.8 Comparison of training SPORE with different orientation angles ....................... 126
Table 5.9 Gaze errors for multiple datasets using 3x3 grid ................................................ 127
Table 5.10 Gaze errors for different grid sizes .................................................................. 127
Table 5.11 Gaze errors for different mirror and frame orientations ..................................... 127
Table 5.12 Results of different system calibration approaches ......................................... 131
Table E.1 Resolution and accuracy .................................................................................. 173
List of Figures

Figure 1.1 Cross section of the human eye ................................................................. 5
Figure 1.2 Formation of Purkinje images ................................................................. 6
Figure 1.3 Angle of gaze ............................................................................................ 9
Figure 1.4 Model-based tracking in a complete VOG-based system ....................... 11
Figure 1.5 GTD Eyetracking System ........................................................................ 16
Figure 1.6 Photograph of GTD eye tracker in action ................................................ 17
Figure 3.1 Difference image with motion artefacts ..................................................... 43
Figure 3.2 Gullstrand model of the eye ..................................................................... 47
Figure 3.3 Geometry of gaze calculation from pupil and glint images in NA camera .... 49
Figure 3.4 Calculation of 3-D pupil location from two cameras ................................. 50
Figure 3.5 Pinhole camera model ............................................................................ 52
Figure 3.6 Using LED array as multiple light sources ................................................. 55
Figure 3.7 Example of SPORE model structure ....................................................... 57
Figure 3.8 Camera model ....................................................................................... 60
Figure 4.1 Overview of system components .............................................................. 67
Figure 4.2 Overview of system software .................................................................. 70
Figure 4.3 Operation of infrared LED circuit ............................................................ 79
Figure 4.4 Synchronized camera triggering and LED control ..................................... 80
Figure 4.5 Unsynchronized camera triggering and LED control ............................... 81
Figure 4.6 Operation of the control and interface circuit .......................................... 82
Figure 4.7 Raw image from NA camera of dark pupil ................................................ 84
Figure 4.8 Noise removed from image of dark pupil ............................................... 84
Figure 4.9 Image from NA camera of bright pupil ....................................................... 85
Figure 4.10 Difference image for a single frame ......................................................... 85
Figure 4.11 Aggregate difference image for three frames ........................................... 85
Figure 4.12 Image from WA camera of dark pupil ..................................................... 87
Figure 4.13 Image from WA camera with LoG operator applied ................................. 87
Figure 4.14 Blobs from WA camera image ................................................................. 87
Figure 4.15 Median filtered blob image from WA camera ........................................... 87
Figure 4.16 Dilated WA camera blob image ............................................................... 88
Figure 4.17 Eroded WA camera blob image .............................................................. 88
Figure 4.18 Thresholded aggregate image from NA camera ....................................... 88
Figure 4.19 Dilated image from NA camera ............................................................... 89
Figure 4.20 Eroded image from NA camera ............................................................... 89
Figure 4.21 Spatio-temporal filtering algorithm ......................................................... 92
Figure 4.22 Overview of gaze calculation using SPORE approximation .................... 94
Figure 4.23 Overview of direct gaze calculation ......................................................... 95
Figure 4.24 Undistorting image data ....................................................................... 100
Figure 4.25 Actual images used for WA camera calibration ....................................... 101
Figure 4.26 Checkerboard pattern used for WA camera calibration ............................ 102
Figure 5.1 Synthetic images for pupil and glint in NA camera ..................................... 108
Figure 5.2 Synthetic images for pupil and glint in WA camera .................................... 109
Figure 5.3 Sample images from Subject #1 without sunlight ..................................... 111
Figure 5.4 Sample images from Subject #1 with sunlight ......................................... 111
Figure 5.5 Sample images from Subject #2 .............................................................. 112
Figure 5.6 Sample images from Subject #3 ........................................... 112
Figure 5.7 5x5 grid of points used for SPORE training and testing ............... 117
Figure 5.8 Formation of data sets for training and testing SPORE approximation using training grid points ....................................................... 118
Figure 5.9 Formation of data set for testing SPORE approximation using off-grid points .......................................................................... 119
Figure 5.10 Sample images of three trials using 5x5 grid .................................. 120
Figure 5.11 9x9 grid of points used for SPORE training and testing ............... 120
Figure A.1 Circuit diagram for infrared LED circuit ..................................... 151
Figure A.2 Circuit diagram for control and interface electronics ...................... 152
Figure B.1 Front view of GTD eye tracker .................................................. 154
Figure B.2 Side view of GTD eye tracker .................................................... 155
Figure B.3 Top view of GTD eye tracker .................................................... 156
Figure C.1 Rotation speed requirements ..................................................... 158
Figure C.2 Angular resolution of motors .................................................... 165
Figure D.1 Photograph of DragonFly camera ............................................. 168
Figure D.2 Location of GPIO pins on DragonFly cameras ............................ 169
Figure D.3 Layout of GPIO pins on DragonFly cameras .............................. 170
Figure E.1 Resolution and accuracy .......................................................... 172
Figure E.2 Geometrical relationship of pupil and orientation of NA camera ...... 174
Figure E.3 Ray diagram for computing location of image of pupil ................. 183
Figure E.4 Calculation of corneal center using multiple light sources .......... 187
Acknowledgments

I would like to first thank my supervisor Dr. Peter Lawrence for his guidance and support. His willingness and ability to allow me to explore research questions using my own creativity while providing me with moral, financial and practical support when most needed is greatly appreciated.

I would also like to acknowledge the support and friendship of several individuals in the Robotics and Control Laboratory (Department of Electrical and Computer Engineering at the University of British Columbia), who not only encouraged me when I needed it, but also provided much needed technical assistance.

This work was funded by two NSERC Grant Programs: A Discovery Grant and an NCE IRIS Research grant.

My work would not have been possible without the constant, loving support of my family, especially my dear wife and daughter who sacrificed a great deal during a critical time of our life together so that I could pursue my research. To them I owe a debt of gratitude beyond words, and I pray that I have made some contribution that is worth their patience.

Borna Noureddin
March 2003
1 Background

1.1 Introduction

This thesis presents a novel design for a human computer interface device that tracks a person’s eye in real time, and that can be used to track and record where a person is looking. An implementation of that design to track a user’s fixations on a computer monitor is also evaluated.

Tracking and recording a person’s eyes has been shown to be useful in medical diagnosis [1], oculomotor and neurological research [1][2][3], assistive devices for people with disabilities [4][5], facial expression analysis, driver awareness systems [6], face recognition, pilot training, visual attention studies, marketing and many other applications.

Recent efforts have also included investigating the use of eye tracking devices in human computer interfaces. Yang and Zhang [7] propose the use of an eye tracker to improve effectiveness of a video-conferencing system. Talmi and Liu [8] report the use of their system to improve the realism of autostereoscopic displays. Heinzmann and Zelinsky [9] have investigated the use of an eye tracker to make robots that are more “human-friendly.”

Sibert et. al. [10] compared an eye tracker with a mouse to determine the former’s usability as a user interface device. They determined that their eye tracker provides faster indication than a mouse and implicitly indicates focus of attention (which can be used to provide a richer user experience). They showed that the advantage of eye selection increases with the distance an object needs to be moved by the user. They also argue that eye selection is impractical for precision positioning tasks. Their studies seem to indicate
that an eye tracking device would be very effective when combined with other input channels.

Despite the number of possible applications and apparent effectiveness of eye tracking devices in a human computer interface, issues such as speed, accuracy, cost and ease of use have so far limited the widespread, practical application of existing devices beyond narrow research markets. The research reported in this thesis attempts to overcome these limitations.

Chapter 2 provides a review of the literature. Aside from a broad overview of eye tracking technologies, a brief analysis of each systems’ performance and suitability for a human computer interface is provided. The main research questions are also raised in this chapter.

Chapter 3 describes the design of the video-oculograph. In addition to the major assumptions and overall objectives of the research, a theoretical framework is presented for tracking the user’s eye in the presence of head movement and environmental changes and for calculating the user’s point of gaze. Section 3.5 also provides some minimal theory required to understand the calibration requirements of the system.

The implementation of the system is presented in Chapter 4. The hardware, software, mechanical and optical components of the constructed system are described in some detail, and the algorithm used to track the user’s eye is provided. Methods for calculating the point of gaze that have been implemented follow, along with various calibration procedures.

Experiments used to assess the performance of the implemented system, its associated methods for calculating point of gaze and procedures for calibrating system parameters
are described in Chapter 5, along with the results obtained from carrying out those experiments.

Finally, Chapter 6 provides an analysis of the performance of the eye tracker, along with major conclusions, additional improvements to the system that could be made, and worthwhile areas of future research.

The remaining sections of the current chapter provide some background information about the physiology of eye movements, a brief introduction to video-oculography, and an overview of the implemented system. The following section on the physiology of eye movements briefly covers topics that are helpful for understanding the main concepts behind constructing an effective eye tracking device.

1.2 Physiology of eye movements

Understanding some basic human physiology related to eye and head movements is important for designing both an effective eye tracking device and a user interface that makes use of that device. This section attempts to give a very brief introduction to the anatomy of the eye and a few major physiological features of the human eye that must be taken into account when designing an eye tracking device. In addition, it describes the role of head movements in establishing and maintaining gaze – that is, where a person is looking – and outlines the types of eye movements that could be measured. In the context of this thesis, gaze can be the angle of gaze (AOG), which is the direction in which a person is looking (see Figure 1.3), or the point of gaze (POG), which is the point in space on which a person is fixating, or both. For a more comprehensive presentation of the physiology and neurology of eye movements, see [1], [11] and [12].
Figure 1.1 shows a cross section of the human eye. Light enters through the cornea, which acts as a focusing surface. The light then travels through a clear aqueous fluid (known as the aqueous humor) and passes through the pupil, which is a small aperture whose size is changed by muscles in the iris to control the amount of light entering the eye. The lens then focuses the light rays, which then proceed through a gelatinous substance called the vitreous humour. The focused light rays project onto the retina, which consists of various photoreceptors (known as rods and cones) that record and transmit the "image" produced by the light rays to the brain via the optic nerve. The fovea is an area of the retina with a high concentration of nerve endings, and is the part of the image that is perceived with the highest clarity. It represents roughly 1° of the visual field.

Of particular importance to video-based eye trackers is the cornea, because its shape needs to be modeled correctly for an accurate measure of the user's angle of gaze. It is also worth noting here that physiological defects and disorders such as astigmatism could also affect the performance of a video-based eye tracking device.

The most prominent parts or "features" of the eye in the image of an eye as viewed by a camera are the sclera (which is a white area that surrounds the eyeball except for the area covered by the cornea), the iris and the pupil. The iris gives the eye its "colour" and is seen as a coloured circular annulus when viewed from directly in front of the eye. Parts of the iris can often be occluded by eyelids and eyelashes. The boundary where the cornea meets the sclera is known as the limbus, and is often hard to distinguish from the boundary between the iris and the sclera in an image. The pupil appears as a dark circle in the centre of the iris, and its radius varies as it opens and closes. In certain structured...
lighting conditions (see Section 3.3.1), the pupil may appear as a bright circle. Unlike the iris, it is rarely occluded.

Figure 1.1 Cross section of the human eye

1 This diagram is adapted from Figure 3.1 on page 64 of "Introduction to the Optics of the Eye" by David A. Goss and Roger W. West (see [12]).
Both the cornea and the lens act as optical surfaces. Both the front and back surfaces of each act as curved reflective and refractive surfaces. Their purpose is to help focus incoming light rays on the retina. However, they have the added effect of transmitting the reflection of the same light rays from the retina back out of the eye. Figure 1.2 shows how incoming light that travels through the cornea and lens is also refracted at each surface to produce a "Purkinje image". Purkinje images are, in a sense, reflections of an object (like a light source) off the eye. The first Purkinje image is by far the brightest. Several eye tracking devices (including the one presented in this thesis) use the first Purkinje image of a known light source. Since this is the reflection off the outer surface of the cornea, it is often referred to as the corneal reflection, corneal reflex or glint.

![Diagram of Purkinje images](image)

**Figure 1.2 Formation of Purkinje images**

---


3 This diagram is adapted from Figure A1.7 on page 316 of "Movements of the Eyes" by R.H.S. Carpenter (see [11]).
A good eye tracking device must also take into account head movements. Head movements are used both to stabilize the head and eyes and to shift gaze toward a visual target. Even when a person intends to keep their head stationary, the head’s eccentric carriage on the neck joints makes it susceptible to slight oscillations (especially in pitch). Also, as a person shifts weight or performs any other movement involving the trunk of the body, the head must be moved to compensate if a visual target is to be kept stable on the retina.

In addition, human beings naturally combine head and eye movements in order to perform a rapid gaze shift, especially ones that require a change of more than 15° in the angle of gaze. Such gaze shifts generally bring the image of an object of interest in the visual periphery to the fovea.

Generally, the angular velocity of a head movement can be as high as 100°/s (e.g., when a person is running). This is important to note for a remote video-based eye tracking device that must be able to compensate for such rapid head movements.

In addition to head movements, the rotation of the eye in its socket also plays a significant role in both stabilizing an image on the retina and shifting the user’s angle of gaze. The goal of stabilization is to counteract “self-motion” (movement of the head or body of the person), and keep the line of sight constant with respect to the visual environment. The opto-kinetic system is a neurological system that maintains continuous, clear vision of objects of interest that are moving up to 5°/s within 0.5° of the centre of the fovea. During fixation on a visual target, small eye movements counter oculomotor noise (i.e., noise associated with neural impulses used to control the eye muscles). These consist of high-frequency, low amplitude (less than 0.01°) tremors,
microsaccades that are less than 0.1° in amplitude and can be suppressed during visual
tasks (like threading a needle), and slow drifts, which typically have an average velocity
of 0.25°/s and a standard deviation of 0.1°.

Gaze shifts are achieved using one or more of three types of eye movements.
Smooth pursuit eye movements follow a moving object or assist in fixation on a near
stationary target during self-motion. Saccades are very rapid (up to 800°/s) eye
movements used to quickly align the line of sight with a visual target. Saccades are
ballistic, meaning that once they are initiated neurologically, they are pre-programmed to
follow a certain trajectory. In addition, since humans have stereo vision, vergence eye
movements are used to maintain binocular convergence for maximum visual clarity and
focus.

While the ability to track all types of eye movements is desirable in a human
computer interface, what is perhaps most useful is the ability to detect and track fixations.
For the purposes of the discussion in this thesis, a fixation is defined as a period where
only tremors, microsaccades and slow drifts take place over at least 350ms. The 3-D
location of the object that falls within the foveal centre during this period is considered to
be the point of gaze. The angle of gaze is more difficult to define. It is the angle
between the line of sight (a line that connects the point of gaze with the centre of the
pupil) and some other reference axis. Figure 1.3 illustrates two definitions of angle of
gaze that seem to be the most common in the eye tracking literature (though most of the
published work does not explicitly define angle of gaze). One uses an axis that is
perpendicular to the frontal plane of the person, and the other uses an axis that is
perpendicular to the viewing plane on which the point of gaze lies. For the system
described in this thesis, the latter definition is used — that is, the angle of gaze is formed by the line of sight and the line that is perpendicular to the computer monitor and passes through the centre of the eye ($\theta_2$ in Figure 1.3). Clearly, when the frontal plane and the viewing plane are parallel, the definitions are the same.

![Diagram of angle of gaze](image)

**Figure 1.3 Angle of gaze**

This section described some of the anatomical and physiological aspects of the eye necessary for understanding the next section, which introduces the most common types of devices that currently exist for tracking and recording eye movements.
1.3 Video-oculography

There are three (3) main techniques commonly used for tracking and recording eye movements. For a good review of the basic principles of these techniques, refer to [11] and [13]. Electrooculography (EOG) involves placing electrodes on the skin and measuring the difference in electric potential caused by the movement of the six eye muscles. The magnetic search coil method is a very accurate one that involves placing contact lenses on the eyes with wires attached, and using large magnets placed around the subject’s head to measure the change in the 3-D position of the contact lens, and hence the eye. The third technique, which is either minimally intrusive (as in head mounted systems) or completely non-contact, is known as video-oculography (VOG) [14].

VOG involves a computer that extracts and tracks information about one or more features using video signals (i.e., a series of sequential images) received from one or more head-mounted or remote cameras. In this context, a feature is some property of a person’s eye (such as the centre of the iris, the corners of the eye or the reflection of a light source off the surface of the eye) or other part of the face (such as the corners of the lips or the tip of the nose). As a technique for tracking and recording eye movements, it is important that at least one of the features be a part of the eye. The extracted information is then often used to determine the person’s gaze. The work presented in this thesis uses VOG to track the eyes and determine a user’s gaze. Thus, the remainder of this section describes in some detail the main tasks that devices that use VOG to determine a user’s gaze must perform.
In a complete VOG system that determines where a user is looking, there are three basic tasks. Figure 1.4 outlines these three tasks, each of which is described in some detail below.

**Task 1**
- Compensate for head motion

**Task 2**
- Detect and extract parameters from features

**Task 3**
- Use parameters to calculate AOG/POG

Figure 1.4 Model-based tracking in a complete VOG-based system

**Task 1: Head motion compensation**

The first task is to provide some mechanism for compensating for natural head movements to ensure that the user’s eye is always in the field of view of the camera(s) tracking the eye. Some systems simply use very wide angle lenses, thus maximizing the space over which the head can move while maintaining the user’s eye within the field of view of the camera(s). The disadvantage of this approach is that the wider the field of view of a lens, the lower the resolution that can be detected by the camera on which it is mounted. Other systems use a zoom or “narrow angle” lens to maintain high resolution, and use mirrors or special tilt and pan equipment attached to the camera. The effort for achieving this first step is minimized - though not completely eliminated - with head-mounted systems. However, in the context of a human computer interface, non-contact systems are highly preferable to those that require the user to wear the device.
**Task 2: Feature detection and parameterization**

A VOG system used to determine a person's gaze employs a model to represent the features being tracked, and uses the parameters of that model to determine the gaze. For example, in some systems, model parameters that together describe both the orientation of the eye in the head and the position and orientation of the head in space is used to completely determine the gaze. The values of the model parameters are determined by the information extracted from the features being tracked. This information consists of measurements of the features, such as their location. Thus, the second task is to detect the features in each image and extract the associated parameters used by the model. For example, many systems track the location of the pupil and the glint. This method allows the VOG system to differentiate between changes in eye position caused by eye movement and those caused by head movement (see [13] for a more detailed discussion).

Effective methods used to perform this task must be able to handle blinks, occlusion, the use of eyeglasses by the subject, differences in facial expression, eye colour, shape and size. In a typical VOG system, this extracting of information – or parameterization – often employs image analysis techniques. Numerous approaches have been used to implement this step, and each of these approaches usually attempts to determine (a) the most likely candidate for the feature(s) being detected and (b) the values for the model parameters that will be used by the system to estimate the user's gaze.

To determine the most likely candidate, methods such as deformable templates [15][16][17], snakes [18] and eigen-features (principal component analysis) [19][20] build a saliency map based on a statistical fit of each point in the input image to the template or model [21]. Least-squared error, maximum likelihood, sum of absolute value
of differences, correlation coefficient, and cross-correlation are some of the statistical
methods employed in such cases [22]. One of the major disadvantages of the above
methods is that the templates or models need to be placed manually close to the actual
location of the feature (e.g., the center of the iris) to work properly. Other methods (such
as the image difference method [23]) apply some image processing algorithm such as
thresholding to find the best candidate directly, and without as much need for manual
setup. In all cases, some image pre-processing is often used to speed up the search.

In a real-time tracking system, the main computational cost for this search for the best
candidate is typically at start-up or when the tracking algorithm has to be reset. The rest
of the time, some efficient algorithm is used to track the desired feature(s) from one
frame to the next.

In order to determine an appropriate model, some a priori knowledge is required
(e.g., that the pupil usually appears as a circle in an image). Most methods then employ a
heuristic or empirically derived set of values to parameterize the model. In all cases,
various image analysis or computer vision techniques such as edge detection, filtering,
contour following or segmentation (see [24]) are used to transform the 2-D pixel data to a
form where suitable model parameter values can ultimately be derived. The choice of an
appropriate model and its associated parameters is often determined by the application of
the VOG system. In a computer interface that makes use of a user’s point of gaze, the
model parameters typically consist of measurements of one or more features of the eye
and possibly face. The system then maps the parameter values to a location in 3-D space
representing the user’s point of gaze. For example, several systems calculate the pupil
center in an image. They calculate either the center of mass of the pixels in the image
representing the pupil or fit a circle to the pupil and find its center. In the latter case, the feature being tracked is a pupil, which is modeled as a circle whose parameter used to calculate gaze is the center of the circle. The value of the parameter is calculated by finding the edges of the pupil in the image, finding the circle that best fits the edge points, and calculating the center of that circle. Note that using the center of mass can be sensitive to pixel noise on the edge of the pupil, and at eccentric positions of the eye relative to the camera, the pupil is closer to an ellipse than a circle. In [25], the pupil is modeled as an ellipse, and the parameter extracted is the center of the ellipse corresponding to the center of the pupil in the image.

Task 3: Gaze estimation

Once appropriate model parameters have been determined, the third task involves mapping the derived parameters to a problem space. In many applications, the value being sought is the point of gaze, in which case the parameters provide, for example, the 2-D or 3-D location of the eye which would then be mapped to an angle of gaze, a point of gaze, or both. Effective mapping functions are generally non-linear. These functions can be derived directly using analytic geometry, or indirectly using functional approximation, artificial neural networks or similar non-linear approximation methods.

Thus, a good VOG-based eye tracking device intended to be used in a human computer interface must maintain the eye in the field of view of the camera(s), detect and extract information about parts of the eye (and possibly face), and use that information to estimate where the person is looking.
1.4 Overview of GTD

This thesis explores and analyzes pre-existing VOG-based eye tracking devices. Based on that investigation, a design for a novel non-contact video-oculograph that performs all three tasks described in the previous section is presented. The details of a particular implementation of that design – the GTD (Gaze Tracking Device) eye tracking system – are provided, along with an analysis of results obtained by testing the system. In addition, the main issues, challenges and research questions encountered are reported.

The GTD system, as illustrated in Figure 1.5 and Figure 1.6, consists of both hardware and software to perform its task. It contains a pair of digital video cameras mounted on a custom frame and equipped with special lenses and infrared filters. Infrared lighting is used to take advantage of certain physiological properties of the human eye to detect it in the scene of each camera and to minimize the effects of changes in ambient light. One of the cameras is equipped with a wide angle lens to allow searching for an eye over a large space through which the user’s head can move. Based on the information obtained from that camera, the second camera, equipped with a narrow angle lens, is oriented toward the user’s eye with stepper motors and a mirror. A high resolution image of the user’s pupil and glint is then obtained. The centre of the pupil in both cameras, and the centre of the glint in the narrow angle camera are calculated and can be used as inputs to algorithms for calculating the person’s angle of gaze and point of gaze.
Figure 1.5: CTD Eyetracking System

- Rotating Frame
- Servopot
- Stationary Wide-Angle Camera
- Geared Stepper Motor
- LED Array
- Narrow-Angle Camera
- Rotating Mirror
- Mechanical Stops
- Eyetracker Isometric View

Scale 0.67
The entire system is connected to the user's computer via a serial port and IEEE 1394 interface. The computer controls the motors and infrared light sources, and communicates with and retrieves images from the cameras through these two interfaces and some simple custom electronics. It also calculates the pupil and glint centres using various image analysis techniques, and estimates the user's point of gaze and optionally angle of gaze.

The GTD eye tracking system successfully implements its entire computational load in software and only requires simple, compact optics and electronics attached to the user's computer. It provides tracking and parameterization of a user's pupil and glint that is robust to changes in eye shape and colour, ambient lighting and the presence of eyeglasses (see Chapter 5). The sub-pixel accuracy of the parameterization (0.1%) and the processing speed (9 frames per second) is sufficient for estimating the user's point of gaze on a computer monitor to within 17 pixels, as shown in Section 3.2.1 and Appendix E.

![Figure 1.6 Photograph of GTD eye tracker in action](image)
2 Previous Work

Research that has contributed to the development of better devices for determining a user's point of gaze can generally be categorized as methods based on taking images of the eye and possibly the head or face (VOG methods), and those based on other physiological characteristics of the eye (non-video methods). By these definitions the GTD system described in this thesis can be considered a VOG-based human computer interface device. Therefore, the following sections will focus primarily on research found in the literature that has been used or can be used in a VOG-based device intended for use in a human computer interface.

As mentioned in the previous chapter, an effective VOG-based eye tracking device intended to be used in a human computer interface must perform three tasks: (i) maintain the eye in the field of view of the camera(s); (ii) detect and extract information about parts of the eye (and possibly the rest of the face); and (iii) use that information to estimate where the person is looking.

Aside from work on developing complete systems, there is a substantial body of literature that describes research relating specifically to task (ii) above of a VOG-based device. Thus, Section 2.3 reviews research relating to the second task. Section 2.4 covers a number of existing systems that either accomplish the second and third task but not the first, or do not provide very high resolution tracking. That is, they successfully track the user's gaze, but are not able to handle large head movements while still providing high resolution. Section 2.5 describes a few systems that accomplish all three tasks with some degree of success. All of the work in Sections 2.3, 2.4 and 2.5 assume
the user is not wearing anything special or required to have markers placed on the face—that is, they are completely non-contact, like the GTD system.

However, some VOG-based devices either are head-mounted or make use of special markers. These systems are described in Section 2.2 even though they are not as desirable as non-contact devices for use in a human computer interface.

For the sake of completeness, Section 2.1 below describes briefly the most widely used non-video methods. In all cases, the emphasis will be on analyzing systems based on their performance and usefulness as a human computer interface device.

2.1 Non-video methods

Electrooculography (EOG) involves placing electrodes on the skin and measuring the difference in electrical potential caused by the movement of the six eye muscles. Martin and Harris [26] used EOG to build a joystick style interface. While EOG is fairly accurate and inexpensive, it suffers from various technical limitations such as drift [13]. In addition, EOG-based systems are susceptible to interference from electromyograph (EMG) and electroencephalograph (EEG) signals from the eye muscles and brain, respectively.

The magnetic search coil method [27] is a very accurate one that involves placing contact lenses on the eyes with wires attached, and using large magnets placed around the subject’s head to measure the change in the 3-D position of the contact lens, and hence the eye. The magnetic coil method is the most accurate of any existing method of tracking the eye. However, it suffers from limitations that make it virtually unusable as a human computer interface device. First, it requires the user to stay within a confined area—namely, between two or more large coils that produce magnetic fields. Second, Larsen
and Stark [28] have shown that search coil methods are difficult to calibrate, especially to allow head movement.

Both of the above techniques require the intrusive attachment of part of the system to the human subject, which is usually acceptable in a clinical setting, but a major disadvantage in other settings such as a computer interface. VOG provides a potentially less intrusive approach to eye tracking and holds more promise as the basis of a useful human computer interface device.

2.2 VOG-based systems requiring user contact

In the 1990s, several head-mounted systems were developed, that required the user to wear a helmet or similar apparatus on the head. Head-mounted systems move with the head, thereby eliminating the need to explicitly perform the first task of a VOG-based eye tracking device. In these systems, cameras are mounted on the apparatus to take high resolution images of one or both eyes. Since the apparatus by definition moves with the user’s head, the eye is always guaranteed to remain in the field of view of the camera.

Robinson and Wetzel [29] developed a system using fibre optics in a helmet. Also, Nakamura et. al. [30] developed a head-mounted system, which had an average running speed of 320ms per frame. Both these systems tracked the pupil centre and the glint, and used analytic geometry based on the relative motion of the pupil centre and glint to estimate the user’s angle of gaze.

Later (1995), DiScenna et al.[31] evaluated and compared a head-mounted system with the search coil method. They used a high end, expensive VOG system capable of running at more than 120Hz, and showed that it can approach the accuracy and speed of the magnetic search coil. Given that the VOG system does not require wearing contact
lenses or being confined to a space between two large magnets, the study provides good
evidence that VOG has the potential of being a viable alternative to the traditionally more
accurate and fast search coil method of tracking a user’s eyes.

The major drawback of head-mounted systems is that they require the user to wear
apparatus in order to use the device. This is cumbersome and adds to the cost of the
equipment.

Aside from head-mounted systems, the other approach that requires user contact is
one that uses markers placed on the user’s face. Kim and Ramakrishna [32] built a
system (1999) that used a marker placed on the face. Their system used edge detection
and template matching to search for the iris and the artificial marker. They then fit an
ellipse to the iris to find its centre. The position of the marker, the radius of the iris and
the vector from the marker to the centre of the iris were used to calculate the point of
gaze of a user on a computer monitor. They achieved an accuracy of 12.5% of the
monitor’s size horizontally and 10% vertically. Unfortunately, it did not handle large
head movements well (Kim and Ramakrishna [32] do not explain why) and was sensitive
to variations in lighting and iris colour.

Miyake et al. [33] more recently (2002) proposed a system that uses a single camera.
They use markers placed on the face of the user to provide a point of reference from
which measurements can be made. Specifically, their system measures the centres of
both irises of a user, and although they mention that their measurements could be used to
calculate the user’s gaze, they do not report any results. As for the iris centre calculation,
they only report whether their results are “correct” or not (i.e., whether the calculated
gaze was within a certain tolerance of where the user was looking). Their system runs at
As they do not give any other measure of performance, or its sensitivity to environmental or user changes, it is difficult to assess its overall merit as an eye tracking device. In addition, the use of markers makes both their system and that of Kim and Ramakrishna less desirable as human computer interface devices than other completely non-contact devices such as the GTD system.

2.3 Feature detection and parameterization

A task common to all VOG-based eye tracking devices – regardless of whether they require user contact or not – is that of detecting and parameterizing features of the eye and possibly face (see Section 1.3 for more details). Some of the research over the past decade relating to eye tracking devices has focused specifically on improving this particular aspect of VOG-based systems using deformable templates, template matching or statistical approaches.

Yuille et al. [15] first proposed (1992) a model whose parameters consisted of the location of the iris boundary, the eyelid boundary and the lip boundary. Given a series of images, they manually selected the values for the parameters in the initial image, and then used deformable templates to track the eye and lip in subsequent images. The advantage of their approach was that it provided more than just the pupil centre, but could be used to extract other physiological characteristics of the eye (such as the size of the pupil) that may be of interest to the human computer interface designer. Their implementation could not process more than 1 frame per second (fps) nor could it handle more than minor head movements. Also, the weighting of the deformable template’s energy functions had to be set manually.
Xie et al. [17] improved on the method proposed by Yuille et al. to achieve better accuracy. Deng and Lai [16] subsequently (1997) modified the model of Xie et al. to remove the need for setting the energy function weights manually. All these deformable template approaches are sensitive to changes in lighting conditions and eye shapes and are too slow for a real-time eye tracking application because of their computational complexity compared to simpler image processing techniques.

Sung and Anderson [22] have compared the least squares error (LSE) and maximum likelihood estimation (MLE) methods of fitting data to a given model of eye features. They found that there is a trade-off between robustness and speed: MLE is more robust but also more computationally intensive.

Interestingly, Wagner and Galiana [21] have compared three template matching schemes for measuring eye position in an image. They conclude that simple image processing algorithms are sufficient for most eye tracking applications, and that complex, computationally expensive model-based approaches are not necessary.

Tian et al. [34] later (2000) developed a dual-state parametric eye tracker that is useful for being able to distinguish between open and closed eyes. Their system uses template matching to track the iris boundary, eyelid centre and eye corners for both eyes, and is very effective for detecting blinks. However, it is computationally intensive (they report a processing speed of 3 frames per second), and does not perform well with head movement. It also only determines the eye position and state, and not the user’s gaze.

In general, good edges in the image are needed for any statistical method (such as the ones described above) to work properly. Changes in users, lighting conditions and eye glasses all significantly affect the performance of eye tracking systems based on
statistical approaches. In addition, such systems typically require quite a bit of training and calibration, often on a per user basis.

Chen et al. [35] recently (2000) developed a device for use in autostereoscopic displays that tracks the eyes and face of a user. First, they detect the face using principal component analysis or PCA (see Moghaddam and Pentland [19][20] for a discussion of using PCA for object detection), and then a winner-update algorithm to track the face. They consequently significantly reduce the portion of the image over which they need to search for images of the eye. Therefore, they use a simple template matching scheme (convolution) to detect and track the eyes. Unfortunately, their system is only capable of low resolution tracking. Also, PCA can also be quite sensitive to lighting changes, scale changes, and facial changes such as hair. It also requires extensive training for robust performance. Finally, they have not yet developed a means of using the face and eye locations to actually calculate the user's gaze.

All of the approaches described above require extensive training (often on a per-user basis), are sensitive to lighting changes, and most are sensitive to different eye shapes and colours. Some methods also require manual intervention and are computationally intensive. Ebisawa et al. [23] used a novel approach called the image difference method (see Section 2.5) to detect the pupil, and fit an ellipse to the pupil to find its centre. Their method requires no training for detecting and parameterizing the pupil, is not very sensitive to lighting changes, different eye shapes or colour, and has been implemented in real time. Therefore, it is the most promising way of detecting and parameterizing features of the eye that can subsequently be used to calculate the user's point of gaze, and is the method used in the GTD system.
2.4 Large head movements and high resolution

Most of the existing eye tracking devices that output a user’s angle of gaze or point of
gaze either do not actually perform the first task of a VOG-based system or provide
relatively low tracking resolution. That is, although they track the user’s gaze (often in
the presence of some head movement), they only work if the eye remains in the field of
view of the camera, and often can only detect the eye features being tracked with limited
resolution. However, these systems have made significant contributions to the
development of complete VOG-based devices. They are thus reviewed in this section.

Cornsweet and Crane [36] developed a system that tracked the first and fourth
Purkinje image. It was accurate to within 1 arc minute (i.e., was able to distinguish eye
orientations to within 1/60 of a degree of their actual orientation), which is better than
most modern video-based eye trackers, and ran as fast as 100Hz. However, it was
sensitive to other Purkinje images (which tended to confuse the system) and required
specialized hardware as well as very accurate and expensive recording devices. It also
required the user to use a “bite bar” (a device often used by ophthalmologists where the
patient bites down on a fixed bar in order to keep the head still). In short, it was not
practical for a human computer interface device.

Johnson et al. [37] developed a system that used fibre optic cables to track the
boundary of the iris and sclera. Light reflected off each eye was transmitted through a
lens (which was placed close to the eye) to a remote detector via a fibre optic cable. The
system compared the output of the two detectors (one for each eye) to determine an angle
of gaze. They were able to achieve accuracy of 1° and better but only when they used a
bite bar. The equipment was expensive and sensitive to head movement.
When calculating the AOG or POG directly, one of the sources of error is noise or errors in the measurements of the model parameters being tracked. To avoid the sensitivity of direct gaze calculation to noise in the measurements used, some systems have instead attempted to employ approximation techniques. For example, Betke and Kawai [38] have built a system that uses a non-linear approximation approach to finding the centre of the pupil with respect to the centre of the eye based on raw pixel data. From the pupil centre location, they then infer an angle of gaze. The system requires manual training and is sensitive to changes in scale and changes in a user’s face. They also do not describe any mechanism for searching for an eye in an image of the entire face (their experiments were on images that only contained the eye). Further, the system does not actually track an eye in the presence of head movement.

Stiefelhagen et al. [39] use an artificial neural network (ANN) to achieve 1.9° accuracy in angle of gaze at 15Hz. They track the face using colour segmentation. They then use an iterative thresholding method to detect “dark regions” in the upper half of the face corresponding to the user’s two irises. The locations of the irises are then passed to an ANN to calculate an angle of gaze. The system is sensitive to facial changes such as scale and hair, requires extensive training for robust performance and is computationally intensive. Their system is sensitive to head movement.

The system described by Matsumoto and Zelinsky [40] also tracks the face and irises. Their system uses stereo vision to track the face and thereby estimate the gaze. They use template matching to detect both the face and the irises in each image, and a correlation-based stereo matching technique to track the face and calculate a 3-D head pose (i.e., the direction in which the user’s head is pointing). The points used for the correlation are the
eye corners and the lip corners of the user. The head pose is combined with the location of the irises to calculate the angle of gaze using analytic geometry. The system runs at 30Hz. They do not, however, report quantitative measures of accuracy and speed in the presence of head movement and changes in lighting conditions, facial expressions, facial hair, etc. Further, the system requires good face templates and provides only a low resolution estimate of the location of the irises (since the whole face must be visible in the image and the irises occupy a relatively small portion of the face).

The eye tracker recently (2002) developed by Theis and Hustadt [41] employs colour segmentation to track the iris and eye corners as well as the face of the user. Specifically, they use colour segmentation to track the face, and then look for “dark regions” in the face to determine the location of the irises. A Hough transform is used to find the edges of the irises, and a corner detector is used to find the eye corners. The iris centre and eye corners are combined to calculate the angle of gaze using analytic geometry. There is no indication of accuracy and speed performance, though the approach seems computationally intensive. As with the system developed by Matsumoto and Zelinsky, the resolution with which it can detect the iris centre or eye corners is limited.

A very promising new (2002) approach has been introduced by Zhu and Yang [42], where eye features are measured to sub-pixel accuracy (as defined in [42]). This is important because small rotations of the eye – and hence small pixel deviations in the image of the eye features – can translate to large lateral offsets in the point of gaze of the user, especially at large distances from the viewing plane. With their system, they report an accuracy of between 1.1° to 1.4° in angle of gaze, but they have only performed off-line calculations, so the speed is unknown. In their system, they use an edge detector to
find the edge of the iris and a corner detector to find the corners of the eye, both to within sub-pixel accuracy. They then map the vector from the eye corners to the irises to a point of gaze using a calibration procedure. After calibration, the system uses simple linear interpolation to calculate point of gaze. Their work seems to focus on improving the accuracy of the image measurements. An obvious means of improving accuracy is to increase the focal length of the camera lens, thereby increasing the number of pixels represented by the iris and eye corners. However, this would limit the space over which the user’s head could move, and their system does not describe how to compensate appropriately. It also requires calibration on a per user basis.

The eye tracker proposed by Shih et al. [43] minimizes the need for calibration by looking at the first Purkinje image of multiple light sources (i.e., multiple glints). They combine the location of the Purkinje images with the centre of the pupil taken from multiple cameras to calculate the angle of gaze and point of gaze using 3-D computer vision techniques. Specifically, they use a priori knowledge of various constraints on the reflection of light off the surface of the cornea to calculate the gaze. They assume the camera parameters are known (which means that some calibration is required) and have only simulated their proposed system so far.

A more recently developed system (2002) by Yoo et al. [44] also tracks multiple corneal reflections and a pupil. Their system employs four infrared light sources fixed on each corner of the monitor, and tracks their reflection off the cornea of the user’s eye. They also track the pupil using the image difference method (see Section 3.3.1). The centre of these five features enables the system to directly calculate the user’s point of gaze using simple 3-D geometry without the need for a second camera. It is an efficient
approach to estimating point of gaze, and they report an accuracy of 2.2% of the monitor's width and height. Unfortunately it is also sensitive to head movement, does not yet provide accurate pupil position detection (their model does not take into account the curvature of the cornea), and requires manual adjustment of a threshold used in the image processing.

Morimoto et al. [45] also recently (2002) proposed using multiple light sources. With their proposal, the need for calibration could be eliminated. However, they have only carried out simulations.

The devices described in this section are all able to calculate the user's AOG or POG. Some of them measure the features they use to perform the calculation of the AOG or POG very accurately, but cannot handle head movements well. Others can handle head movements but cannot measure the features with very high resolution (usually because they use a wide angle lens). A few complete VOG-based systems exist that track the AOG or POG with reasonable resolution and can also handle some small head movements. These are reviewed in the next section.

2.5 Complete VOG-based devices

Talmi and Liu [8] use a three camera system for use in an autostereoscopic display. Two fixed cameras are used to determine the 3-D pupil position by applying traditional stereo vision. Another high-resolution camera mounted on a pan/tilt apparatus is used to determine the accurate pupil position. Specifically, they use PCA to detect the eye in two of the cameras. They then extract the depth of the eye using stereo matching. They fit a circle to the pupil in the third camera and calculate its centre. They also calculate the centre of mass of the brightest pixels near the pupil in the same camera, and use that as
the location of the glint. Their use of PCA suffers from the same problems described in Section 2.3 for the system developed by Chen et al. Also, they only report a speed of 1fps when the user’s head moves significantly.

Between 1994 and 1998, Ebisawa et al. [23][46][47][48] developed a novel approach they refer to as the image difference method. This approach exploits the property of the fovea whereby infrared light shone onto it through the pupil produces a bright disk (caused by the retinal “red-eye” reflection) if the light source is on the viewing axis and a dark disk if it is off axis (see Section 3.3.1 for more details). The system alternates between two light sources – one on the principal axis of the camera, and one off the principal axis – between successive frames. The intensity difference in the images between consecutive frames is then used to detect the pupil. The major advantages of the image difference method are that it is less sensitive to environmental (especially lighting) changes, and that it makes the image processing used to find the pupil in an image simpler and less computationally intensive. In their system, an ellipse is fitted to the edges of the detected pupil, and the ellipse centre used along with the centre of the glint (which is also tracked) to calculate the user’s point of gaze and angle of gaze. The device runs at 30Hz. Their system also uses image analysis techniques to handle cases where the user is wearing eye glasses. This is important because light reflected off the surface of the eye glasses can be easily confused with the glint used by the system to calculate the user’s point of gaze. However, the system needs a fair amount of user calibration, and specialized hardware to do the image processing.

The original system did not compensate for head movements very well. Sugioka et al. [49] and Ebisawa et al. [50] addressed this deficiency by using an ultrasonic
measurement device to detect the depth of the user's face, and a single mirror and camera equipped with a motorized zoom lens to track the pupil and glint. The ultrasonic device and a pan/tilt apparatus for the mirror are used to maintain the pupil and glint within the field of view of the camera, even in the presence of large head movements. In [51], Marui and Ebisawa further propose a method to search for the eye using the mirror and pan/tilt apparatus. Rotating the mirror takes 400ms, which means that the maximum speed of the system with large head movements is 2.5fps. In addition, the ultrasonic meter is not very precise, which gives rise to errors in gaze estimation. The system also needs specialized hardware and an expensive motorized zoom lens.

A device developed at the IBM Almaden Research Center between 1998 and 2002 was first described by Morimoto et al. [52] and is particularly attractive as it is relatively inexpensive. It was successful at detecting multiple pupils in an image in the presence of specular reflection (e.g., those caused by eye glasses). They used the image difference method introduced by Ebisawa et al. As such, the system was relatively insensitive to changes in lighting, eye shape or eye colour. However, the system could not handle motion artefacts caused by eye movements very well. Also, the system only detected pupils, and did not calculate the user's gaze. In addition, in order to keep the eye in the field of view of the camera, it used a wide angle lens, which meant that the resolution of the image used to detect the pupils was low.

Haro et al. [53] subsequently (2000) improved the system initially developed by Morimoto et al. [52] by modeling both the structure and dynamics of the eye to increase robustness. They used a Kalman filter to handle motion artefacts. Haro et al. [54] then improved the same system further to eliminate false positives (other image artefacts...
incorrectly identified as the pupil) by using probabilistic principal component analysis (PPCA). However, both systems ([53] and [54]) were still sensitive to changes in lighting, eye shape and colour. They too did not calculate the user's gaze and used a wide angle lens, which meant that even the pupil detection was done with relatively low resolution.

Morimoto et al. [4] use a second order polynomial function to map the vector between the glint and the pupil centre to a point of gaze on a computer monitor. The coefficients of the polynomial function are determined using a calibration process that involves having a user look at a grid of 3x3 grid of points and recording the corresponding positions of the glint and pupil centre. For small head movements their system achieves an accuracy of about 1° in angle of gaze (which they define as 1cm on the screen when the user is sitting 50cm away) at 30Hz. They use a pan/tilt mechanism to maintain the eye in the field of view of the camera. However, the system does not tolerate large head movements very well, since the pan/tilt apparatus is too slow to keep up with large, natural head movements.

Wang and Sung [55] developed a system in 2002 that tracks the iris, eyelids and eye corners. Their system consists of two cameras. One is stationary and used to detect the face and head pose. The second camera is a high-resolution camera on a pan/tilt/zoom apparatus used to detect the iris and corners of the eye accurately. The two cameras are used to build an integrated system. The system detects the edges of the iris, and then applies morphological operators to find the longest edges, which it then uses to estimate an ellipse representing the iris using a novel ellipse-fitting algorithm (the "one-circle" algorithm). While the proposed algorithm shows promise, it has certain numerical
limitations as manifest in degenerate cases (e.g., when the user's pose is not close to the frontal view). They report an accuracy of 1.5cm at a distance of 1.5m in POG.

The system tends to be sensitive to changes in lighting conditions, the use of eye glasses, and changes in facial hair. In addition, different facial expressions seem to interfere with accurate head pose calculations. They do not report how fast their system works. However, calculating head pose on top of all the other calculations could be computationally too expensive to be implemented in real-time.

All the systems described above only detect the location of the eye features (pupil, glint, iris centre, etc.) to within a pixel at best. As shown in Appendix E, this places a severe restriction on the best accuracy in POG able to be achieved. In addition, none of the systems are able to track the eye features (and hence the POG) in real time in the presence of large head movements.

A VOG-based eye tracking device that can effectively be used in a human computer interface to track a person's fixations on a computer monitor must be inexpensive, provide accurate and continuous measurements of eye features that can be used to calculate point of gaze in real time even in the presence of large head movements, operate at a speed of at least 3Hz (since a typical fixation lasts about 350ms), require no per-user calibration and little or no overall calibration, and be robust to changes in lighting, facial hair, facial expressions or different users. None of the complete VOG-based eye tracking devices described above meet all of these requirements. The GTD system described in this thesis, on the other hand, is designed to be a complete VOG-based eye tracking device that meets all of these requirements and provides a more accurate location of the eye features required to calculate the user's POG than any pre-existing device.
2.5.1 Camera calibration

The GTD system can use both a direct and indirect method of estimating the user's point of gaze, as described in Section 3.4. The direct method uses analytic geometry to map measured 2-D image pixel data to 3-D positions and relationships. This mapping often assumes perfect measurements. Therefore, slightly inaccurate measurements can sometimes drastically degrade the performance of the gaze estimation, and hence the eye tracker as a user interface device. Calibrating the cameras and lenses used to obtain the 2-D image data can help reduce those inaccuracies in measurement. The following provides an overview of research in the area of camera calibration as it pertains to the GTD system.

While a thorough review of camera calibration is beyond the scope of this thesis, a very brief description of the problem and a couple of useful approaches to its solution are presented in this section. Also, Section 3.5 gives a more detailed explanation of camera calibration.

Mapping 2-D image data to the 3-D coordinates of a physical point in space that would produce the corresponding pixel values in the image can be done using a model that assumes a pinhole camera (Figure 3.5 illustrates the pinhole camera model). Most modern digital cameras use a spherical lens. However, the use of such a lens can affect the focal length and the centre of the image plane used in the model. In addition, a spherical lens often introduces radial and sometimes skew distortion in the image. These intrinsic lens and camera parameters can be estimated using a variety of camera calibration algorithms. Several algorithms also calculate the extrinsic parameters of the camera. The extrinsic parameters consist of the three rotation and three translation parameters used to map a 3-D point in a world frame of reference to the 3-D coordinate...
system of the camera frame. Figure 3.8 illustrates how the intrinsic parameters of a camera can be used to compensate for lens distortion in a pinhole model, and how the extrinsic parameters can be used to map 3-D coordinates of physical objects to a pixel coordinate in the image obtained from the camera.

Tsai [56] provides a very versatile camera calibration approach. From it, one can extract the focal length of the lens, the true centre of the image plane (or the so-called "principal point"), and lens distortion parameters. It requires information about the camera sensor, which is not always available. C'Donnell [57] has developed a similar approach for a camera-aided log volume input system.

Zhang [58] and Heikkilä and Silvén [59] have proposed even more versatile camera calibration algorithms that require no knowledge of sensor or camera parameters. As with Tsai, their approaches calibrate both the extrinsic and intrinsic parameters of the camera. Both approaches only require a pattern (e.g., a checkerboard pattern) imaged at several views. No other *a priori* knowledge or measurements are required.

It turns out that while existing algorithms provide a very accurate means of calibrating both intrinsic and extrinsic camera parameters, at high magnification (i.e., a large lens focal length) it is difficult to take the measurements from the images needed to perform the calculations accurately. This thesis therefore proposes an alternative method (using non-linear optimization) of calibrating camera parameters simultaneously with other system parameters that require calibration for accurate results.

The final task (see task (iii) at the beginning of this chapter) of calculating an absolute point of gaze is of great importance in any eye tracking device. The method used needs to be accurate, robust and efficient. While approximation methods provide robustness to
noise, they are more computationally intensive than direct methods. Direct calculation of the gaze, when combined with camera calibration, seems to be more efficient and in theory more accurate. However, in practice, this has not yet been shown.
3 System Design

The main purpose of this thesis is to present a novel design for a video-oculograph, and to apply the design to the construction of a human computer interface device that tracks the user's fixations on a computer monitor. In the process of describing the design and subsequent implementation, the main issues, challenges and research questions encountered will be reported.

3.1 Assumptions and overall objectives

The eye tracking system described in this thesis is designed as a human computer interface device. It is designed to be capable of tracking fixations but not necessarily saccades or micro-saccades. Therefore, the system is not necessarily designed for use in a clinical setting. Further, the user is assumed to have voluntary control over the eyes.

The system is designed to be relatively insensitive to changes in lighting conditions. It is also expected to perform well for any human who is not visually disabled or otherwise suffer from ophthalmic pathology that cannot be corrected with eye glasses or contact lenses. It is robust to changes in facial expressions or features (such as hair), and compensates for natural head movements in real time.

Several existing devices report accuracy in terms of visual angle, and all of them only poorly define at best what is meant by this angle. What we are more interested in is the accuracy of a system in determining the user's point of gaze on a computer screen. Appendix E describes the relationship between the resolution and accuracy of the system.

For a 15” monitor with a screen resolution of 1024x768 pixels, this relationship implies that the system should be designed to determine the point of gaze to within 17 pixels (as shown in Table E.1 in Appendix E).
We assume that the monitor screen is roughly flat, and its dimensions and origin with respect to the apparatus is known.

3.2 Tracking approach

The first task of a non-contact, VOG based eye tracker is to track the pupil. In order to allow large head movements, we must approximate the location of the pupil in 3-D space, and then ensure that our camera is oriented so that the pupil falls within its field of view. One approach would be to use a single camera with a short focal length. However, wide angle lenses generally provide lower resolution images and are more prone to distortion. On the other hand, using a single camera with a longer focal length would provide better resolution but would limit the field of view and hence the amount of head movement tolerated by the system.

We use instead a two camera system, as in Figure 1.5 (see also Figure 3.4). One camera (the wide angle or WA camera) is fixed, equipped with a wide angle lens and faces away from the screen and toward the user (i.e., in the positive z direction). The other camera (the narrow angle or NA camera) is mounted on a frame that rotates about the x-axis (tilt), equipped with a narrow angle lens, and faces the WA camera. The narrow angle lens provides a higher resolution image of the eye, and hence allows a more accurate measurement of the point of gaze. The frame also holds a mirror that rotates about the y-axis and reflects light rays into the NA camera. Thus, the NA camera is able to rotate about two of the axes, and thereby track the eye of the user even in the presence of head movement. The mirror is much lighter than the camera or the frame and therefore requires less torque from the stepper motors used to perform the rotation. This allows tracking eye movements even in the presence of rapid horizontal head movement.
At the heart of a VOG system is the image acquisition subsystem. Therefore, it is important to choose the right cameras, lenses, filters and PC interfaces.

3.2.1 Image capture

The eye tracker presented in this thesis uses digital video cameras. A purely digital interface reduces the amount of noise and distortion in the images. A standard interface such as IEEE1394 or USB2 also makes the interface cheap and accessible. Note that the low bandwidth of USB 1.0 (200Mb/s) makes it undesirable for tracking rapid eye movements.

The cameras should be able to deliver at least 15 frames per second in uncompressed format, which is sufficient for detecting fixations. They should also either allow external triggering or provide a synchronization signal, either of which could be used to synchronize the start of the image integration by the camera sensor with appropriate lighting conditions.

In particular, because the system takes advantage of certain physiological properties of the eye with respect to infrared lighting (see Section 3.3.1 and Appendix E), the lighting conditions are as follows. An illumination scheme is used that consists of two rings of infrared or near-infrared light emitting diodes (LEDs). One ring is placed sufficiently far from the centre of the camera lens to produce a “dark pupil”. The other ring would be close to the optical axis of the camera in order to produce a “bright pupil” (see Section 3.3.1 for a definition of “dark pupil” and “bright pupil”). Only one ring of LEDs would be turned on at any given time, alternating between each ring from one

---

4 We are primarily interested in tracking fixations. The longest average fixation for a human lasts approximately 350ms. The particular image processing technique used works best when a temporal filter is applied (see Section 4.6.5). Assuming a temporal filter length of 5 frames, this means that we should be able to capture an image at least every 70ms, which implies a frame rate of 14.3 Hz. During a saccade or a blink, we are not interested in tracking the pupil.
frame captured by the cameras to the next. According to the image difference method, the difference between successive image frames highlights the location of the pupil. Each camera is equipped with its own pair of LED rings. Section 4.6.1 shows how the image difference method has been used in the GTD system, and Figure 4.8 to Figure 4.11 illustrate an example of the method in action.

Such a scheme requires that the camera sensors be sensitive to infrared light. In addition, infrared filters need to be used to maximize the signal of interest (i.e., the reflected infrared light off the pupil of the user’s eye).

3.2.2 Head movement compensation

In order to successfully track the eye so that a good image of the pupil can be obtained, the system must first provide a means of rotating the NA camera so that the user’s eye falls within its field of view. In other words, a process is required to track the eye in the presence of head movements. We use the WA camera to drive this tracking process.

The WA camera lens’ focal length should be short enough to detect a human face at normal distances of a user from a computer screen. The image difference method proposed by Ebisawa et al. [23] (see Section 3.3.1) is used to locate both pupils of the user in the image. When both pupils are in the field of view of the WA camera, the distance between the pupils is used to approximate the distance of the user from the screen. This depth approximation, along with the location of one of the pupils in the image, is then used to position the frame and mirror. This is equivalent to rotating the NA camera about the midpoint of the mirror (point M in Figure 1.5). Appendix E describes how the angles required to properly orient the NA camera are calculated.
3.3 Eye parameterization

Once the eye is in the field of view of the NA camera, a high resolution image of the eye is available for parameterization, which is the second task of a VOG based eye tracker. Since we know the location of the light source we are using, we can calculate the vector from the light source to the reflection of that light source off the surface of the cornea (the "glint") if we calculate the pixel location of the glint in the NA camera’s image. It can be shown (see Section 3.4.1) that the 3-D location of the pupil and the ray from the light source to the glint provides sufficient information to calculate the angle of gaze, from which the point of gaze of the user can be inferred.

Thus, in the system presented in this thesis, the eye is parameterized by the centre of the pupil in both the NA camera and WA camera’s image (from which the 3-D location of the pupil can be calculated), and the centre of the glint in the NA camera (from which the ray from the light source to the glint can be calculated).

In order to perform this parameterization, we use the image difference method to detect the pupil in both the NA camera and WA camera’s image, and ad hoc image processing techniques to detect the glint. Once the three features have been detected, they are each modeled as an ellipse. The parameterization of the eye therefore consists of the centres of the three resulting ellipses.

3.3.1 Pupil and glint detection

Before the eye parameters can be extracted, some image processing needs to be performed on the images obtained from each of the WA and NA cameras. The following describes briefly how the pupil and glint can be detected in each image.
The fovea reflects some of the light entering the eye, especially light in the infrared and near infrared part of the electromagnetic spectrum. Suppose a bright light is shone into a person’s eye along the viewing direction of an observer. Then all the light reflected off the fovea and back through the person’s pupil can be seen by the observer, and the pupil looks like a bright disk (referred to as a “bright pupil”). This is the basis of the so-called “red eye effect” caused by flash photography. If, on the other hand, the light is shone at a different direction than the viewing direction, then the light reflected by the fovea travels in a different direction, and the pupil does not appear as a bright disk (referred to as a “dark pupil”). If the observer utilizes an infrared filter, then the brightness of the pupil will be notably different under the two lighting conditions, regardless of the actual colour of the user’s pupil. By using infrared lighting and infrared filters in front of the camera sensors, and by alternating between a light source that emits light along the viewing axis of the camera and one that emits light along a different direction, we can exploit this characteristic of the fovea, and detect the pupil by examining the difference between two successive images of the eye in a sequence of video frames.

First, in a sequence of video frames, the last image with a dark pupil is subtracted from the last image with a bright pupil to obtain an instantaneous difference image. If the eye has not moved at all from one frame to the next, then the only sizable area of relatively high pixel intensity in the instantaneous difference image is the pupil. However, in practice the eye moves between frames – sometimes significantly – and hence a means of dealing with motion artefacts is needed.
For example, Figure 3.1 shows a single difference image (image with dark pupil subtracted from image with bright pupil). The user's head has moved between the two images used to produce the difference image, thus causing motion artefacts. These artefacts result in the large number of relatively bright objects in the scene that make the task of identifying the pupil very difficult. Compare this to Figure 4.11 (page 85), where three consecutive difference images are averaged and the numbers of motion artefacts are significantly reduced. In the latter, over the three frames, the difference in the brightness values caused by the switching of the LEDs is much greater than those caused by motion between individual frames. Therefore, a moving temporal window is applied whereby a few consecutive difference images are averaged to obtain an aggregate difference image. As shown in Figure 4.11, this step eliminates almost all the bright pixels in the image shown in Figure 3.1 that correspond to motion artefacts.

The aggregate difference image is then thresholded to obtain a black and white image consisting of several "blobs". Morphological operators are then applied to each blob (specifically, each blob is dilated and eroded). The threshold level is automatically set by looking at the histogram of the image. If the intensities of the two rings of LEDs are
properly set\(^5\), the blob representing the pupil should be the largest one in the image.

Therefore, detecting the pupil consists primarily of finding the largest blob in the image.

The glint is detected from the image with the dark pupil, since it provides a better contrast in cases where the glint falls within the area of the pupil. As with the aggregate difference image, the dark pupil image is thresholded to obtain a black and white image, and dilation and erosion operators are applied. Again, the threshold is automatically set using the image histogram. This time, an ellipse is fit to each blob in the image, and the one whose centre is closest to the last known pupil location is chosen as the glint.

**3.3.2 Pupil and glint centre calculation**

Once the pupil is detected in each image as the largest blob, the next step in the parameterization is to find the ellipse that most closely resembles the outline of the blob. The centre of the ellipse is then assumed to be the centre of the pupil. The same approach is used to fitting ellipses to the blobs used to locate the glint. Specifically, in each case, once the appropriate image blob has been identified, the points on its contours are assumed to lie on or close to an ellipse, and are thus used as inputs into an ellipse fitting algorithm.

Appendix E describes in some detail the algorithm used to fit ellipses in this thesis. The algorithm is used to model the pupil and glint as ellipses in both the WA camera and NA camera images. The centre of the fitted ellipse is then used as the centre of the pupil or glint. Using this approach, the following are approximated:

- x-coordinate of the centre of the projection of the pupil in the WA camera’s image
- y-coordinate of the centre of the projection of the pupil in the WA camera’s image

\(^5\) In the GTD system, the intensity of each ring of LEDs is set using a resistor (see Section 4.5.1). The value of each resistor was chosen empirically.
- x-coordinate of the centre of the projection of the pupil in the NA camera’s image
- y-coordinate of the centre of the projection of the pupil in the NA camera’s image
- x-coordinate of the centre of the projection of the glint in the NA camera’s image
- y-coordinate of the centre of the projection of the glint in the NA camera’s image

These six (6) parameters of the user’s eye (3 pairs of 2-D coordinates) can then be used to calculate the point of gaze.

### 3.4 Gaze calculation

The three ellipse centres found as parameters of the eye features being tracked can be used either directly or as input to a non-linear functional approximation algorithm to estimate the user’s point of gaze.

For a direct calculation, the problem can be broken down to two sub-problems. We define three vectors as follows:

\[
\begin{align*}
\mathbf{v}_w &= [u_w \ v_w] \quad (1) \\
\mathbf{v}_N &= [u_N \ v_N] \quad (2) \\
\mathbf{v}_G &= [u_G \ v_G] \quad (3)
\end{align*}
\]

where \( u_w, v_w, u_N, v_N, u_G, \) and \( v_G \) are shown in Figure 3.3 and Figure 3.4.

For the first problem, given the projections \( \mathbf{v}_w \) and \( \mathbf{v}_N \) onto the WA camera and NA camera sensors, respectively, of the light rays reflected off the pupil, and the projection \( \mathbf{v}_G \) onto the NA camera sensor of the glint, we want to find an analytic expression of the gaze vector \( \overline{E} \). That is, we want to find the 3-D location of the pupil and the vector representing the direction of the user’s gaze with its origin at the pupil.
For the second problem, we want to calculate the projection $E$ of $\overline{E}$ onto the plane representing the monitor screen.

For an indirect approach such as non-linear functional approximation, the six (6) variables represented by the 2-D vectors $v_w$, $v_N$ and $v_G$ are used as inputs, along with the rotation angles $\rho_{\mu}$ and $\rho_{\nu}$ of the NA camera, for a total of eight inputs. We then model two output variables representing approximations of the 2-D coordinates of $E$.

### 3.4.1 Calculation of angle of gaze

The least computationally intensive and theoretically the most accurate means of calculating the angle of gaze is to use the eye parameters directly. To do this, we use ray tracing to follow the path light takes from our sources (LED array), reflecting off the eye and to our camera sensors.

Allvar Gullstrand proposed a schematic of the eye that models the optical characteristics of the human eye (see [12] for details). Briefly, the schematic models the eye as a series of optical media: the cornea, aqueous humor, the crystalline lens and the vitreous humor. Each of these media is a refractive surface and can be schematically represented as an optical lens or mirror.

Using the Gullstrand model of the eye (see Figure 3.2), we can treat the corneal surface of the eye as a spherical optical refractive surface of radius $r = 7.7\, \text{mm}$ (this is an average value, and the actual radius for a given user can vary). We can then trace back the light rays from each of the pupil centre and glint reflected off the mirror and onto the NA camera’s sensor or image plane.

This approach assumes we have a point light source. In the system designed, we use a ring of LEDs as our light source, so in reality it is not a point light source. However,
since the LEDs are arranged as a concentric ring with its axis being the optical axis of the camera, we will get a ring of bright points (or a single solid ellipse if they emit strong enough light) in our image corresponding to the reflection of those light sources off the surface of the user's cornea. We can thus take the centre of the ring or ellipse as the equivalent of a single point source coaxial with the optical axis of the NA camera.

Figure 3.2 Gullstrand model of the eye

Figure 3.3 shows schematically the rays used to calculate the angle of gaze. Figure 3.4 similarly shows schematically information from both cameras that can be used to calculate the 3-D location of the pupil. Although both figures are shown in 2-D for the purpose of illustration, the theory below is presented using 3-D vectors. Note that the figures are not drawn to scale. The variables shown are as follows:
O: origin of our world coordinate system (located at the centre of the WA camera image sensor—see Figure 1.5); also assumed to be centre of WA camera's image sensor

F: focal centre of NA camera

W: focal centre of WA camera

H: origin of gaze coordinate system (computer monitor surface)

M: centre of rotation of the mirror (see also Figure 1.5)

N: centre of NA camera's image sensor

C: centre of radius of the cornea, or the centre of the spherical surface in our model

P: virtual image of the centre of the pupil

G: location of glint on corneal surface (brightest point of reflection of light source, which is at the intersection of ray from F to C with surface of cornea)

E: point of gaze on monitor

\overline{E}: gaze vector, given by unit vector from C to E

S: point of reflection of ray from F to P

\overline{S}: unit vector of ray from S to P

R: point of reflection of ray from F to G

\overline{R}: unit vector of ray from R to G

Q: unit vector of ray from W to P

\hat{n}: unit normal vector for mirror surface
Figure 3.3 Geometry of gaze calculation from pupil and glint images in NA camera
Figure 3.4 Calculation of 3-D pupil location from two cameras
We also make the following assumptions:

1. The monitor surface is flat and parallel to the x-y plane (i.e., the z-axis is normal to the monitor surface and pointing toward the user).
2. The mirror is an ideal flat reflecting surface with its centre $M$ lying on the x-axis and $M_z = 0$.
3. There are no aberrations on the corneal surface and the centre of the pupil corresponds exactly to the centre of the lens and the centre of the fovea.
4. The NA camera can be modeled as a pinhole camera with its focal centre at $F$ and image centre $N$ lying on the x-axis and $N_z = 0$.
5. The WA camera can be modeled as a pinhole camera with its focal centre at $W$ and image centre $O = [0 \ 0 \ 0]^T$.

With the above information, the angle of gaze calculation can be performed by observing that

$$S + \alpha_s \bar{S} = P \quad (4)$$
$$W + \alpha_q \bar{Q} = P \quad (5)$$

Appendix E describes specifically how the 3-D position of the pupil can be calculated, and how the angle of gaze can be determined using a single light source. All the calculations make two fundamental assumptions that, in practice, are generally not true. First, we assume that the cameras and their lenses can be perfectly modeled as a pinhole camera (as shown in Figure 3.5). In reality, there are distortions caused by the lens that may adversely affect the results. Second, we assume that we are able to get the exact location (at least up to the precision of the computer) of the pupil in each image. However, noise in the images introduces errors in those measurements. In addition, the
model of the eye we are using is, in the end, still a model and may differ slightly from the actual physical structures we are measuring. For example, we model the cornea as a perfectly spherical surface, but in reality it is not, and this affects the accuracy of our interpretation of the measured locations of the pupil and corneal reflections. Finally, errors in the measurements of the extrinsic parameters of each camera (the rotation and translation relative to our world coordinate system’s origin) also contribute to errors in the system.

All these assumptions lead to potential errors in our calculation. We can reduce these errors using various means, some of which are mentioned in Section 6.2. The first assumption also leads to an erroneous measure for the focal centres \( F \) and \( W \) as well as the centre \( N \) of the NA camera’s image plane. However, the assumptions have an even more serious implication. All the assumptions also lead to errors in our calculation of the rays \( S \), \( \overline{S} \) and \( \overline{Q} \) from each camera sensor to the 3-D pupil location \( P \). The result is
that the vectors $\mathbf{S}$ and $\mathbf{Q}$ we calculate do not actually intersect, and hence equations 4 and 5 cannot be solved.

For the first assumption, we can take two actions to reduce the consequent errors. They are as follows:

1. We can calibrate the cameras directly using known methods such as those described in Section 2.5.1.

2. We can use calibration points to infer suitable camera calibration parameters for both cameras simultaneously. This could be done as follows. We present to the user a series of targets whose positions on the monitor screen are known. For each target, we measure the corresponding pixel values. This gives us an input vector $(u_w, v_w, u_g, v_g)$ and an output vector $\mathbf{E}$. We use the input vector to calculate an approximate output vector $\mathbf{\tilde{E}}$ using some initial guess at the camera calibration parameters for both cameras. We can then use a non-linear least-squares error approach to minimize the error $\|\mathbf{\tilde{E}} - \mathbf{E}\|^2$ for all the input-output vector pairs (calibration points). So, for example, to model each camera's focal length, image centre and 2nd-order radial distortion model, we would require at least fourteen such calibration points. Of course, more calibration points will generally lead to a more robust estimate of the camera calibration parameters.

Sections 3.5 and 4.8 provide a more detailed explanation of approaches that can be taken to perform the above noted calibration procedures. Note that, in theory, such calibration procedures need only be done once (as long as neither the apparatus nor
computer is moved), since the camera calibration parameters should not change from session to session or even from user to user.

For the second assumption, we can improve our image processing techniques to provide even better accuracy. For the third assumption, the eye can be modeled more precisely and its parameters calibrated for each user's eye. Finally, reducing measurement noise will also help reduce overall system errors.

While it is unlikely that we can ever achieve precise enough measurements to solve equations 4 and 5 directly, if the cameras are properly calibrated, noise in the images is minimized, and the image processing is kept as accurate as possible, then we can obtain an estimate for $P$ from equations 4 and 5 by finding values for the two scale factors that minimizes the distance between $S + \alpha_S \bar{S}$ and $W + \alpha_Q \bar{Q}$.

Using multiple light sources

An alternative method of calculating gaze involves computing the centre of the cornea $C$ first, and then inferring the 3-D pupil location. The centre is not an observable object like the pupil centre, so it cannot be measured using the two cameras. However, if there is more than one light source, then using ray tracing and optics, the 3-D location of $C$ can be computed. The advantage of this approach is that it avoids problems with calibrating two cameras. The disadvantage is that it requires multiple light sources and more precise detection of the images of those light sources in the NA camera's image.

In the GTD system, since a ring of lights is used, we can take any one of the lights on the ring as the second light source. If the LEDs emit light that is strong enough, we will not be able to distinguish the image of one LED's reflection off the cornea from another. However, since together they form an ellipse in our image, we can take a point on one of
the four vertices corresponding to the endpoints of the ellipse's principal axes as the image of a corresponding LED on our ring (see Figure 3.6 below). Appendix E describes how to use multiple light sources to obtain a more robust estimation of the user's angle of gaze.

![Figure 3.6 Using LED array as multiple light sources](image)

**Figure 3.6 Using LED array as multiple light sources**

### 3.4.2 Calculation of point of gaze

Having calculated an angle of gaze directly as shown in the previous section, we can compute the actual point of gaze on the computer monitor as follows:

\[
E = C + \alpha_E \bar{E} 
\]

\[
E_z = H_z 
\]

where \( \alpha_E \) is a scale factor. We assume here that the monitor screen can be modeled as a plane with origin \( H \). Note that if \( H \) is not known, then its values can be found using three (3) or more calibration points. That is, the user can be presented with three or
more points whose location on the computer monitor relative to the origin is known, and the resulting gaze direction vectors used to calculate a value for \( H \).

The above direct approach to calculating the point of gaze requires an accurate measure of \( C \). Any errors in the measurement of \( v_n \), \( v_g \), \( \rho_n \) or \( \rho_v \) can significantly reduce the accuracy of \( C \), thus making the direct approach to calculating \( E \) less robust. An indirect approach such as non-linear functional approximation could potentially provide a more robust solution.

What we require is a non-linear mapping from the eight (8) input values (3 vectors \( v_{w}, v_{n}, v_{g} \) and 2 scalars \( \rho_{n} \) and \( \rho_{v} \)) to a three-dimensional output vector \( E \) representing the point of gaze in space. In fact, since we are specifically interested only in the horizontal and vertical offset of the user’s point of gaze from some origin on a theoretically flat surface (e.g., a computer monitor), a mapping that provides a two-dimensional output vector would suffice.

The SPORE (Space Partitioning, self-Organizing, and dimensionality REducing) approximation method [61] has been explored in this thesis to investigate its usefulness in providing such a mapping. The SPORE approximation uses a non-parametric regression methodology to construct a predictive model for mapping the input space to the output values, based on a set of learning examples. In particular, we construct two approximations (one for the horizontal and one for the vertical offset of the point of gaze), each using all eight inputs, and each one producing a single output.

In the SPORE approximation, the model is constructed incrementally, adding small, low dimensional parametric building blocks or functional units one at a time. At each step, the outputs of the previously added blocks are used as inputs to the new ones.
Figure 3.7 shows an example of a model constructed by the SPORE approximation. The model is provided with \( L \) inputs \((x_0, x_1, x_2, \ldots, x_L)\), and produces a single output \( \hat{f}(x) \). The building blocks are functions (e.g., two-dimensional polynomial functions). As shown in the figure, the output of each function is used as part of the input to a function in the next stage, thus producing a cascading effect. The output is then calculated by taking a linear combination of the functions. The particular form of the functions (e.g., the coefficients for polynomial functions) and the weights used in the linear combination are determined using a set of learning examples. A learning example consists of a set of input values with a corresponding, predetermined output value \( f(x) \). The functions and weights are chosen to minimize the difference between \( \hat{f}(x) \) and \( f(x) \) for all the learning examples provided.
The SPORE approximation has three main advantages that make it very attractive for this particular application:

1. It requires a relatively small number of learning examples. Other non-linear approximation methods such as artificial neural networks often require a large number of learning examples to provide a stable and robust solution.

2. It is computationally feasible and relatively inexpensive. This is important since we want to track the point of gaze in real-time.

3. The design of the model (i.e., the structure of the building blocks or cascading functions) is implicitly part of the approach, making the learning effectively unsupervised. Further, the order in which input variables are added does not affect the performance of the approach. Therefore, the approach requires minimal user interaction beyond the learning examples provided.

Grudic [61] compares the SPORE method to a number of other approaches, none of which perform as well while maintaining the advantages of the SPORE approximation described above.

In order to obtain the required set of learning examples, the user can be presented with a set of points on the screen (whose coordinates are known) and asked to look at them. As the user's head and eye are tracked, the values for \( v, v, v, \rho, \rho \) are measured and used as input values. Since the point of gaze coordinates (i.e., the desired output values) are known, those coordinates provide output values for the model (i.e., \( f(x) \)). Together, each set of ten (10) values provides a learning example. Each point is shown to the user individually until a sufficient number of learning examples is obtained.
In theory, this learning stage only need occur once, and should not need to be performed on a per-user basis. Uniform noise in the measured input values should not affect the performance of the approximation, as it would form part of the model. This approach could therefore potentially provide a more robust way of calculating the point of gaze than direct calculation.

Note, however, that it will need to be performed every time the apparatus is moved in relation to the monitor, either by moving the eye tracker or the monitor. This is because the model outputs represent the point of gaze relative to the origin of the monitor screen’s plane. By moving the eye tracker or the monitor, the effective coordinate of that origin is changed, thus requiring a new model to be constructed.

3.5 Calibration

Clearly, whether a direct or approximation approach is taken to point of gaze calculation, some degree of system calibration is required. At the very least, calibration procedures are required to determine the spatial relationship between the monitor and the eye tracker. In the case of the indirect approach, a simple yet extensive calibration procedure is required in order to map the information gleaned from the raw images and the angle sensor readings to a point of gaze on the computer screen. In the case of the direct approach, the accuracy of the system depends largely on the accuracy of the image processing performed on the images from the two cameras. In essence, we are inferring three-dimensional information from a pair of two-dimensional images and the spatial relationship between the two cameras from which the images are obtained. The accuracy of such an inference depends, in turn, largely on the accuracy of the parameters used to
model the cameras. This modeling of camera parameters is often referred to as camera calibration.

In the context of three-dimensional machine vision, camera calibration involves modeling one or both of two sets of parameters: the internal geometric and optical characteristics of the camera (often referred to as intrinsic parameters), and the three-dimensional position and orientation of the camera frame relative to a certain world coordinate system (extrinsic parameters). In a typical camera model, there is at least one and at most an infinite number of intrinsic parameters, and exactly six (6) extrinsic parameters. The precise number of intrinsic parameters depends on the accuracy of the model required. For each of our two cameras then, we need to determine the values of both the intrinsic and extrinsic parameters.

Figure 3.8 Camera model
Figure 3.8 illustrates the relationship among the model parameters described above. The intrinsic parameters account for the optical characteristics of the lens and image sensor and positioning of the lens on the image sensor. The optical characteristics of the lens include its focal length and distortions caused by lens aberrations. The intrinsic parameters can thus be used to map the image pixel coordinates \((u',v')\) in Figure 3.8 to a 3-D vector pointing toward the physical object being imaged, given in the camera’s frame of reference. The extrinsic parameters consists of three rotation angles and a 3-D translation vector, which are then used to form a transformation of the 3-D vector in camera coordinates to a 3-D vector in some world reference frame.

In order to determine appropriate values for the parameters in the model, a calibration procedure is typically carried out. First, a set of object points whose 3-D world coordinates are accurately known are imaged by the camera, and the corresponding image pixel coordinates recorded. Each physical object point \(P_w\) can be related to its corresponding recorded image pixel coordinate \((u',v')\) as follows:

\[
P_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}
\]

\[
P_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R P_w + T
\]

\[
x_u = x_c / z_c
\]

\[
y_u = y_c / z_c
\]

\[
x_d = D_x(x_u, y_u)
\]

\[
y_d = D_y(x_u, y_u)
\]

\[
\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = K \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix}
\]
where: 

\( P_c \) is the 3-D camera coordinates of the physical object

\( R \) is a 3x3 rotation matrix used to map points given in the world reference frame to the camera reference frame

\( T \) is a 3-D translation vector used to map points given in the world reference frame to the camera reference frame

\( x_u, y_u \) represent the coordinates of the projection on the \( z_c=1 \) plane of the vector from the origin of the camera coordinate system to the object

\( D_x(), D_y() \) are non-linear, two-dimensional polynomial functions that model the distortion caused by the camera lens

\( K \) is a matrix that performs a projective mapping of the distorted camera coordinates of vector pointing towards the object onto the image plane

\( f_x, f_y \) are the effective focal lengths given in both the horizontal and vertical directions, respectively (note: in many cases, these are modeled by a single focal length \( f \) as shown in Figure 3.8)

\( c_x, c_y \) represent the coordinates of the principal point or centre of the image plane

\( \alpha \) is a parameter used to model the angle between the horizontal and vertical axes of the image sensor pixels

\[
K = \begin{bmatrix}
    f_x & \alpha \cdot f_x & c_x \\
    0 & f_y & c_y \\
    0 & 0 & 1
\end{bmatrix}
\]
If no lens distortion is required in the model, then \( D_x(x_u, y_u) = x_u \) and
\( D_y(x_u, y_u) = y_u \). However, in practice, a polynomial function is chosen for each. For example, for a sixth-order model:

\[
D_x(x_u, y_u) = \left(1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6\right) x_u + 2\kappa_3 x_u y_u + \kappa_4 (r^2 + 2x_u^2)
\]  

(16)

\[
D_y(x_u, y_u) = \left(1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6\right) y_u + \kappa_3 (r^2 + 2y_u^2) + 2\kappa_3 x_u y_u
\]  

(17)

\[
r^2 = x_u^2 + y_u^2
\]  

(18)

where: \( \kappa_1, \kappa_2, \kappa_3, \kappa_4, \kappa_5 \) are distortion parameters of the model.

Thus, in this case, there are six (6) extrinsic parameters (used to define \( R \) and \( T \)) and ten (10) intrinsic parameters (\( \kappa_1, \kappa_2, \kappa_3, \kappa_4, \kappa_5, f_x, f_y, c_x, c_y, \alpha \)). Therefore, at least sixteen (16) calibration points (pairs of \( P_u \) and \( (u', v') \)) are required.

Zhang [58] and Heikkilä and Silvén [59] give examples of how the intrinsic and extrinsic camera parameters can be determined using relatively simple experimental procedures and numerical approximation methods. Section 4.8 describes the procedure we have used to try to calibrate the cameras used in the GTD eye tracker. We found that it was very difficult to accurately calibrate the NA camera primarily due to its high focal length.

Instead, a different approach to calibration is proposed. For a direct approach to point of gaze calculation, in addition to camera calibration parameters for each of the two cameras, we need to determine \( H \) (the origin of the computer monitor plane). Assuming an accurate measure of some feature present in both cameras’ fields of view is available, all the model parameters required can be determined simultaneously. An example was noted toward the end of Section 3.4.1. In this case, the centre of a pupil can be tracked.
and measured accurately in both images. The user can be presented with a series of points on the screen whose coordinates (i.e., offsets from H) are known. For each such target point, the corresponding pupil centre coordinate in each of the two images can be recorded. Once a sufficient number (greater than or equal to the number of model parameters being calibrated) of target and pupil pairs are collected, bounded non-linear approximation methods can be employed to determine all the model parameters simultaneously. This analysis can be done offline and, in theory, only when the apparatus is moved in relation to the monitor, as in the case of the indirect approach.

The advantage of calibrating parameters for a direct approach is that, in theory, the direct calculation can be implemented very efficiently, whereas the indirect or approximation approach would typically require the evaluation of complex polynomial functions that may be computationally expensive. In addition, a direct calculation approach provides more insight into the particular results achieved and the corresponding errors. In particular, the indirect approach provides only the point of gaze on the computer screen, and the angle of gaze is not available. Therefore, it is impossible to gauge the resolution or accuracy of the approximation method used in terms of visual angle.
4 System Implementation

The previous chapter presented the theory and design behind a novel eye tracking system. This chapter is intended to describe the implementation of that theory and design as applied to a new user interface device called GTD (Gaze Tracking Device). Figure 1.6 shows the GTD system in operation. It consists of hardware used to capture relatively high resolution images of a user's pupil, and software to analyze the acquired images, detect fixations, and consequently infer the user's point of gaze (on a computer monitor) and possibly the angle of gaze. It does not require a separate computer, and employs minimal custom electronics. The cost of the present hardware is low (under $2500US for all the parts including the cameras and motors), and the software has been implemented in real-time on a typical IBM-compatible computer performing other tasks simultaneously. Considering that similar motors and cameras have been used in computer mice (e.g., optical mice available from companies such as Logitech), the manufacturing cost in large quantities could conceivably be very low.

The GTD system uses infrared lighting and the image difference method to provide insensitivity to changes in lighting. It uses special image processing software to allow it to accurately track the user's point of gaze, even if the user is wearing eyeglasses or contact lenses. In addition, since it only uses information about a user's pupil, it works equally well for a variety of users with different facial features. Custom image processing software and a robust ellipse-fitting algorithm are used to handle different shaped eyes, and to maximize its insensitivity to occlusions of the pupil (e.g., caused by blinking or squinting).
Sections 4.1 and 4.2 provide overviews of the hardware and software components of the GTD system, respectively. The subsequent sections provide more detail about each of the various components of the system.

4.1 Hardware overview

Figure 4.1 shows the relationships among the various hardware components in the GTD system. At the heart of the system is a pair of IEEE 1394 (FireWire) cameras mounted on a common metal frame as shown in Figure 1.5, stepper motors to orient the NA camera toward the user’s pupil, and some minimal control electronics. This section provides a brief overview of the physical components of the system, while Sections 4.3, 4.4 and 4.5 describe the details of the mechanical, optical and electronic components of the system.

The control electronics consists of a custom printed circuit board powered by an 18VDC power supply. It controls:

a) the operation of LEDs mounted in front of each camera used to provide appropriate illumination,

b) the movement of the stepper motors used to orient the NA camera

c) the sensing of the orientation angles of the NA camera,

d) the external triggering of both cameras, and

e) most of the communication of the GTD system with the PC via a standard serial port on the computer.

The two sets of LED circuits – each with a pair of rings of LEDs – are used to provide the illumination necessary for the image differencing method (see Sections 3.3.1 and 4.6.1).
Figure 4.1 Overview of system components
The rays of light provided primarily by the LEDs are reflected off the pupil and focused onto the image sensors of each camera using appropriate lenses. Infrared filters are also mounted between each image sensor and its corresponding lens. These filters only allow light in the near-infrared section of the electromagnetic spectrum to pass through, thus making the image processing much more robust with respect to changes in lighting. Note that each camera’s image sensor is sensitive to infrared lighting. This is important because some camera manufacturers specifically use sensors that are designed to block infrared light, since for applications that use visible light, infrared lighting can degrade the quality of the image.

Each camera communicates with the computer (PC) via an IEEE 1394 interface card installed in the PC. The interface is used to control various aspects of the cameras, and to send images from the cameras back to the PC at a rate of 400Mb/s.

The NA camera is oriented toward the pupil via stepper motors controlled by special electronics. The current orientation angles of the NA camera are also sensed using A/D converter electronics on the custom PCB, which provide readings of potentiometers (servopots) attached to the apparatus at appropriate locations (see Figure 1.5).

The A/D circuitry is controlled via signals provided by the PC to the control electronics through the serial port. Likewise, commands to move the stepper motors are provided by the PC through the same serial port. Note from Figure 4.1 that the location of the pupil from the WA image as estimated by the system software is used to determine the appropriate commands to control the stepper motors (see Section 3.2.2).
4.2 Software overview

The physical components of the GTD system provide the main input (the raw images of the pupil) to the software. Part of the output of the software is, in turn, used to control the hardware.

Figure 4.2 illustrates the flow of data through the various software components of the system. The software can be divided into two major categories: code used to detect and parameterize eye features (pupil and glint locations), and code used to estimate the direction and point of gaze using the eye feature parameters.

Preliminary image processing (as described in Sections 4.6.1 and 4.6.2) is performed on the raw images provided by the cameras. The pair of pre-processed images at each frame is then used to detect the pupil in both cameras and the glint in the NA camera (see Section 4.6.3). The detected features are then modeled as ellipses, and the centre of the best fitting ellipse is determined in each case (see Section 4.6.4). This results in three pairs of 2-D ellipse centres, thereby providing the main six parameters required for the gaze calculation ($u_H, v_H, u_N, v_N, u_G$ and $v_G$ as described in Section 3.4).

A spatio-temporal filter is applied to each pair of 2-D ellipse centres (see Section 4.6.5). The filtered coordinates, which directly correspond to vectors pointing from the image sensors to the feature each represents, are then applied as inputs to gaze calculation algorithms as described in Sections 3.4 and 4.7. Any calibration data (see Sections 3.5 and 4.8) are applied at this point as well to produce an estimate of the point of gaze and, in the case of the direct approach, the angle of gaze.
Figure 4.2 Overview of system software
4.3 Mechanics

Most of the electronics designed for the eye tracker presented in this thesis is for controlling and measuring the orientation of the mechanical system (see Figure 1.5) used to track a user's head in space so that an accurate image of the user's eye can be obtained.

Figure B.1, Figure B.2 and Figure B.3 in Appendix B show different views of the entire apparatus with the main physical measurements. A common base is provided to position the centre of the two camera sensors a fixed distance apart. It is also designed so that the physical centre of the two camera sensors are on a common axis that passes through the centre of the shaft of the motor that rotates the frame and the centre of the shaft of the corresponding potentiometer (the x-axis in Figure 1.5). The distance between the two sensor centres is 307mm.

The WA camera is held in place by manually adjustable knobs A and B as shown in Figure B.1. Built-in protractors are provided to give an estimate of the orientation of the WA camera.

A frame (see Figure 1.5) holds together the following components:

- the NA camera and lens
- the LED array and associated circuit for the NA camera
- a mirror used to provide virtual rotation of the NA camera in the horizontal direction
- the stepper motor (the "mirror motor") and potentiometer used to control and measure the angle rotation of the mirror.

Thus, when the stepper motor (the "frame motor") whose shaft is coaxial with the x-axis of our world coordinate system rotates the frame around the x-axis, all the
components — including the NA camera and mirror — are rotated simultaneously.

Together, the frame, the mirror and the two motors provide a means of effectively rotating the NA camera’s optical axis in two directions: about the x and y axes (horizontal and vertical axes, respectively in the front view). Note that the centre of rotation for both rotations is the point M as shown in Figure 1.5.

4.3.1 Pan/tilt motors

The actual rotation of the frame and mirror are performed by two stepper motors. Stepper motors were chosen to simplify the control interface. Appendix C describes in detail calculations performed to determine the torque requirements of the motors.

It was found that the angular speed of both motors must be at least 0.5818 rad/s. Calculations further showed that one of the motors needs to be able to provide $1.2 \times 10^{-3}$ Nm to rotate the mirror at the required speed, and the other needs to be able to provide $14.1 \times 10^{-3}$ Nm to rotate the frame and everything attached to it. Finally, it was found that both motors must have a resolution of 0.054° or better. Having calculated the torque and angular resolution requirements of our apparatus, we then proceeded to choose the appropriate motor and gearhead combinations.

For both motors, we chose the AM1524 motor from MicroMo Electronics Inc. For the mirror motor, we combined it with a MicroMo 15/8 (262:1 gear ratio) zero backlash gearhead. These motor/gearhead combinations, when operated at speeds of 582 steps/sec, provide sufficient torque and angular resolution to meet our requirements as shown in Appendix C.

It should be noted that we neglected several small sources of inertia for the frame. Most of these were not expected to exceed the torque capacity of the frame motor and
gearhead at 582 steps/sec. However, since some of the components the frame motor rotates (such as the camera, LED circuit and the potentiometer) are connected to electronics via cables, tension from those cables in certain cable positions could cause the motor to skip steps. Therefore, we decided to operate the frame motor at 450 steps/sec. The consequence of this was that we could only move the camera at 25.7°/sec in the vertical direction, which in turn meant we could only handle head movements as fast as 77°/sec in the vertical direction. This was found to be a high enough speed for most natural head movement.

4.3.2 Angular Position sensors

One of the advantages of stepper motors is that they can be operated without feedback. That is, under normal operation, since they are rotated in "steps", we can predict the angle the motors will turn given the number of pulses we apply. However, feedback sensors are still useful (and in our case required) for two reasons. First, they provide a way of measuring the absolute angular position of the motor shaft and any items attached to it. Second, they can be used to ensure that the gears attached to the motor do not slip due to insufficient torque. That is, if we can predict where the motor will be after applying a certain number of pulses, then we can take a measurement from our sensors after applying the pulses and see if the motor shaft is where we thought it would be.

In the system described in this thesis, we have attached simple potentiometers to the axis of rotation of each of the frame and mirror motors. These are identified as "servopots" in Figure 1.5. The potentiometer providing the angle of rotation of the frame \( (\rho_y) \) is attached to the opposite side of the frame from the frame motor, with its shaft
inserted into the frame along the same axis as the axis of rotation of the frame motor's shaft. The one for providing the angle of rotation of the mirror (\( \rho_n \)) is attached to the shaft that rotates the mirror.

For each potentiometer, the voltage from the wiper terminal is measured using a simple A/D circuit (see Section 4.5.2, Figure 4.6 and Figure A.2). The voltage is then translated to an angle. At system start-up, the absolute angular position of each motor shaft is measured to provide a point of reference. Then, every time the motors are moved, the voltages from the potentiometers are measured to ensure that they have moved the attached items to the angular position intended.

We used standard 5k\( \Omega \) servo-mount plastic potentiometers (Spectrol model 158-146-02). Wirewound potentiometers were found to not be suitable due to their inherently limited resolution.

4.4 Optics

The GTD system uses two cameras to track the pupil and point of gaze of a user. It makes use of special properties of the human retina with respect to infrared light (see Sections 3.2.1 and 3.3.1). The infrared lighting for each camera is provided by two rings of LEDs (see Section 4.5.1).

The following sections describe the cameras, lenses and filters needed to acquire the images of the pupil used to infer the user's point of gaze using the image difference method.

4.4.1 Cameras

The GTD eye tracking system was designed as a low-cost device for human computer interaction. As such, we wanted to keep the cost of the parts as low as possible, use
existing technology and PC interfaces whenever possible, and choose parts that are as small and lightweight as possible.

For the cameras, this meant that we wanted cameras that are equipped with a standard digital interface. In particular, we wanted to avoid using a frame grabber for a number of reasons. First, frame grabbers add to the cost of the overall system. Second, there is no standard software interface for frame grabbers, whereas support for standard digital interfaces such as IEEE 1394 or USB are now built in to most PC operating systems such as Windows 2000 or Windows XP. Third, many frame grabbers introduce noise or distortion in the image. Finally, frame grabbers cannot be used on laptops, whereas cards are readily available for standard digital interfaces such as IEEE 1394 and USB. As mentioned previously, the speed and ready availability of IEEE 1394 cards made that standard an ideal choice for a good camera interface. Therefore, we chose cameras that have a built-in IEEE 1394 interface.

We also wanted the cameras to provide an external triggering mechanism. This is important for synchronization with the switching of the rings of infrared LEDs (see Section 4.5.2). It was also important that the cameras be relatively compact and lightweight. The size and weight affect the torque requirements of the motor used to rotate the frame. In addition, we wanted to keep the overall device as small as possible. We also wanted cameras that can capture images at a rate of at least fifteen frames per second (15 fps) (see 3.2.1 for details on this requirement).

We were particularly interested in cameras that could handle lenses of varying size and focal length (e.g., any standard C-mount or CS-mount lens). Finally, it was important that the manufacturers of the camera provide good support for their product,
including a software application programming interface (API) for controlling and
acquiring images from the cameras via the IEEE 1394 interface.

One product we found to meet all the above requirements for a reasonable price is the
DragonFly camera (see Appendix D for specifications and details) from Point Grey
Research, Inc. Both the NA camera and WA camera are greyscale versions of the
DragonFly equipped with different lenses.

### 4.4.2 Lenses and filters

Aside from the camera itself, the system needed special lenses for each camera to provide, on the one hand, a wide enough field of view to maintain tracking of the pupil with large head movements, and on the other, a high enough resolution to provide an accurate estimate of the 3-D pupil location. In addition, filters that pass infrared light and block other light in the visible spectrum were used to help provide a crisper image.

#### 4.4.2.1 WA camera lens

For the WA camera, we required a lens that has a wide field of view. The Computar T0812FICS-3 lens used in the GTD system has a focal length of 8mm. Figure 3.5 shows a diagram of the pinhole model of a camera. Using this model and similar triangles, we obtain the following formula:

\[
\frac{D}{z} = \frac{d}{f}
\]

Thus, at a distance of one meter from the CCD sensor, the Computar lens provides a field of view of 600mm horizontally and 450mm vertically. If a user's head is 20cm wide, this provides a large area over which the head can be positioned with the pupil still within the field of view of the WA camera.
4.4.2.2 NA camera lens

For the NA camera, we wanted a small enough field of view to provide the maximum resolution possible for our image. At the same time, we wanted to ensure that the entire pupil and glint would still fall within the camera’s field of view. For the user with a 20cm wide head, a field of view of 10cm horizontally (half the width of the head) would ensure that generally a single eye is visible in the NA camera’s image at any given time, should the camera be oriented properly. A lens with a focal length of 48mm would provide this field of view.

The Pelco 13VA5-50 varifocal lens used in the GTD system provides an adjustable focal length of between 5mm and 50mm. At 50mm, this provides a field of view of 5.3° horizontally and 4.1° vertically or a range of 96mm horizontally and 72mm vertically at a distance of 1m.

4.4.2.3 Infrared filters

Aside from the lenses themselves, the GTD eye tracker uses infrared filters to minimize image artefacts caused by ambient light. Note that there is no filter that can pass enough light to detect the reflection of the light from the LEDs off the pupil and cornea but block all ambient infrared light. Therefore, for environments with high levels of ambient infrared lighting, the filter provides limited protection, and it is possible that the eye tracker would have difficulty locating the pupil. For example, on a sunny day, the eye tracker is not expected to work well outdoors. In practice, we have found the GTD system able to locate the pupil and glint even with high amounts of ambient infrared lighting.

The particular filter used in the GTD system is an ILFORD SFX A infrared gel filter. This filter passes infrared light with wavelengths longer than about 875nm. Any similar
gel filter that works around the 875nm wavelength range would probably work. Small pieces have been cut out of the gel filter and carefully placed between the image sensor and lens of each camera. This encloses the filter and, like the image sensor, is thus protected by the lens enclosure from dust and scratches.

4.5 Electronics

We chose cameras with an IEEE 1394 ("FireWire") interface to keep the costs low. IEEE 1394 interface cards are inexpensive, provide a wide bandwidth and are becoming increasingly common on PCs. The cameras also came with appropriate drivers and a software application programming interface (API). This avoided the need to buy or build special image acquisition hardware or software. Note that we did not choose USB because of its low bandwidth, and USB2 cameras were not available at the time.

We attempted to use a minimum amount of custom electronics. We built three circuits in total. Two of the circuits are identical, placed in front of each camera and hold and control the two rings of infrared LEDs. The third circuit is used to communicate with the serial port, control the infrared LED circuits, provide external triggering signals for the cameras, and control the stepper motors that rotate the NA camera and mirror. The following sections provide detail on each of the two circuit designs.

4.5.1 Infrared LED circuit

In order to produce the "dark pupil" and "bright pupil" images, we needed to provide special infrared lighting. One may be concerned about the safety of using infrared LEDs.  

---

6 Note that we opted to use the serial port for all our communication with the custom electronics. This was because all the integrated circuit (IC) components in our circuits (such as the analog-to-digital chips or the motor controllers) are off-the-shelf packages that work with standard serial signals from a PC. There are no equivalent inexpensive components available that work directly with IEEE 1394 signals. In addition, it is much easier to communicate with a PC's serial port than its parallel port from some operating systems such as Microsoft Windows 2000 or Microsoft Windows XP.
We found no evidence in the literature that using infrared LEDs would cause any damage. The Commission Internationale de l'Eclairage (CIE or International Commission on Illumination in English) is, according to their website (http://www.cie-usnc.org), “recognized by the International Standards Organization (ISO), the International Electrotechnical Commission (IEC), and the European Standards Organization (CEN) as the International Standards writing organization responsible for standards in the science of light such as colour and vision and all the forms of lighting.” During their 2001 Expert Symposium on LED Measurement, presentations such as those by David H. Sliney and Bruce E. Stuck indicated that even long-term exposure to an infrared LED at close distances (10cm) is well within the safety limits required to avoid retinal injury.

We built a special circuit and placed one in front of each camera as shown in Figure 1.5. An overview of the operation of the circuit is shown in Figure 4.3, and further details are given in Appendix A.

![Figure 4.3 Operation of infrared LED circuit](image)
On each circuit, one ring (the "inner ring") of infrared LEDs was placed as close as possible to the optical axis of the camera. When all the LEDs in this ring are turned on, they cause any pupil in the field of view of the camera to appear as a bright disk. The other ring ("the outer ring") was placed far enough away from the optical axis so that the pupil appears as a dark disk. The rest of the circuit (shown in Figure A.1 and described further in Appendix A) provides the power and control signals to switch between one ring and the other. It was designed so that only one ring could be on at any given time.

Note that since we used two cameras, each with its own pair of infrared LED rings, light from one camera’s LEDs affects the other camera’s image. However, this "inter-camera interference" simply contributes to the ambient infrared light as seen by the other camera, and does not affect the performance of any of the image processing algorithms.

4.5.2 Control and interface circuit

Aside from providing a control signal for switching between LED rings, we also needed to ensure that the timing of the camera sensor’s image acquisition process is synchronized with the switching of the LEDs. That is, we do not want to start the integration of the image by each camera until the corresponding LEDs are fully lit.

![Figure 4.4 Synchronized camera triggering and LED control](image-url)
Figure 4.4 shows a timing diagram indicating that the triggering of the camera is synchronized properly with the switching of the LED circuit between the inner and outer rings. Figure 4.5 shows what would happen if the two signals were not synchronized. In that case, part of an image may be acquired while the inner ring is turned on and the rest while the outer ring is on. To ensure that the timing is as shown in Figure 4.4, we provide, via the PC, a signal that triggers each camera's image acquisition process.

In addition to the four logic control signals described above, the PC is also used to control the rotation angles of the mirror and frame shown in Figure 1.5. Figure 4.6 gives an overview of a circuit used in our system to provide the LED control signal, drive two stepper motors used to rotate the mirror and frame, and to trigger the cameras. The schematic for the circuit is shown in Figure A.2 and further details about the circuit are found in Appendix A.
4.6 Pupil and glint tracking

The images obtained from each of the two cameras need to be processed and analyzed in order to estimate the projection of the centres of the pupil and glint in each image. For the WA camera, this allows us to properly position the NA camera as described in Section 3.2.2, and for the NA camera, this gives us the set of parameters required to calculate the user’s point of gaze, as described in Section 3.4.

Generally, the steps performed in both cases are:
1. Perform some pre-processing and calculate an aggregate difference image of the last few frames.

2. Threshold the aggregate image to obtain a black and white image of "blobs".

3. Trace the contours of the blobs to find the one that represents the pupil.

4. Fit an ellipse to the pupil to find its size and centre.

5. Apply a spatio-temporal filter to the last few pupil locations to provide smooth tracking of the pupil.

6. For the NA camera, find the glint.

The following describes the above steps in more detail. Note that in this section we assume the pupil and glint are both present in the image being examined.

### 4.6.1 Pre-processing and image differencing

The first step in detecting the centres of the pupil and glint is to perform some pre-processing on the input image and apply the image differencing operation described in Section 3.3.1.

We first remove noise from each image by using pyramid decomposition. We down-sample the image by applying a 5x5 Gaussian filter to the image and rejecting even numbered rows and columns. Then we up-sample the resulting image by injecting zeros at even numbered rows and columns and applying a 5x5 Gaussian filter. Figure 4.7 shows a raw image with a dark pupil acquired from the NA camera, and Figure 4.8 shows the same image with the noise removal applied.
The image differencing is performed by keeping track of which image has a dark pupil and which one has a bright pupil. We calculate an “aggregate difference image” as follows:

\[
D_k = \frac{1}{T} \sum_{m=k-T+1}^{k} \Delta I_m, \quad k > T - 1
\]  

(20)

\[
\Delta I_k = \begin{cases} 
\max(I_k - I_{k-1}, 0), & I_k \text{ has a bright pupil} \\
\max(I_{k-1} - I_k, 0), & I_k \text{ has a dark pupil}
\end{cases}
\]

(21)

where \( k > 0 \) is the frame number,

\( T \) is the size of a temporal window (expressed in number of frames),

\( D_k \) is the \( k^{th} \) aggregate difference image (same dimensions as \( I_k \)),

\( I_k \) is the \( k^{th} \) raw image with noise removal applied,

\( 0 \) is an image (same dimensions as \( I_k \)) with all pixels set to zero, and

\( \Delta I_k \) is the instantaneous difference image (same dimensions as \( I_k \)).

We found that a temporal window of three (3) frames \( (T=3) \) was sufficient to filter out motion artefacts when dealing with fixations. Figure 4.10 shows the result of
subtracting the image in Figure 4.8 from the image from the NA camera with a bright pupil shown in Figure 4.9.

![Figure 4.9 Image from NA camera of bright pupil](image)

Figure 4.9 Image from NA camera of bright pupil

![Figure 4.10 Difference image for a single frame](image)

Figure 4.10 Difference image for a single frame

![Figure 4.11 Aggregate difference image for three frames](image)

Figure 4.11 Aggregate difference image for three frames

Once a complete aggregate difference image has been assembled, all the subsequent processing and analysis (with the exception of finding the glint) is performed on the aggregate difference image. Figure 4.11 shows such an example of an aggregate difference image. The pupil can be easily identified in Figure 4.11 as a bright disk. In
some cases, the pupil in the aggregate difference image is not as clear as in the instantaneous difference if the user is in the process of moving his eye. However, even in the presence of such movement, the time windowing prevents the system from trying to track the movement since we are only interested in fixations (see Section 1.2).

4.6.2 Thresholding

The next step, then, is to threshold the aggregate image to get a black and white image with several “blobs”, one of which will represent the pupil.

For the WA camera, we track both pupils so we can estimate the depth of the person’s face. In this case, the pupils generally occupy a very small portion of the image acquired (see Figure 4.12). Therefore, it is not always possible to segment the pupil blobs from other blobs (such as those caused by motion or ambient lighting differences between the bright pupil and dark pupil images). Instead, we apply some image processing and analysis techniques to identify the pupil blobs.

We first apply a Laplacian of Gaussian (LoG) operator to highlight edges in the aggregate image. Specifically, we use a 5x5 Gaussian filter followed by a 5x5 Laplacian filter, which yields an image where the pupils appear much brighter than most other objects in the scene. An example is shown in Figure 4.13. The resulting image is then thresholded at half the brightest possible pixel value (128 in our case) to obtain a black and white image of “blobs” (see Figure 4.14). We then apply a median filter to the image to take out noisy edge points from the pupil blobs. This consists of setting each pixel in the output image as the median of all pixel values in a 5x5 window with its centre at the pixel coordinate being set. See Figure 4.15 for an example.
Finally, a dilation operator followed by an erosion operator (see [24]) is applied to the filtered blob image to smooth the edges. This process also removes specular reflections from eyeglasses that would otherwise cause problems in our estimation of the centre of the pupil. The dilation operator turns on a pixel in the output image if and only if the pixel or any of its eight neighbouring pixels are on. The erosion operator turns off a pixel in the output image if and only if the pixel or any of its eight neighbouring pixels are off.

Figure 4.16 and Figure 4.17 show the result of a dilation and erosion operation, respectively. Note that in Figure 4.14, Figure 4.15, Figure 4.16 and Figure 4.17, only a small section of the actual image is shown to show the effects of the image processing more clearly.
For the NA camera, we track only one pupil as well as the glint. In this case, the pupils generally occupy a sizeable portion of the image, and are much brighter than anything else other than the glint in the aggregate difference image. We can therefore segment the pupil blobs simply by their size and an appropriate threshold value.

The threshold is automatically set with a simple, efficient histogram-based algorithm. We compute the histogram and set the threshold so that only the top 1000 brightest pixels are set in the output image. Figure 4.18 shows an example of a thresholded image.

Then we perform a single dilation followed by two erosions. The extra erosion removes specular noise introduced by the higher resolution of the NA camera. Figure 4.19 and Figure 4.20 show the result of the dilation and two erosion operations.
respectively on a NA image. Note how this combination of morphological operations significantly reduces the noise blobs seen in Figure 4.18.

![Dilated image from NA camera](image1)

![Eroded image from NA camera](image2)

4.6.3 Selecting the pupil blob

Once we have processed the image to obtain a black and white image with a set of blobs, our next task is to identify the blobs that are most likely to represent a pupil. We begin by tracing the contours for all the blobs in the image. We identify the blobs with the two largest bounding boxes as pupil blobs, since in both the WA camera and NA camera images, we now expect the pupil(s) to be the largest blob(s). We ignore any blobs with fewer than six (6) pixels, since the minimum number of points required to fit an ellipse is six. Also, in the case of the NA camera’s image, we ignore blobs whose bounding box area (width multiplied by height) is less than 180 square pixels. Note that even in the NA camera image, we select the two largest blobs. The reason for this is explained in Section 4.6.6.

In order to make the above algorithm more efficient, once we have found the pupil once, we use a tracking window to limit the portion of the image in which we will look...
for the pupil in subsequent images. We use the size of the ellipse fitted to the pupil itself to determine the size of the tracking window. Specifically, we centre the window at the centre of the last pupil found, and set its size to be six (6) times larger than the ellipse fitted to the last pupil found. We ensure that the window is at least 150 by 150 pixels in the NA camera’s image, and 24 by 24 pixels in the WA camera’s image.

If no suitable match is found within the tracking window, then we look outside the tracking window.

4.6.4 Finding the pupil centre

Once two blobs are identified as candidates for possible pupils in the image, we then proceed to find their centre and, in the case of the NA camera’s image, select the one that is most likely to be the pupil of interest. We do this by fitting the best ellipse (see Section 3.3.2) possible to each candidate blob. The centre is simply the centre of the resulting ellipse (see equations 119 to 125 in Appendix E).

In the case of the NA camera’s image, if we have previously found a pupil centre, we choose as our new pupil centre the blob whose fitted ellipse centre is closest to the last pupil centre found. If we do not have a previous pupil centre, we simply choose the larger of the two blobs.

In the case of the WA camera’s image, although we track both pupils for a rough depth estimation, we use the left pupil for gaze calculation.

4.6.5 Spatio-temporal filtering

If we were to update the fixation point at every frame – even at a rate of eight or nine frames per second – we would run into two difficulties. First, spurious points caused by such things as motion artefacts would adversely affect the ability of the system to
accurately detect the user's point of gaze during a fixation. Second, since we are interested in an eye tracker that tracks fixations for use in a human computer interface, we do not want the gaze point to be updated every time the user searches around the computer monitor. Instead, we want to wait until the user's gaze has been stable over a certain period of time (e.g., the shortest fixation length of 350ms) before updating the gaze point.

To this end, we have included a spatio-temporal filter that we apply to the pupil centre found in each frame. It is a simple filter, whose algorithm is shown in Figure 4.21. Note that in the algorithm, $d_k$ is the square of the displacement of the pupil centre from one frame to the next, B is a buffer (array) whose elements represent the speed at which the pupil is moving (i.e., the difference in pupil centre displacement from one frame to the next), and D is a running sum of the elements of B, indicating the average speed at which the pupil is moving. The variable IsBufferFull indicates whether the buffer B is full (i.e., whether at least T pupil centres have been found so far, where T is an input threshold set by the user).
Inputs: pupil center coordinates from previous frame (x_{k-1}, y_{k-1})
pupil center coordinates from current frame (x_k, y_k)
spatial threshold (S)
temporal threshold (T)

Outputs: filtered pupil center coordinates (x, y)

Initial values: i = 0; IsBufferFull = false; D = 0

\[ d_k = (x_k - x_{k-1})^2 + (y_k - y_{k-1})^2 \]

if IsBufferFull then
    \[ D = D - B[i] \]
    \[ B[i] = |d_k - d_{k-1}| \]
    \[ D = D + B[i] \]
    \[ i = i + 1 \]

if not IsBufferFull or D \( \geq \) (T*S*S) then
    \[ x = x_{k-1} \]
    \[ y = y_{k-1} \]
else
    \[ x = x_k \]
    \[ y = y_k \]
if i > T then
    \[ i = 0 \]
    IsBufferFull = true

Figure 4.21 Spatio-temporal filtering algorithm

The values for the temporal and spatial thresholds (T and S) can be adjusted to allow fine-tuning of tracking tolerances. We have found that values of T = 5 and S = 50 for the NA camera images and T = 2.5 and S = 25 for the WA camera images produce smooth yet responsive tracking of fixations.

At this point, the only further analysis we perform on the data from the WA camera’s image is to estimate the depth of the user’s face from our world coordinate frame. We do this by taking the distance in pixels between the two pupils found and, using the pinhole camera model shown in Figure 3.5, calculate the depth that would result in that pixel distance. Note that for an accurate estimate of depth, this assumes the user’s head is not rotated about the y-axis. The more the head is rotated about the y-axis, the less accurate the estimate, with the estimated depth always being greater than the actual depth.
4.6.6 Finding the glint

One additional task is required to be performed on the NA camera's image once a new stable pupil position is found after applying the spatio-temporal filter described in Section 4.6.5: we must find the location of the glint (see Sections 3.3 and 3.4.1). Note that we have not implemented the multiple light source method of calculating gaze (see Section 0), so we are only interested in the glint caused by the composite reflection of all the LEDs off the surface of the user's cornea.

We look for a glint in the last image we have with a dark pupil. We threshold this image to produce a black and white image containing a set of blobs. The threshold is set by calculating a histogram of the image, and setting the threshold so that only the top 15% brightest pixels are on in the resulting black and white image.

We then proceed to apply the dilation and erosion operators described in 4.6.2 to remove blobs caused by image noise. As with the pupil detection, we then trace the contours of all the resulting blobs and fit an ellipse to all the contours with at least six pixels. We select the blob whose fitted ellipse centre is closest to the centre of the most recent pupil as the glint blob.

At this point, if we find that we mistook the glint blob for a pupil blob in Section 4.6.3, we select the other pupil (the one that is smaller or farther away from the last pupil) as our actual pupil. The outer ring of LEDs produces a slightly larger glint than the inner ring. Therefore, in the aggregate difference image, a small doughnut-shaped "blob" representing the difference in glint sizes is produced. Normally, this blob is much smaller than the pupil blob. However, during a blink or rapid eye movement, it can be larger and may confuse the algorithm used to select the pupil blob. This is why in Section 4.6.3 we keep track of the two most suitable pupil blob candidates.
4.7 Gaze calculation

Once the centre of the pupil in each of the WA camera and NA camera’s images, as well as the location of the glint in the NA camera’s image are estimated, these six values are used along with camera parameters for each of the two cameras to calculate the angle of gaze. For a given location of the computer monitor relative to our world coordinate system, the angle of gaze then gives rise to a particular point of gaze on the screen.

Figure 4.23 shows an overview of the process of calculating the direction and point of gaze directly. The parameters used at each step are described below in Section 4.7.1, and the formulae used to calculate the specific values are those presented in Section 3.4.

Figure 4.22 shows an overview of the process using the SPORE approximation described in Section 3.4.2. Section 4.7.2 describes the process used to arrive at the point of gaze and, in the case of the direct approach, also the angle of gaze.

![Diagram of gaze calculation process](image)

Figure 4.22 Overview of gaze calculation using SPORE approximation
Figure 4.23 Overview of direct gaze calculation
4.7.1 Description of parameters

The direct approach requires the following generally constant system parameters, which can be determined by appropriate calibration techniques.

**WA camera intrinsic parameters:**

- $f_w$ is the focal length of the WA camera.
- $m_w$ is the horizontal pixel offset of the principal point of the WA camera.
- $n_w$ is the vertical pixel offset of the principal point of the WA camera.
- $k_{w1}, k_{w2}, k_{w3}, k_{w4}$ are the distortion parameters of the WA camera for a sixth-order model.

**NA camera intrinsic parameters:**

- $f_N$ is the focal length of the NA camera.
- $m_N$ is the horizontal pixel offset of the principal point of the NA camera.
- $n_N$ is the vertical pixel offset of the principal point of the NA camera.
- $k_{N1}, k_{N2}, k_{N3}, k_{N4}$ are the distortion parameters of the NA camera for a sixth-order model.

Note: once calibrated, the above fourteen parameters should never change unless the cameras and/or lenses are replaced.

Also, $\beta_v, \beta_h$ are the vertical and horizontal angles of rotation of the WA camera with respect to the world coordinate frame. They only change if the WA camera's orientation is manually adjusted.
The position of the upper left hand corner of the computer monitor, \( \mathbf{H} \) (which is a 3-D vector), is also a system constant parameter that only changes if the monitor or apparatus is moved.

Note that, for the indirect approach, the SPORE approximation implicitly models the above system constants, even though it is impossible to extract their values from the approximation. Note also that this is why a new approximation model must be constructed every time the orientation of the WA camera is adjusted or the computer monitor or apparatus is moved.

In addition, both approaches require the following measured input parameters:

- \( u_{wp} \)' is the (possibly) distorted x-coordinate of the pupil centre identified in the WA camera’s image.
- \( v_{wp} \)' is the (possibly) distorted y-coordinate of the pupil centre identified in the WA camera’s image.
- \( u_{np} \)' is the (possibly) distorted x-coordinate of the pupil centre identified in the NA camera’s image.
- \( v_{np} \)' is the (possibly) distorted y-coordinate of the pupil centre identified in the NA camera’s image.
- \( u_{g} \)' is the (possibly) distorted x-coordinate of the glint centre identified in the NA camera’s image.
- \( v_{g} \)' is the (possibly) distorted y-coordinate of the glint centre identified in the NA camera’s image.
\( \rho_v, \rho_h \) are the vertical and horizontal angles of rotation of the frame and mirror with respect to the world coordinate system.

The above eight parameters form the input to both the direct and indirect approaches to gaze calculation.

### 4.7.2 Direct versus approximated approaches

In the case of the direct approach to gaze calculation (see Figure 4.23), the six input parameters that represent the possibly distorted coordinates of the pupil and glint are first undistorted and transformed to the appropriate camera coordinate system, using the relevant camera intrinsic parameters.

In the case of the pupil coordinates in the WA camera image, the resulting vector is then transformed to world coordinates using the WA camera's intrinsic and extrinsic parameters described in the previous section.

In the case of the pupil and glint coordinates in the NA camera image, the vectors in the camera coordinate system are transformed to the mirror's coordinate system (which changes according to \( \rho_v, \rho_h \)). The resulting vectors or "rays" are then traced by reflecting them off the surface of the mirror and rotating and translating the resulting vectors to the world coordinate system.

The four vectors found above representing the pupil (as described in Section 3.4.1) are then used to estimate the 3-D position of the user's pupil. The coordinates of the pupil, along with the two vectors representing the glint are then used, as described in Section 3.4.2, to estimate the direction and point of gaze of the user.
For the indirect approach using the SPORE approximation, all eight input parameters are used as the model’s inputs, and the coordinates of the user’s point of gaze are treated as the model’s two outputs.

4.8 Camera calibration

In Section 3.5, two approaches to calibration were outlined. In this section, a description of the experimental procedures used to perform the calibration is presented.

An initial, traditional camera calibration was performed on each of the NA and WA cameras using a toolbox written for MATLAB by Jean-Yves Bouguet (see http://www.vision.caltech.edu/bouguetj/calib_doc). This toolbox provides an integrated means of inputting images, detecting features (corners in a black and white checkerboard pattern) and estimating both the intrinsic and extrinsic camera parameters. It is based on the work by Zhang [58] and Heikkilä and Silvén [59].

First, a checkerboard pattern was printed on a laser printer (a reduced version of the pattern is shown in Figure 4.26), and the paper was glued onto a wooden board. The board was held at different depths and angles and the images recorded (see Figure 4.25 for an example). The toolbox’s corner detection code was used to detect the corners of the squares in each image, thus providing the calibration points.

For the WA camera, we set the skew parameter $\alpha$ to zero (i.e., we assume rectangular image pixels), ignored the $6^{\text{th}}$-order polynomial term (i.e., set $\kappa_6 = 0$, since it did not seem to contribute to the accuracy of the calibration), and set $f_x = f_y = f_w$. We ran the calibration code in the toolbox and obtained values for the focal length ($f_w$), the principal point of the WA camera image ($m_w, n_w$) as well as the distortion parameters
Note that although the extrinsic parameters are given by the toolbox for each image shown in Figure 4.25, they are given with respect to the particular orientation and position of the checkerboard pattern used in each case, rendering the results useless for our purposes. We are instead interested in the specific angle and position of the WA camera with respect to our world frame of reference, which this procedure does not provide. Therefore, for this procedure, we estimated the extrinsic parameters using the focal length obtained and the protractors built into the apparatus (see Figure B.1 and Figure B.3).

Having obtained the intrinsic parameters, we then addressed the first assumption outlined in Section 3.4.1 by replacing equation 70 (see Appendix E) with the following:

\[
\overline{Q}_{w} = \begin{bmatrix} x_{w} \\ y_{w} \\ 1 \end{bmatrix}
\]
\[\text{(22)}\]

\[
x_{w}' = \frac{u_{w} - m_{w}}{f_{w}}
\]
\[\text{(23)}\]

\[
y_{w}' = \frac{v_{w} - n_{w}}{f_{w}}
\]
\[\text{(24)}\]

where the relationship among \( x_{w}' \), \( x_{w} \), \( y_{w}' \) and \( y_{w} \) is shown in Figure 4.24.

\[\begin{array}{c}
x_{w}', y_{w}' \\
nonlinearapproximation\\
K_{w1}, K_{w2}, K_{w3}, K_{w4}
\end{array}\]

\[\begin{array}{c}
x_{w}, y_{w}
\end{array}\]

\[\text{Figure 4.24 Undistorting image data}\]
A similar procedure was carried out to find new values for $\overline{A}_v$ and $\overline{B}_v$ by calibrating the NA camera. However, finding the appropriate calibration parameters for the NA camera using this approach proved very difficult. The NA camera has a very high focal length (50mm), which magnifies any small errors in the corners detected. In addition, the high focal length also accentuated shadows, further decreasing the accuracy of the corner detection. Since the toolbox relies heavily on the accuracy of the corner detection algorithm, the resulting camera calibration results were not usable for obtaining better estimates for $\overline{A}_v$ and $\overline{B}_v$.

Figure 4.25 Actual images used for WA camera calibration
On the other hand, since we are using two cameras to track a common object (the pupil), it is in theory possible to use some sort of matching algorithm to extract the camera calibration parameters. Specifically, we want to find the values for the camera calibration parameters that minimize the distance between the value for $P$ in equation 4 and the value for $P$ in equation 5. That is, we want to minimize the following:

$$J(x) = |e_1| + |e_2| + |e_3|$$

(25)

$$e_i = s_x + \alpha_s \bar{s}_x - W_x - \alpha_q \bar{Q}_x$$

(26)
This is a multidimensional non-linear optimization problem. We tried three approaches to finding a suitable solution, all with limited success. Upon further examination, we discovered that the function \( J(x) \) is not only highly non-linear, but also not very stable. This means the problem is ill conditioned, and straightforward optimization algorithms are not always able to find a good solution.

As a simplification, for all three approaches, we decided to only use a first-order distortion model. For the third approach, we also decided to calibrate the WA camera’s two orientation angles \( \beta_\nu \) and \( \beta_\mu \). In addition, to account for possible inaccuracies in the measurement of the potentiometer values, we added the NA camera’s two orientation angles \( \rho_\mu \) and \( \rho_\nu \) to the set of parameters to be optimized.

The first approach we used was to use the WA camera parameters found above and optimize the four NA camera intrinsic parameters using a bounded non-linear optimization algorithm. We used the \texttt{fmincon} function in the MATLAB Optimization Toolbox, which employs a Sequential Quadratic Programming method. We bounded the NA focal length (in mm) to be in the range \([45, 55]\), the first-order distortion parameter \( k_{nw} \) to be in the range \([-1, 1]\), the x-coordinate of \( C_N \) (in pixels) to be in the range \([300, 340]\) and the y-coordinate in the range \([220, 260]\).

The second approach we used was to try optimizing all the intrinsic camera parameters, using the same optimization algorithm. We bounded the NA parameters as in the first approach. We bounded the WA focal length (in mm) to be in the range \([7, 9]\), \( \ldots \)
the first-order $\kappa_m$ distortion parameter to be in the range [-1,1], the x-coordinate of $C_w$ (in pixels) to be in the range [300,340] and the y-coordinate in the range [220,260].

For the third approach, we tried optimizing all the intrinsic camera parameters as well as the orientation angles of both cameras, for a total of twelve (12) parameters (two orientation angles for each camera). We bounded the intrinsic camera parameters as in the second approach, $\beta_v$ to be in the range [-2,2], $\beta_h$ to be in the range [28,32], $\rho_h$ to be in the range $[\rho_{h0} - 5^\circ, \rho_{h0} + 5^\circ]$, and $\rho_v$ to be in the range $[\rho_{v0} - 5^\circ, \rho_{v0} + 5^\circ]$, where $\rho_{h0}$ and $\rho_{v0}$ are the angles measured by the potentiometers.

The results of all the attempts at calibration described above are provided in Section 5.4.
5 Results

This chapter provides a report of the performance of the GTD eye tracking system. Specifically, results of experiments to investigate the speed, accuracy, resolution, reliability and cost of various components are provided.

Section 5.1 reports on the overall hardware performance of the system, including its ability to track the eye in the presence of natural head movements, the speed and theoretical accuracy limit of the pupil tracking algorithm, and an estimated cost for the system.

Section 5.2 describes the experiments used and the results subsequently obtained in order to test the ability of the system to detect and parameterize the pupil and glint in the two cameras’ images.

Section 5.3 explains the results of trying to implement various approaches to calculating the angle of gaze and point of gaze. Results of employing both direct and indirect methods are given.

Finally, for the direct method, Section 5.4 explains various calibration techniques and their effect on overall system performance.

5.1 Hardware

We found the motors provided sufficient torque (as described in Section 4.3.1 and Appendix C) to provide accurate movement of the frame and mirror to within 0.057° and 0.036° at speeds of 582 steps/sec and 450 steps/sec horizontally and vertically, respectively. We found that the limit of 77°/sec in the vertical direction was more than sufficient for natural head movements. That is, the system seemed capable of keeping up with natural head movements.
The DragonFly cameras used pose an upper limit on the speed and resolution of the overall system. The cameras have a limit of 15 frames per second (fps) when external triggering is enabled. As mentioned in Section 3.2.1, this is sufficient to detect a fixation of average duration using an “aggregation” time window of five frames. In fact, we found that a time window of three frames was sufficient to eliminate most motion artefacts, making the speed limit of the camera quite acceptable. In addition, the cameras are equipped with sensors that are 4.8mm wide and 3.6mm high. With images that contain 640 pixels horizontally and 480 pixels vertically, that translates to 0.0075mm/pixel in each direction as the maximum sensor resolution. Appendix E describes how this relates to the accuracy of the system in determining the user’s point of gaze. Specifically, with the sub-pixel accuracy of the GTD system, the POG can be calculated to within 13 pixels horizontally (3.7mm) and 8 pixels (2.4mm) vertically on a 15” screen when the user is 50cm away from the screen (the difference in horizontal and vertical accuracy is due to the aspect ratio of the images obtained from the cameras – see Appendix E).

In order to drive the motors, measure the angles of rotation of the frame and mirror, trigger the cameras and switch the LED circuits, a serial connection is used. Currently, the interface to the chip that performs all these tasks is a serial connection operating at 2400bps\(^7\). For each frame, to trigger both cameras and switch the LED circuits, seven bytes must be sent across this interface. This means that 23.3ms are spent on this communication. At the current processing rate of around 9fps, if we could save this time, the rate would automatically increase to 11.4fps.

\(^7\) The speed of the serial communication is limited by the particular ICs selected for controlling the stepper motor.
It is worth noting here the estimated cost breakdown of the system as implemented for this thesis. The DragonFly cameras cost $800US each, and the two lenses total approximately $100US. The motors cost around $250US and the electronics cost roughly $300US for all the components (including the infrared LEDs and filters) and printed circuit boards. Not including the cost of the material for the frame and any labour to assemble the apparatus, this puts the cost at approximately $2,250US. All parts were purchased at retail and in small quantities, both of which add significantly to the cost. As a product, the GTD system could conceivably be produced at an even lower cost. For example, the most costly components are the two cameras. An optical mouse has a built-in digital camera, and the mouse can be purchased for less than $30US retail. Therefore, it is reasonable to expect that the cost of the system could be significantly reduced if it were to be mass-produced. Thus, the goal of implementing a very inexpensive remote eye tracking system that is robust with respect to environmental changes and substantial head movement has been achieved.

5.2 Detection and parameterization

This section reports the accuracy, speed and robustness of the algorithms used for detecting and parameterizing the pupil in both cameras and the glint in the NA camera. We used both synthetic and real data to find the results reported. The tests used to obtain the measurements were performed on a personal computer supplied with an AMD Athlon microprocessor with a CPU speed of 1400GHz and 512MB DDR RAM and running Windows 2000 Professional.

Since an artificial eye was not available, we instead created images that mimic the characteristics of a typical image of the eye under infrared lighting conditions (both with
on-axis and off-axis illumination). Creating such a synthetic data set allowed us to set the expected pupil and glint centres, and thus measure exactly the accuracy of the algorithms.

Figure 5.1 shows the images used to analyze the performance of the algorithms for the NA camera. The top row contains images of a dark pupil in three different locations, and the bottom row contains right pupils in the same locations. In each case, they are eight bit images. We filled each image with a pixel intensity value of 50. We then placed an ellipse representing the sclera in the image, with a pixel intensity value of 200. Next, we placed ellipses to represent the iris (intensity value of 100), pupil (intensity value of 20 for a dark pupil and 150 for a bright pupil) and glint (intensity value of 255). We added Gaussian noise with a range of +/-60 intensity values to the image, and then added a DC noise of 10 intensity values to the whole image. We created four instances of the leftmost pair of images (alternating dark and bright pupil), and five instances each of the other two pairs for a total of twenty eight (28) images, in that order. This simulates a user's eye moving to three different locations, with a moving glint and varying noise.

Figure 5.1 Synthetic images for pupil and glint in NA camera
Figure 5.2 shows similar images used to analyze the performance of the algorithms for the WA camera. The only differences are that we use necessarily smaller sizes for the ellipses (due to the shorter focal length of the WA camera) and include two separate eyes. The right eye is tilted slightly to simulate the effect of a tilted head.

![Synthetic images for pupil and glint in WA camera](image)

Using the synthetic data, we were able to determine the following:

1. For the NA image, we found an average pupil error (distance of pupil centre estimated by the algorithms and the actual pupil centre) of 0.758 pixels or 0.119% in x and 0.492 pixels or 0.103% in y.

2. In the NA image, we found that, for those cases where the glint was detected correctly, there was an average error of 0.0910 pixels or 0.014% in x and 0.324 pixels or 0.051% in y.
3. For the WA image, we found an average pupil error of 0.408 pixels or 0.064\% in x and 0.598 pixels or 0.125\% in y for the left pupil, and 0.651 pixels or 0.102\% in x and 0.578 pixels or 0.102\% in y for the right pupil.

In addition to the synthetic data described above, we obtained images from three different subjects’ eyes. Subject #1 has dark eyes and was not wearing glasses. Subject #2 has light eyes and also was not wearing glasses (see Figure 5.5). Subject #3 has dark eyes and was wearing glasses as shown in Figure 5.6. For Subject #1, we obtained images under two different lighting conditions (one in a room with only fluorescent lighting as shown in Figure 5.3, and one in the same room with sunlight entering the room through a nearby window as shown in Figure 5.4). The reason to include data from a room with sunlight present is that light from the sun contains ambient infrared light, which may interfere with the image differencing method. Indeed, as seen in Figure 5.4, the difference between the dark and bright pupil is far less pronounced than in the other cases, where ambient infrared light does not “wash out” the light from the LEDs. In each case, the user was asked to sit comfortably in front of the computer. The frame and mirror were positioned manually so the user’s left eye was in the field of view of the NA camera. During the experiment, the motors were kept stationary, and the user was asked to keep his/her head relatively still while looking at various parts of the screen. Images from both cameras were then recorded without processing the frames. Processing was then performed off-line in order to analyze the performance of the algorithms.
Figure 5.3 Sample images from Subject #1 without sunlight

Figure 5.4 Sample images from Subject #1 with sunlight
Figure 5.5 Sample images from Subject #2

Figure 5.6 Sample images from Subject #3
Since for real data we cannot predict the exact location of the pupil or glint, it is impossible to report accuracy as described above for the synthetic data. Instead, Table 5.1 reports the following for the pupil in the NA camera:

1. **Reliability**: This is the percentage of frames in which the algorithm calculated the pupil (or glint) as being somewhere inside the actual pupil (glint) and "close" to its centre. This was measured by visually inspecting each frame and the corresponding pupil centre in the WA image and the pupil and glint centres in the NA image calculated by the algorithm for all twenty eight frames in the synthetic case and 204 frames for each of the real user cases. In all cases, a temporal window of three frames was used to detect fixations, and the first four frames of the sequence were excluded from the calculations. These first few frames are excluded because at start-up it takes as many frames as the length of the temporal window to establish that a fixation has been detected. For the pupil (glint) centre, we count the number of frames \( F \) (out of a total of \( N \) frames) where the calculated pupil (glint) centre is within the area of the pupil (glint). We also count the number of frames \( F' \) where the pupil (glint) centre has been within the area of the pupil (glint) at least once over the last \( T \) frames, where \( T \) is the size of the temporal window (three in our case). For each case, two ratios are given. The first is \( F/N \) and the second is \( F'/N \). If the pupil or glint was not visible in a certain frame, it is considered "correct" if the algorithm failed to find the object of interest.

2. **Image differencing**: This is the average time (in milliseconds) taken to perform the image differencing method. It includes the time taken to calculate
the instantaneous difference image, and incorporate it into the aggregate difference image.

3. **Blob image calculation**: This is the average time (in milliseconds) taken to binarize the aggregate difference image. It includes the time taken to calculate an appropriate threshold value, threshold the image and perform the morphological operations (dilation and erosion).

4. **Pupil blob selection**: This is the average time (in milliseconds) taken to select the blob in the binarized image corresponding to the pupil. For the WA image, this is the time taken to find both pupils.

5. **Spatio-temporal filter**: This is the average time (in milliseconds) taken to apply the spatio-temporal filter.

Table 5.2 reports similar results for the pupils in the WA camera. Table 5.3 reports the reliability (as described above) and speed (in milliseconds) of the algorithm used to calculate the location of the glint in the NA camera. Note that for the speed reported in Table 5.3, the first time the glint is calculated it is higher than the reported average value, since the tracking window has not yet been established and the search area is much larger. Also, for the first few synthetic images, before an appropriate tracking window was established, the system was not able to successfully select the appropriate blob for the glint. Since we only had 28 images in total, the reliability of the glint detection algorithm is thus noticeably lower. However, in real cases, those few images do not affect the reliability significantly.
### Table 5.1 Comparison of performance under different conditions for various subjects (Pupil in NA camera)

<table>
<thead>
<tr>
<th>Subject</th>
<th>Eye colour</th>
<th>Sunlight present?</th>
<th>Wearing glasses?</th>
<th>Reliability</th>
<th>Image differencing</th>
<th>Blob image calculation</th>
<th>Pupil blob selection</th>
<th>Spatio-temporal filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic</td>
<td>Black</td>
<td>N/A</td>
<td>N/A</td>
<td>92% / 100%</td>
<td>15</td>
<td>24</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>#1</td>
<td>Brown</td>
<td>No</td>
<td>No</td>
<td>98% / 100%</td>
<td>16</td>
<td>22</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>#1</td>
<td>Brown</td>
<td>Yes</td>
<td>No</td>
<td>97% / 100%</td>
<td>16</td>
<td>23</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>#2</td>
<td>Blue</td>
<td>No</td>
<td>No</td>
<td>80% / 88%</td>
<td>15</td>
<td>24</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>#3</td>
<td>Brown</td>
<td>No</td>
<td>Yes</td>
<td>94% / 99%</td>
<td>15</td>
<td>22</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

### Table 5.2 Comparison of performance under different conditions for various subjects (Pupil in WA camera)

<table>
<thead>
<tr>
<th>Subject</th>
<th>Eye colour</th>
<th>Sunlight present?</th>
<th>Wearing glasses?</th>
<th>Reliability</th>
<th>Image differencing</th>
<th>Blob image calculation</th>
<th>Pupil blob selection</th>
<th>Spatio-temporal filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic</td>
<td>Black</td>
<td>N/A</td>
<td>N/A</td>
<td>92% / 100%</td>
<td>10</td>
<td>13</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>#1</td>
<td>Brown</td>
<td>No</td>
<td>No</td>
<td>98% / 100%</td>
<td>10</td>
<td>13</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>#1</td>
<td>Brown</td>
<td>Yes</td>
<td>No</td>
<td>82% / 95%</td>
<td>10</td>
<td>10</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>#2</td>
<td>Blue</td>
<td>No</td>
<td>No</td>
<td>95% / 98%</td>
<td>10</td>
<td>13</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>#3</td>
<td>Brown</td>
<td>No</td>
<td>Yes</td>
<td>98% / 100%</td>
<td>11</td>
<td>13</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

### Table 5.3 Comparison of performance under different conditions for various subjects (Glint in NA camera)

<table>
<thead>
<tr>
<th>Subject</th>
<th>Eye colour</th>
<th>Sunlight present?</th>
<th>Wearing glasses?</th>
<th>Reliability</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic</td>
<td>Black</td>
<td>N/A</td>
<td>N/A</td>
<td>50% / 58%</td>
<td>3</td>
</tr>
<tr>
<td>#1</td>
<td>Brown</td>
<td>No</td>
<td>No</td>
<td>96% / 100%</td>
<td>3</td>
</tr>
<tr>
<td>#1</td>
<td>Brown</td>
<td>Yes</td>
<td>No</td>
<td>83% / 93%</td>
<td>3</td>
</tr>
<tr>
<td>#2</td>
<td>Blue</td>
<td>No</td>
<td>No</td>
<td>64% / 74%</td>
<td>3</td>
</tr>
<tr>
<td>#3</td>
<td>Brown</td>
<td>No</td>
<td>Yes</td>
<td>90% / 96%</td>
<td>3</td>
</tr>
</tbody>
</table>
It is important to note that Subject #2 made two very large head movements, and the system required a few frames each time to re-establish accurate pupil and glint centre estimates. This makes the reliability measure reported lower, but this is only related to the head movements, and not due to a difference in eye colour. Indeed, if the frames associated with the time the user moved his head are removed from the data, the measures become consistent with the other subjects. If the processing speed and frame rate were higher, the effects of such head motion would be less and the reliability measure higher. Therefore, this is an implementation issue that could be resolved using a faster computer and higher frame rate cameras, and is not related to the system design.

Also, for the first few synthetic images, before an appropriate tracking window was established, the system was not able to successfully select the appropriate blob for the glint, which explains the apparently low reliability of the glint estimate for the synthetic data. Note also that while the reliability of the system to detect the correct pupil and glint at any given instant is lower when ambient infrared light is present, the use of the spatio-temporal filter significantly enhances performance.

In addition to the above results, it is worth mentioning that we have consistently been able to achieve a net speed of approximately nine frames per second. This includes time taken to communicate with the hardware, capture the images from both cameras and apply the image analysis algorithms – all while the same CPU also runs the operating system and possibly other applications. The timing was measured on the computer described at the beginning of this section while it was running the native operating system.
Finally, it must also be mentioned here that the tracking seems to work very well. As long as the image processing determines the location of the pupil in both cameras appropriately, the tracking algorithm positions the two motors correctly so that the left eye of the user always falls within the field of view of the NA camera. The procedure described in Section 3.2.2 for initially orienting the NA camera also seems to work without any problems.

5.3 Gaze calculation

In order to test the accuracy of the SPORE approximation method for calculating the point of gaze, eight datasets were used. All eight datasets were obtained from the same user (Subject #1) and lighting conditions (no sunlight). The computer monitor display was set to a resolution of 1024 pixels horizontally and 768 pixels vertically. We used third order polynomials as the building block functions. Each dataset consists of a training set, an external test set and an off-grid test set as described below.

Figure 5.7 5x5 grid of points used for SPORE training and testing
For the first dataset, the user was presented with a 5x5 grid of points on the computer screen. Figure 5.7 illustrates this grid, except that during the experiment, all the points were represented as red circles. In Figure 5.7, the white circles (which form a 3x3 grid) represent training grid points used to train the SPORE approximation, while the grey squares represent off-grid points used to test its performance for points not on the training grid. Each point was presented independently to the user, and the user asked to fixate on the point visible at all times. That is, at any given time, only one of the twenty-five points was visible.

![Diagram](image)

**Figure 5.8** Formation of data sets for training and testing SPORE approximation using training grid points

Each point was presented to the user until fifteen (15) samples were obtained, where a sample consists of the eight input parameters and two output parameters described in Sections 4.7.1 and 4.7.2. Figure 5.8 illustrates how samples from the 3x3 training grid (the white circles in Figure 5.7) were used to train and test the SPORE approximation.
For each of these training grid points, of the fifteen samples obtained, the first and last five samples were ignored, as the user could have been in the process of moving his eyes to/from the next/previous point. From the remaining five raw samples, two were added to the training data for the SPORE algorithm, one was added to the test data, and one was added to the validation data. All the samples, including the remaining one not added to the training set, were added to a file ("External Test Set") used later for independent testing of the approximation. This produced a SPORE training set of eighteen (2 x 3 x 3 = 18) training samples, nine testing (1 x 3 x 3 = 9) samples and nine validation samples, and an external test set containing 45 (5 x 3 x 3 = 45) samples.

Samples from the other points on the 5x5 grid (the off-grid points represented by grey squares in Figure 5.7) were used to test the ability of the approximation to interpolate between points not on the grid used to train it. Figure 5.9 illustrates how these samples were used to create an off-grid test set. As with the training and external test sets, the first and last five samples for each grid point were ignored. The remaining five samples per point were added to the off-grid test set, resulting in 80 (5 x (5x5 - 3x3)) samples in total.

![Off-Grid Test Set](image)

*Figure 5.9 Formation of data set for testing SPORE approximation using off-grid points*
The above procedure was repeated twice, with each time the user’s head being in a slightly different location. In each of the three trials, the user was asked to keep his head relatively still. In all three cases, the mirror and frame were kept at a constant orientation (mirror rotated to 37.508° away from the positive z-axis towards the negative x-axis, and frame rotated to 30.987° away from the positive z-axis towards the positive y-axis).

Figure 5.10 shows sample images of the NA camera for each of the three trials.

Figure 5.10 Sample images of three trials using 5x5 grid

Figure 5.11 9x9 grid of points used for SPORE training and testing

The fourth dataset was obtained using the same procedure described above, except that a 9x9 grid of points was used (a 5x5 grid of which was used for training), as
illustrated in Figure 5.11. Again, white circles represent training grid points, and grey rectangles represent off-grid points. This time the training set contained 50 (2 x 5 x 5) training samples and 25 (1 x 5 x 5) testing and validation samples each. The external test set contained 125 (5 x 5 x 5) samples, and the off-grid test set contained 280 (5 x (9x9 - 5x5)) samples. Again, the mirror and frame were kept at a constant orientation (mirror rotated to 37.984°, and frame rotated to 32.215°).

An additional three datasets were constructed using the 9x9 grid, with the mirror and frame moved to a different orientation each time. A final dataset was constructed using a 13x13 grid (a 7x7 grid of which was used for training) and the same procedure as described above. In this case, the training set contained 98 (2 x 7 x 7) training samples and 49 (1 x 7 x 7) testing and validation samples each. The external test set contained 245 (5 x 7 x 7) samples, and the off-grid test set contained 600 (5 x (13x13 - 7x7)) samples. Table 5.4 summarizes the eight datasets obtained.

Table 5.4 Datasets used to train and test SPORE approximation

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Grid size presented to user</th>
<th>Grid size used for training</th>
<th>Mirror rotation angle</th>
<th>Frame rotation angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5x5</td>
<td>3x3</td>
<td>37.508°</td>
<td>30.987°</td>
</tr>
<tr>
<td>2</td>
<td>5x5</td>
<td>3x3</td>
<td>37.508°</td>
<td>30.987°</td>
</tr>
<tr>
<td>3</td>
<td>5x5</td>
<td>3x3</td>
<td>37.508°</td>
<td>30.987°</td>
</tr>
<tr>
<td>4</td>
<td>9x9</td>
<td>5x5</td>
<td>37.984°</td>
<td>32.215°</td>
</tr>
<tr>
<td>5</td>
<td>9x9</td>
<td>5x5</td>
<td>38.098°</td>
<td>32.119°</td>
</tr>
<tr>
<td>6</td>
<td>9x9</td>
<td>5x5</td>
<td>38.075°</td>
<td>29.205°</td>
</tr>
<tr>
<td>7</td>
<td>9x9</td>
<td>5x5</td>
<td>37.053°</td>
<td>32.311°</td>
</tr>
<tr>
<td>8</td>
<td>13x13</td>
<td>9x9</td>
<td>38.189°</td>
<td>32.191°</td>
</tr>
</tbody>
</table>

At first, dataset 1 was used for training, and various combinations of test data from datasets 1, 2 and 3 were used to test the resulting approximation. The results are shown in Table 5.5. Along with the raw pixel errors for both the horizontal and vertical.
coordinates in each case, we have given a relative and absolute error. The absolute error is given as:

\[ \varepsilon_h = \frac{e_h}{H} \]  
\[ \varepsilon_v = \frac{e_v}{V} \]  

where \( \varepsilon_h, \varepsilon_v \) are the horizontal and vertical absolute errors, respectively, 
\( e_h, e_v \) are the horizontal and vertical pixel errors, respectively, 
\( H, V \) are the width and height, respectively, of the screen in pixels, and

The relative error is given as:

\[ \varepsilon'_h = \frac{e_h}{H/(N-1)} \]  
\[ \varepsilon'_v = \frac{e_v}{V/(N-1)} \]  

where \( \varepsilon'_h, \varepsilon'_v \) are the horizontal and vertical relative errors, respectively, and \( N \) is the size of the grid used for training (e.g., 3 for datasets 1, 2 and 3).

For example, for a pixel error of 9.97, the absolute error would be 9.97/1024 = 0.0097 or 0.97\%, and the relative error for a 3x3 training grid would be 9.97/(1024/2) = 1.95\%.

The results in Table 5.5 indicate that with only a 3x3 training grid, we are able to achieve reasonable results. Furthermore, training with very few samples produces an approximation that can extrapolate well for input data that looks quite different from those used for training, as shown in the second row, where the external test set from different datasets than that used for training was used to test the approximation. In addition, the approximation can interpolate well for gaze points not included in its training data, as shown in the third row, where an off-grid test set was used to test the approximation.
Encouraged by these results, we then proceeded to use combinations of two and three datasets for training, and all the data from the first three datasets for testing the resulting approximation. Table 5.6 shows the results of these experiments, which indicate that making the training data more representative of the possible combinations of inputs improves the accuracy of the approximation. These findings are confirmed in Table 5.9, where the mean and standard deviation of the gaze error are shown for approximations trained using one, two and three datasets. All the datasets were obtained using a 3x3 training grid. The gaze error is defined as follows:

\[
\varepsilon = \sqrt{(x-x)^2 + (y-y)^2}
\]

where \( \varepsilon \) is the gaze error,

\( \hat{x} \) is the output of the SPORE approximation for the horizontal gaze position for a given sample,

\( x \) is the actual horizontal gaze position for a given sample,

\( \hat{y} \) is the output of the SPORE approximation for the vertical gaze position for a given sample, and

\( y \) is the actual vertical gaze position for a given sample.

In order to investigate the effects of changing the grid size, we used datasets from each of the grid sizes for training, and datasets from different grid sizes for testing. The results shown in Table 5.7 indicate that, in general, increasing the grid size (and hence the number of points used to train the approximation) improves the accuracy of the approximation. Table 5.10 confirms these findings by showing the mean and standard deviation of the gaze error for each of three grid sizes.
Finally, Table 5.8 shows the results of experiments in training the approximation with data obtained using different orientations of the mirror and frame. Table 5.11 also shows the mean and standard deviation of the gaze error for training data obtained from a single mirror and frame orientation, and for training data obtained from three different mirror and frame orientations. These results indicate that using data from different orientations of the mirror and frame also improves the accuracy of the approximation.

In order to test the performance of the direct calculation of point of gaze, we used the known gaze points from a subset of the fourth dataset used to test the SPORE approximation. In particular, we took 160 samples from the training data used to test the SPORE approximation. We began by using “ideal” or “uncalibrated” camera parameters: focal lengths of 8mm and 50mm for the WA and NA cameras, respectively, image centres of (320,240) in image pixel coordinates and zeros for all distortion parameters. We further assumed orientation angles of $\beta_v = 30^\circ$ and $\beta_h = 0^\circ$, and used the measurements from the potentiometers to determine $\rho_v$ and $\rho_h$ (the NA camera was not moved during the experiment).

We calculated values for $\varepsilon_1$, $\varepsilon_2$, $\varepsilon_3$, and $J(x)$ as described in Section 4.8. The resulting $J(x)$ gave us an indication of how well the rays of light producing the pupil centre pixels in each camera’s image line up (a lower $J(x)$ indicates the rays line up better). As a further measure, we calculated an approximation to the 3-D pupil location by averaging the two values for $P$ found in equations 4 and 5, and comparing the resulting depth of the pupil with the depth estimated from the WA camera during tracking as described in Section 3.2.2 and Appendix E.
Table 5.5 Results of training SPORE approximation with a single dataset

<table>
<thead>
<tr>
<th>Training Sets Used</th>
<th>External Test Sets Used</th>
<th>Off-grid Test Sets Used</th>
<th>Horizontal Error (pixels)</th>
<th>Horizontal Error (relative)</th>
<th>Horizontal Error (absolute)</th>
<th>Vertical Error (pixels)</th>
<th>Vertical Error (relative)</th>
<th>Vertical Error (absolute)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>None</td>
<td>9.97</td>
<td>2.0%</td>
<td>1.0%</td>
<td>63.98</td>
<td>16.9%</td>
<td>6.2%</td>
</tr>
<tr>
<td>1,2,3</td>
<td>None</td>
<td>None</td>
<td>339.5</td>
<td>67.0%</td>
<td>33.2%</td>
<td>231.28</td>
<td>61.0%</td>
<td>22.6%</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>82.64</td>
<td>32.9%</td>
<td>8.1%</td>
<td>148.84</td>
<td>79.6%</td>
<td>14.5%</td>
</tr>
</tbody>
</table>

Table 5.6 Results of training SPORE approximation with two and three datasets

<table>
<thead>
<tr>
<th>Training Sets Used</th>
<th>External Test Sets Used</th>
<th>Off-grid Test Sets Used</th>
<th>Horizontal Error (pixels)</th>
<th>Horizontal Error (relative)</th>
<th>Horizontal Error (absolute)</th>
<th>Vertical Error (pixels)</th>
<th>Vertical Error (relative)</th>
<th>Vertical Error (absolute)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2,3</td>
<td>1,2,3</td>
<td>1,2,3</td>
<td>362.11</td>
<td>144.3%</td>
<td>35.4%</td>
<td>251.10</td>
<td>134.3%</td>
<td>24.5%</td>
</tr>
<tr>
<td>1,2,3</td>
<td>1,2,3,4</td>
<td>1,2,3,4</td>
<td>151.90</td>
<td>60.5%</td>
<td>14.8%</td>
<td>172.56</td>
<td>92.3%</td>
<td>16.9%</td>
</tr>
<tr>
<td>1,2,3</td>
<td>1,2,3</td>
<td>1,2,3</td>
<td>147.41</td>
<td>58.7%</td>
<td>14.4%</td>
<td>139.59</td>
<td>74.6%</td>
<td>13.6%</td>
</tr>
<tr>
<td>1,2,3</td>
<td>1,2,3</td>
<td>1,2,3</td>
<td>147.37</td>
<td>58.7%</td>
<td>14.4%</td>
<td>209.22</td>
<td>111.9%</td>
<td>20.4%</td>
</tr>
<tr>
<td>1,2,3</td>
<td>1,2,3</td>
<td>1,2,3</td>
<td>172.53</td>
<td>68.7%</td>
<td>16.8%</td>
<td>202.99</td>
<td>108.6%</td>
<td>19.8%</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>339.51</td>
<td>67.0%</td>
<td>33.2%</td>
<td>231.28</td>
<td>61.0%</td>
<td>22.6%</td>
</tr>
<tr>
<td>1,2,3</td>
<td>1,2,3</td>
<td>None</td>
<td>0.49</td>
<td>0.1%</td>
<td>0.05%</td>
<td>22.54</td>
<td>5.9%</td>
<td>2.2%</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>82.64</td>
<td>32.9%</td>
<td>8.1%</td>
<td>148.85</td>
<td>79.6%</td>
<td>14.5%</td>
</tr>
<tr>
<td>1,2,3</td>
<td>1,2,3</td>
<td>1,2,3</td>
<td>100.01</td>
<td>39.8%</td>
<td>9.8%</td>
<td>115.30</td>
<td>61.7%</td>
<td>11.3%</td>
</tr>
</tbody>
</table>
Table 5.7 Comparison of training SPORE with different grid sizes

<table>
<thead>
<tr>
<th>Training Grid Size</th>
<th>Training Sets Used</th>
<th>External Test Sets Used</th>
<th>Off-grid Test Sets Used</th>
<th>Horizontal error (pixels)</th>
<th>Horizontal error (absolute)</th>
<th>Vertical error (pixels)</th>
<th>Vertical error (absolute)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x3</td>
<td>1</td>
<td>1,2,3</td>
<td>None</td>
<td>339.51</td>
<td>33.2%</td>
<td>231.28</td>
<td>22.6%</td>
</tr>
<tr>
<td>5x5</td>
<td>4</td>
<td>1,2,3,4</td>
<td>None</td>
<td>199.99</td>
<td>19.5%</td>
<td>174.48</td>
<td>17.0%</td>
</tr>
<tr>
<td>7x7</td>
<td>8</td>
<td>1,2,3,8</td>
<td>None</td>
<td>160.75</td>
<td>15.7%</td>
<td>146.58</td>
<td>14.3%</td>
</tr>
<tr>
<td>3x3</td>
<td>1</td>
<td>1</td>
<td>82.64</td>
<td>8.1%</td>
<td>148.85</td>
<td>14.5%</td>
<td></td>
</tr>
<tr>
<td>5x5</td>
<td>4</td>
<td>4</td>
<td>54.07</td>
<td>5.3%</td>
<td>122.58</td>
<td>12.0%</td>
<td></td>
</tr>
<tr>
<td>7x7</td>
<td>8</td>
<td>8</td>
<td>53.37</td>
<td>5.2%</td>
<td>98.51</td>
<td>9.6%</td>
<td></td>
</tr>
<tr>
<td>3x3</td>
<td>1</td>
<td>1,2,3</td>
<td>362.11</td>
<td>35.4%</td>
<td>251.10</td>
<td>24.5%</td>
<td></td>
</tr>
<tr>
<td>5x5</td>
<td>4</td>
<td>1,2,3,4</td>
<td>183.43</td>
<td>17.9%</td>
<td>192.03</td>
<td>18.8%</td>
<td></td>
</tr>
<tr>
<td>7x7</td>
<td>8</td>
<td>1,2,3,8</td>
<td>150.53</td>
<td>14.7%</td>
<td>172.99</td>
<td>16.9%</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.8 Comparison of training SPORE with different orientation angles

<table>
<thead>
<tr>
<th>Training Sets Used</th>
<th>External Test Sets Used</th>
<th>Off-grid Test Sets Used</th>
<th>Horizontal error (pixels)</th>
<th>Horizontal error (absolute)</th>
<th>Vertical error (pixels)</th>
<th>Vertical error (absolute)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4,5,6,7</td>
<td>None</td>
<td>242.89</td>
<td>23.7%</td>
<td>239.71</td>
<td>23.4%</td>
</tr>
<tr>
<td>5,6,7</td>
<td>4,5,6,7</td>
<td>None</td>
<td>97.35</td>
<td>9.5%</td>
<td>112.34</td>
<td>11.0%</td>
</tr>
<tr>
<td>4</td>
<td>4,5,6,7</td>
<td>4,5,6,7</td>
<td>235.45</td>
<td>23.0%</td>
<td>251.92</td>
<td>24.6%</td>
</tr>
<tr>
<td>5,6,7</td>
<td>4,5,6,7</td>
<td>4,5,6,7</td>
<td>119.62</td>
<td>11.7%</td>
<td>152.68</td>
<td>14.9%</td>
</tr>
</tbody>
</table>
Table 5.9 Gaze errors for multiple datasets using 3x3 grid

<table>
<thead>
<tr>
<th>No. of training sets used</th>
<th>Mean error (pixels)</th>
<th>Standard deviation of error (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>495</td>
<td>346</td>
</tr>
<tr>
<td>2</td>
<td>300</td>
<td>238</td>
</tr>
<tr>
<td>3</td>
<td>179</td>
<td>160</td>
</tr>
</tbody>
</table>

Table 5.10 Gaze errors for different grid sizes

<table>
<thead>
<tr>
<th>Training grid size</th>
<th>Mean error (pixels)</th>
<th>Standard deviation of error (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x3</td>
<td>189</td>
<td>143</td>
</tr>
<tr>
<td>5x5</td>
<td>146</td>
<td>151</td>
</tr>
<tr>
<td>7x7</td>
<td>123</td>
<td>121</td>
</tr>
</tbody>
</table>

Table 5.11 Gaze errors for different mirror and frame orientations

<table>
<thead>
<tr>
<th>No. of orientations represented in training data</th>
<th>Mean error (pixels)</th>
<th>Standard deviation of error (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>377</td>
<td>271</td>
</tr>
<tr>
<td>3</td>
<td>167</td>
<td>237</td>
</tr>
</tbody>
</table>

In Table 5.12, each column represents a different method of obtaining the calibration parameters (as described below). In addition to the actual parameter values found in each case, the average value for $J(x)$, the average depth error, and the average horizontal and vertical gaze point errors (see below) over all the samples are given. The second column of Table 5.12 shows that, using ideal calibration parameter values, the two rays were closest at depths that were quite far (about 249mm on average) from the depth estimate found using the two pupils in the WA camera’s image.

Since we could not find a value for $H$ without having accurate calibration parameters, in order to calculate an approximate error for the absolute point of gaze, we first needed to calculate an estimate for $H$. We did this as follows. We calculated $C$ and $E$ for each sample using the approximation for $P$ (i.e., the average of the two values...
found using equations 4 and 5). We then calculated $E_0 = C + \alpha_{\theta_0} \overrightarrow{E}$ such that $E_{0z} = 0$ (i.e., the projection of $\overrightarrow{E}$ onto the x-y plane of our world coordinate system). The projection of $\overrightarrow{E}$ onto the screen is given by equation 6 with $\alpha_{\theta} = \frac{H_s - C_z}{E_z}$. Therefore, the error in each direction is given by:

$$
\varepsilon_x = E_{0x} - E_x = -\frac{E_x}{E_z} H_z
$$

$$
\varepsilon_y = E_{0y} - E_y = -\frac{E_y}{E_z} H_z
$$

We converted the gaze points obtained in screen pixel coordinates to physical distances by assuming a particular screen width and height (1024 by 768 pixels, and 12" by 9"). This allowed us to calculate actual values for $\varepsilon_x$ and $\varepsilon_y$. From these error values, we then estimated $H_z$ for each sample as follows:

$$
\hat{H}_z = \frac{E_z}{2} \left( \frac{\varepsilon_x}{E_x} + \frac{\varepsilon_y}{E_y} \right)
$$

We calculated an average value for $\hat{H}_z$ over all the samples, and used that average value to calculate the projection of each sample onto the screen. The resulting gaze point estimates were then compared to the actual gaze points obtained during the experiment, as shown in Table 5.12.

It is clear from the first column of Table 5.12 that the estimate is not very reliable and that using that estimate, we obtained an average absolute error of 428 pixels (42% of the screen width) horizontally and 232 pixels (30% of the screen height) vertically. This is not surprising since the estimate for $P$ was quite inaccurate.
5.4 Calibration

In order to improve the performance of the direct approach to calculating the point of gaze, we performed various calibration procedures. We first attempted to calibrate both cameras and determine their intrinsic camera parameters. Having realized (as shown in Section 5.4.1) that, for the NA camera, finding the intrinsic parameters would be extremely difficult, we then explored other means of calibrating the system. The following sections describe the results of attempts to calibrate the system.

5.4.1 Camera calibration

First we tried to calibrate the cameras using the procedure described in Section 4.8. The procedure resulted in the following estimations for the WA camera’s intrinsic parameters:

\[ f_w = 8.15 \pm 0.03 \text{mm} \]

\[ m_w = 316.281 \pm 4.654 \]

\[ n_w = 265.382 \pm 5.520 \]

\[ \kappa_{w1} = -0.355155 \pm 0.031485 \]

\[ \kappa_{w2} = -0.548459 \pm 0.455546 \]

\[ \kappa_{w3} = -0.000956962 \pm 0.000721719 \]

\[ \kappa_{w4} = 0.00004482605 \pm 0.000711330 \]

Initial attempts also resulted in the following estimations for the NA camera’s intrinsic parameters:

\[ f_N = 44.60 \pm 1.58 \text{mm} \]

\[ m_N = 109.697 \pm 81.554 \]
\[ n_N = 248.175 \pm 25.111 \]
\[ \kappa_{N1} = 4.7899 \pm 1.0886 \]
\[ \kappa_{N2} = -220.8044 \pm 154.0021 \]
\[ \kappa_{N3} = -0.0083 \pm 0.0157 \]
\[ \kappa_{N4} = -0.1318 \pm 0.0597 \]

In both cases, there was significant uncertainty in the estimates of the distortion parameters, but especially so for the NA camera. For the WA camera, the estimates of the focal centre and principal point are close to what we would expect (the lens has a nominal focal length of 8mm and a perfect lens would have a principal point at coordinates (320,240)). The uncertainties associated with those values are also quite low, making the estimates acceptable. Also, the uncertainty for the first order distortion parameter is also acceptable. On the other hand, for the NA camera, most of the uncertainties are quite high, making the estimates very unreliable. In addition, the focal length is quite a bit lower than what we expected (the lens has a nominal focal length of 50mm).

Notwithstanding, these camera parameters were used as in the previous section to assess whether using them improves the overall performance of the direct gaze calculation. We used the same 160-sample dataset and calculated the errors in the same manner. The third column in Table 5.12 shows that although the actual 3-D pupil location is slightly less accurate than when we used uncalibrated camera parameters, the estimate for \( \hat{H} \) is significantly better (as indicated by a lower \( J(x) \)). This is reflected by a markedly lower gaze error both horizontally and vertically over using uncalibrated camera parameters.
5.4.2 Complete calibration

In order to improve the estimate of the 3-D pupil location further, three approaches using non-linear optimization were explored as described in Section 4.8. We used the same 160-sample dataset and calculated the same performance measures as in the previous section.

The fourth column of Table 5.12 shows the results of the first approach – using the calibrated WA camera intrinsic parameters found above and optimizing the NA camera
intrinsic parameters. The results are roughly the same as with using both sets of camera calibration parameters as in the previous section in terms of gaze point error.

The last column of Table 5.12 likewise shows the results of the second approach — optimizing both the WA and NA camera intrinsic parameters using only gaze point data. The results are not as good as the with the first approach. Indeed, the average errors rise to 291 pixels horizontally and 299 pixels vertically.

Attempts to use the third approach — optimizing both sets of intrinsic parameters as well as the two sets of two camera orientation angles — failed. The non-linear optimization algorithm used did not converge after several thousand iterations, possibly because of limitations of the particular algorithm used. It is likely that the ill-conditioned nature of the underlying system makes it too difficult for common non-linear optimization algorithms to find a solution.

The results of experiments with the GTD system indicate that it is capable of reliably tracking the pupil and glint of the eye of a user in real-time (9 frames per second) in the presence of rapid natural head movements. Specifically, the tracking algorithm positions the two motors correctly so that the left eye of the user always falls within the field of view of the NA camera. It is also able to track the pupil with sub-pixel accuracy, and is robust to changes in eye colour and shape, and lighting conditions (indicated by a high degree of reliability in finding the pupil and glint).

Experiments with using an indirect approach to calculating the point of gaze using the SPORE approximation method revealed three insights. First, making the training data more representative of the possible combinations of inputs improves the accuracy of the
approximation. Second, increasing the size of the grid of points used for training improves the accuracy of the approximation. Third, using data from different orientations of the mirror and frame improves the accuracy of the approximation.

In addition, initial findings indicate that calibrating camera intrinsic parameters lowers the gaze error when using the direct method of calculating the point of gaze.
6 Conclusions and Future Work

Having designed and implemented a novel eye tracker, it is worthwhile to analyze its accuracy, resolution, reliability, robustness and speed. The following section provides such an analysis, while Section 6.2 presents additional improvements and suggests areas of further research.

6.1 Analysis of results

The design of the hardware of the GTD system, including the choice of mechanical, electronic and optical components, are sufficient to maintain adequate tracking of natural head movements and eye fixations. Although the software and hardware currently can process roughly nine frames per second (9fps), theoretically the system can run as fast as 15fps without changing the cameras. The major restrictions on speed are due to a limitation of the speed of the serial interface to electronic components used to drive the motors, and time required to perform the necessary image processing algorithms.

No other complete VOG-based eye tracking system was found that could maintain real-time tracking of the eye in the presence of large, natural head movements. In addition, the sub-pixel accuracy achieved in calculating the location of the pupil and glint in eye images enables the GTD system to provide significantly better accuracy in POG compared to pre-existing, complete VOG-based systems, all of which report only achieving accuracy no better than one pixel.

More specifically, we have observed that, in the presence of natural head movements, during fixations as short as 350ms, a user's pupil and glint can be tracked with more than 90% reliability and 0.1% pixel accuracy (using synthetic data). In fact, with the use of an appropriate spatio-temporal filter, the reliability increases to over 96% on average (see
Section 5.2). This reliability is acceptable since the system is able to recover from errors automatically within a few frames. Here, reliability is defined as the percentage of frames in which the algorithm can calculate the pupil as being somewhere inside the actual pupil and “close” to its centre (see Section 5.2 for more details). Therefore, the results indicate that, for a stationary head or small head motion, the system performs with over 96% reliability and recovers from any loss in tracking within a few frames. It is also expected that with a faster computer and higher camera frame rate, similar reliability would be achieved for large head motion.

Furthermore, the results obtained indicate that the pupil and glint tracking algorithms employed by the GTD system are robust with respect to changes in eye shape and colour, ambient lighting and the presence of eyeglasses. We did find that large head motion between successive frames noticeably affected the accuracy of the algorithms during the head motion. Faster processing (and hence a higher net frame rate) would reduce the difference between successive video frames and thus reduce such the effects of large, rapid head motion on the accuracy of the image processing algorithms.

Based on the pupil and glint locations found in both camera’s images, an indirect approach to approximating the point of gaze on a computer monitor using the SPORE approximation method yielded gaze errors of 53 pixels (or 5.2% of the monitor’s width) horizontally and 99 pixels (or 9.6% of the monitor’s height) vertically when trained and tested with a single dataset (see Table 5.7), and 120 pixels (or 11.7%) horizontally and 153 pixels (or 14.9%) vertically when tested with data not represented by the training data (see Table 5.8). Note that these errors are significantly higher than the expected errors of 13 pixels horizontally and 8 pixels vertically, as calculated in Appendix E.
With adequate representation of the input space in the data used to train it, the SPORE approximation interpolated data to within the grid spacing used for training. That is, there is clear evidence that using a higher resolution grid of points on the monitor during training and using training samples from data obtained with the mirror and frame at different orientation angles would both increase the accuracy of the approximation.

A direct approach to calculating the point of gaze using ideal camera parameters initially yielded poor results, with errors of 41.8% of the monitor’s width horizontally and 35.6% of its height vertically. Attempts to calibrate the intrinsic camera parameters were met with limited success. However, combined with various non-linear optimization methods, the system’s parameters were calibrated enough to reduce the errors to 22.0% horizontally and 27.8% vertically. Again, these errors are significantly higher than the expected errors of 13 pixels horizontally and 8 pixels vertically.

Clearly, the indirect method performed better, primarily because the ill-conditioned nature of the calibration problem makes it very difficult to solve optimally. However, the notable effect of even inaccurate calibration indicates that further investigation into accurately calibrating the camera parameters would likely significantly improve the accuracy of the direct method. This is desirable because the direct method provides information about the angle of gaze as well as the point of gaze. In addition, once an effective means of calibration has been identified, calibrating the direct method (which needs to be done every time the monitor or eye tracker is moved) will be much easier than calibrating the indirect method.
6.2 Future work and applications

The GTD system presented in this thesis performs well as a device for tracking features of a user’s eye. It is useful for some basic applications that require tracking of a user’s eye, and provides a good foundation for future research into tracking a user’s point of gaze and angle of gaze. Several areas of possible research follow.

The readings obtained from the potentiometers used to determine the orientation of the frame and mirror are prone to errors due to noise. Investigating further the sources of noise and modifying the electronics to better shield against them would no doubt improve the accuracy of the measurements. In addition, since it has been determined that a CRT-based computer monitor is one of the sources of noise affecting the electronics, it would be worthwhile to try using a liquid crystal display (LCD) screen instead.

While the performance of the image analysis algorithms used to detect and parameterize the pupil and glint is very good, there are two main components that would benefit from further research: the reliability of the detection algorithms in the presence of motion artefacts and the accuracy of the pupil centre estimation.

Rather than the simple spatio-temporal filter currently used, a more sophisticated Kalman filter could be used to compensate for blinks and motion artefacts caused by large, rapid head movements.

The system is already able to provide sub-pixel accuracy when estimating the centre of the pupil and glint. However, better contour tracing and ellipse-fitting algorithms may provide even better estimates and hence improved overall visual accuracy, especially using the direct approach to gaze calculation. Talmi and Liu [8] used a different approach (including the use of the Hough transform) to estimating the area and perimeter
of an ellipse from a binary image. Zhu et al. [25] also developed a robust curvature algorithm for pupil centre detection. Since the GTD system also characterizes both the pupil and glint as ellipses, investigating such approaches may improve the overall accuracy of the gaze calculation. In addition, further investigation into other possibly more robust and efficient statistical means of estimating ellipse parameters – such as those presented by Gander et al. [62], Yang et al. [63], Bennet and Burridge [64] or Halir and Flusser [65] – may well prove useful in improving the overall accuracy of the system.

The area of research that would probably be of greatest benefit to improving the performance of the GTD system is the actual gaze calculation based on the pupil and glint parameters extracted from the video images. For the indirect approach, using more extensive training of the SPORE approximation, especially with different orientations of the mirror and frame, would help verify its usefulness and ultimately improve its performance.

As for the direct approach to gaze calculation, the main issue is investigating better calibration procedures. In particular, an effective means of accurately determining the intrinsic parameters (focal length, principal point and radial distortion parameters) of both cameras is necessary. One such means may involve the construction of a “synthetic eye” and a controlled calibration environment where the exact 3-D location of the eye can be measured. In addition, using multiple light sources (see [45]) may well improve the robustness of the overall gaze estimation using direct calculation.

Aside from accuracy, robustness and reliability, two further steps could be taken to improve the overall speed of the system. First, the DragonFly cameras could be operated in continuous mode rather than with an external trigger. These cameras provide an
electronic signal that could be used to synchronize the LED circuit directly, rather than using the FT649 and software. This would eliminate most of the serial communication that currently accounts for 23ms for every video frame. Further, it would increase the hard limit of the processing speed of the system from 15fps to 30fps. Second, investigating ways to increase the communication bandwidth with the electronics (which is currently limited to 2400bps) would be of additional benefit to improving the speed of the system.

While this thesis presented numerical measures of the accuracy, robustness, reliability and speed of the image processing and gaze calculation software, only subjective measures of the performance of the actual tracking was reported. In the case of reliability, even the quantitative assessment was based on synthetic data. By using marked contact lenses, it may be possible to determine the actual position of the pupil, and measure the reliability of the system using real data as we! Also, investigating ways to measure the ability of the GTD system to initially find and then track the user’s left pupil for different users and different head positions and orientations would help identify further the strengths of the system as well as areas that could benefit from further research. In addition, the robustness of the GTD system to ambient infrared lighting could be further quantified by performing experiments where remote control devices, laser pointers and other devices that emit infrared light are present and turned on.

Finally, it was beyond the scope of this thesis to explore human factor issues such as ease of use (including ease of calibration), potential integration with other input devices, and specific applications that would benefit from this technology. Therefore, exploration of these issues would no doubt uncover further areas of improvement.
Overall, this thesis explored various VOG-based eye tracking devices, presented a novel design for such a tracker for use in a human computer interface and reported on the performance of a particular implementation of that design. The resulting GTD system provides accurate, robust and fast tracking and parameterization of a user's pupil and corneal reflection of a known light source. An indirect approach to estimating the user's point of gaze using the SPORE approximation was explored. While a more desirable, direct approach did not produce results that were as accurate, it is expected that further investigation into effective calibration procedures would make it an equally (if not more) effective alternative to estimating the user's point of gaze.
References


Appendix A – Electronic Circuits

The following provides the schematics and brief descriptions of the two electronic circuits used in the GTD system.

Infrared LED circuit

For each infrared LED circuit (see Figure A.1), we used two sets of eight (8) HSDL-4220 AlGaAs infrared LEDs from Agilent Technologies that emit light around the 875 nm wavelength range at a viewing angle of 30°. Each LED provides up to 190 mW/sr of on-axis radiant intensity, and each set of eight is arranged in a concentric ring. Other infrared LEDs were tested and these ones seemed to produce very good illumination.

Connector JP1 provides an 18VDC power signal to the LEDs, which are arranged in series with a resistor. The values of the two resistors were chosen to minimize the difference between the total ambient illumination within the field of view of the cameras when each ring of LEDs is switched on. Since the inner ring of LEDs is closer to the optical axis of the camera, more of the light emitted from those LEDs would be sensed by the camera than from the LEDs on the outer ring. Therefore, by choosing a higher resistor value for R2, we pass less current through the inner ring of LEDs, thus compensating for the difference in intensities. The reason an 18V power supply is needed is so that sufficient current is available to the LEDs to produce the required intensity.

The power from the supply is also fed into a voltage regulator U1, which in turn powers two NAND gates used as logic inverters. The output of each gate provides a logic signal that switches each of U3 and U4 between saturation and cut-off. U3 and U4
are power MOSFETs that essentially act as switches, determining whether current is drawn through each ring of LEDs or not.

The input to the NAND gates are provided by connector JP2. Diodes D17 and D18 are simple protection diodes. We connected JP2 of each of the two LED circuits to a digital I/O signal provided by the control and interface circuit described below. Ultimately, the control signal comes from the PC via the serial port.

**Control and interface circuit**

The control and interface circuit shown in Figure A.2 uses two ICs available from Ferrettronics. The Ferrettronics FT649 router IC provides 5 digital input/output (I/O) lines and is controlled via a 2400 baud serial line. It can be connected directly to a PC’s standard RS232 line using a diode and a couple of resistors (see Figure A.2). Two (2) of the I/O lines are used to send control signals to the motors (see below). Two of the lines are used to provide the trigger signal for each of the two cameras. The remaining line is used to provide a single control signal for both LED circuits.

In order to control each motor (one for rotating the mirror and the other for rotating the frame), our circuit uses off-the-shelf ICs. First, through the PC serial port, commands are sent to the FT649 chip indicating which motor is to be controlled. This enables one of the two I/O lines on the FT649 used to control the motors. The output of that I/O pin is connected to the input of a Ferrettronics FT609 chip, which is a stepper motor logic controller. Like the FT649, it is controlled via commands sent over a 2400 baud serial line. The FT609 has, in a single IC package, the following features that made it an ideal choice for our design:

- It works with simple commands over a serial port.
• It can provide signals at various frequencies, allowing the operation of the motor at different speeds.
• It can operate the motor in either direction.
• It can single step the motor or step it a given number of steps (i.e., it has a built-in count function).
• It has a home pin that can be used to manually stop the motor even if it is in the middle of stepping a certain number of steps.
• It provides automatic ramp up and ramp down functions to accommodate large loads or high speeds.

Commands indicating the direction, speed and number of steps to step the motor are passed from the PC serial port, via the FT649 chip to the FT609. The logic signals from the FT609 are then passed through an opto-isolator to an L293D half-H driver. The FT609 does not have the high current capacity to directly drive the stepper motor; therefore the L293D is used to actually provide the drive signals for the motor. The opto-isolator is used to ensure that the digital components (including the PC) used to provide the control signals are protected. Finally, the drive signals from the L293D are connected to the motor through a network of protection diodes.

In order to provide a “home” position for each of the motors, and to provide a mechanism for checking that the motors have not slipped, potentiometers are connected to the shaft of each motor, and their varying resistance is measured as an indication of the angle of rotation of the shaft. Specifically, the varying resistance is converted to a varying voltage. The voltage is quantified and measured digitally using a small analogue to digital (A/D) converter. As shown in Figure 4.6, the voltage from each potentiometer
is measured by the A/D converter (Maxim MAX187 A/D converter ICs are used in our system). A single serial line converter chip (a Maxim MAX2201C is used in our system) converts the serial signals of the A/D to levels compatible with a standard PC RS232 interface. We initially found that ambient electromagnetic noise was interfering with accurate and consistent output from the A/D circuit. We took the following steps to reduce such effects:

1. We reduced the length of wires used to connect the circuit to the motors, LED circuits and potentiometers.
2. We used shielded cables to connect the circuit to the potentiometers.
3. We took several samples for each reading and averaged the results.

The two biggest sources of interference seemed to be the operation of the motors and radiation emitted by the computer monitor's CRT (which necessarily is positioned close to the circuit and potentiometers). By using shielded cables, the effect of the motors (which are generally not moving while we take readings) was nearly eliminated. The shielding also reduced the effects of the radiation caused by the monitor. Together with the averaging of several samples, the monitor signal's interference was also reduced to acceptable levels. The interference from the monitor causes a sinusoidal error signal, which is why averaging several samples works well.

Thus, the control of the LED rings, the triggering of the cameras, the operation of both motors and the measurement of the angle of rotation of the mirror and frame are all handled via simple commands sent through the PC serial port to a single control and interface circuit.
Figure A.1  Circuit diagram for infrared LED circuit
Figure A.2 Circuit diagram for control and interface electronics
Appendix B – System Diagrams

The following are diagrams illustrating the GTD system.
Figure B.1 Front view of GTD eye tracker
Figure B.2 Side view of GTD eye tracker
Appendix C – Motor Requirements

The following describes the requirements of the motors used in the GTD system.

Angular speed

We first calculated the speed at which the motors would be required to turn. Figure C.1 shows a simplified diagram that can be used to estimate the angular speed required. Suppose initially the centre of the NA camera’s image is pointing to the centre of the user’s pupil. Now suppose that, in a single time unit, the head rotates about its axis $\theta_H$ radians, and the eye does not rotate in its socket. Then the linear distance that the centre of the pupil has moved is $l_H < r_H \theta_H$.

In the worst case, this projects to a distance on the NA camera’s image plane of $l_m < l_H$, where $l_m \approx r_m \theta_m$ for large $r_m$. Therefore, the rotation angle required is given by:

$$\theta_m \approx \frac{l_m}{r_m} < \frac{l_H}{r_m} \frac{r_H \theta_H}{r_m}$$

(37)

Assuming that the user’s head is at most 10cm in radius ($r_H=10cm$), that the user is at least 30cm from the centre of rotation ($r_m=30cm$), and that the head rotates at most 100° per second ($\theta_H=100°$), then the maximum angle the motor is required to rotate in a single second is $\frac{100°/100°}{300mm}$. Therefore, it was determined that the angular speed of the motors must be at least $\omega_o \approx 33.3°/s = 0.5818 rad/s$. This holds true for both the mirror
and frame motors. Next, we calculated the required output torque of each motor, as described below.

Figure C.1 Rotation speed requirements
Mirror motor

The mirror motor rotates a rectangular 50g, 70mm x 90mm x 2mm mirror about its axis. The inertial load of the mirror can be calculated as:

$$I_{\text{mirror}} = \frac{M_c}{12} (L_c^2 + H_c^2)$$  \hspace{1cm} (38)

where $I_{\text{mirror}}$ is the inertia,

$M_c$ is the mass of the mirror, 

$L_c$ is the length of the mirror, and 

$H_c$ is the height of the mirror.

We wanted to be able accelerate the motor from 0 to $\omega_0$ in half the time it takes the cameras to capture a frame. The output torque required from the motor and any associated gears to rotate the mirror was therefore found to be:

$$\tau_m = 2F\omega_0 I_{\text{mirror}}$$ \hspace{1cm} (39)

where $\tau_m$ is the torque (in N·m), 

$\omega_0$ is maximum angular speed (in rad/s) required and 

$F$ is frame rate of the camera (in Hz).

This ensures that at the maximum angular speed of the object being tracked (in this case the head), the motor can move the mirror the required angular distance within half a camera frame. Given an angular speed of $\omega_0 = 33.3^\circ/s$ and a frame rate of $F = 30$ Hz, and since we are rotating about the y-axis, we found the minimum output torque required to be $\tau_m \approx 1.2 \times 10^{-3}$ Nm.
The frame motor rotates the following pieces that contribute to the net inertial load on the motor that rotates them about the x-axis:

1. Mirror motor, shaft, gears and clamp used to attach motor assembly to frame
2. Potentiometer for measuring angle of rotation of mirror
3. Mirror and shaft for rotating mirror
4. LED array circuit and bars used to attach circuit to frame
5. NA camera lens
6. NA camera circuit board, CCD sensor and bars used to attach camera to frame
7. Hollow rectangular metal frame

If we neglect inertial load caused by the potentiometer (since it is relatively small and light), friction, wires attached to the circuits being, as well as the shafts and bars used to attach items to the rotating frame rotated (items 4 and 7), we can calculate the inertia for each item as follows (presented in the same order as the list above):

1. We approximate the entire mirror motor, gear and shaft assembly as a big rectangular block, and put it on the axis of rotation (about the x-axis). The actual inertial load should not be too much larger than this value. Therefore,

\[ I_a = \frac{M_a}{12} (L_a^2 + H_a^2) \] (40)

where \( I_a \) is the inertial load of the mirror motor
\( M_a \) is the mass of the equivalent rectangular block
\( L_a \) is the length of the equivalent rectangular block
\( H_a \) is the height of the equivalent rectangular block.
For our system, we use the values \( M_a = 38g \), \( L_a = 2.9cm \) and \( H_a = 5.8cm \), so \( I_a = 133g \cdot cm^2 \).

2. We ignore the potentiometer, as it is very light and small.

3. We approximate the mirror as a thin rectangular block, and calculate its maximum inertia (i.e., when it is parallel to the NA camera image plane), ignoring the inertial load of the shaft around which it rotates. This leads to:

\[
I_c = \frac{M_c}{12} (L_c^2 + H_c^2) \tag{41}
\]

where \( I_c \) is the inertial load of the mirror

\( M_c \) is the mass of the mirror

\( L_c \) is the length of the mirror

\( H_c \) is the height of the mirror.

For our system, we use the approximate values \( M_c = 50g \), \( L_c = 9cm \) and \( H_c = 7cm \), so \( I_c = 542g \cdot cm^2 \).

4. We approximate the LED array circuit and the bars used to attach it to the frame as a thin rectangular block, and calculate its inertia. This leads to:

\[
I_d = \frac{M_d}{12} (L_d^2 + H_d^2) \tag{42}
\]

where \( I_d \) is the inertial load of the equivalent rectangular block

\( M_d \) is the mass of the equivalent rectangular block

\( L_d \) is the length of the equivalent rectangular block

\( H_d \) is the height of the equivalent rectangular block.
For our system, we use the approximate values $M_d = 100g$, $L_d = 8cm$ and $H_d = 8cm$, so $I_d = 1067g \cdot cm^2$.

5. We approximate the NA camera lens as a solid cylinder, which has an inertial load of:

$$I_e = \frac{M_e R_e^2}{2} \quad (43)$$

where $I_e$ is the inertial load of the lens

$M_e$ is the mass of the lens

$R_e$ is the radius of the lens cylinder.

For our system, we use the values $M_e = 90g$ and $R_e = 2cm$, so $I_e = 180g \cdot cm^2$.

6. We approximate the NA camera circuit board, CCD sensor and the bars used to attach them to the frame as a thin rectangular block, and calculate its inertia. This leads to:

$$I_f = \frac{M_f}{12} (L_f^2 + H_f^2) \quad (44)$$

where $I_f$ is the inertial load of the equivalent rectangular block

$M_f$ is the mass of the equivalent rectangular block

$L_f$ is the length of the equivalent rectangular block

$H_f$ is the height of the equivalent rectangular block.

For our system, we use the approximate values $M_f = 250g$, $L_f = 5cm$ and $H_f = 6cm$, so $I_f = 1271g \cdot cm^2$. 
7. We calculate the inertia of the metal frame as the difference between two solid rectangular blocks. For the outer block, we get:

\[ I_{g1} = \frac{M_{g1}}{12} \left( L_{g1}^2 + H_{g1}^2 \right) \]  

(45)

where \( I_{g1} \) is the inertial load of the outer rectangular block

\( M_{g1} \) is the mass of the outer rectangular block

\( L_{g1} \) is the length of the outer rectangular block

\( H_{g1} \) is the height of the outer rectangular block.

For the inner rectangular block, we get:

\[ I_{g2} = \frac{M_{g2}}{12} \left( L_{g2}^2 + H_{g2}^2 \right) \]  

(46)

where \( I_{g2} \) is the inertial load of the inner rectangular block

\( M_{g2} \) is the mass of the inner rectangular block

\( L_{g2} \) is the length of the inner rectangular block

\( H_{g2} \) is the height of the inner rectangular block.

Assuming uniform mass, we can also express \( M_{g2} \) as a function of \( M_{g1} \):

\[ M_{g2} = \frac{L_{g2} H_{g2}}{L_{g1} H_{g1}} M_{g1} \]  

(47)

Also, if we know the actual mass \( M_g \) of the hollow frame, we can express the mass of the outer frame as:

\[ M_{g1} = \frac{M_g}{1 + \frac{L_{g2} H_{g2}}{L_{g1} H_{g1}}} \]  

(48)
Therefore, combining equations 45, 46, 47 and 48, we get:

\[
I_g = \frac{M_g L_{g1} H_{g1}}{12 L_{g1} H_{g1} + 12 L_{g2} H_{g2}} \left( L_{g1}^2 + H_{g1}^2 - \frac{L_{g2} H_{g2}^3}{L_{g1} H_{g1}} - \frac{L_{g1} H_{g2}^3}{L_{g1} H_{g1}} \right)
\]

where \( I_g \) is the inertial load of the hollow rectangular frame.

For our system, we use the approximate values \( M_g = 50\, g \), \( L_{g1} = 30\, cm \)
\( H_{g1} = 11\, cm \), \( L_{g2} = 28\, cm \) and \( H_{g2} = 9\, cm \), so \( I_g = 852\, g \cdot cm^2 \).

The net inertial load on the frame motor can then be expressed as:

\[
I_{frame} = I_o + I_c + I_d + I_e + I_f + I_g = (133 + 542 + 1067 + 180 + 1271 + 852) = 4044 g \cdot cm^2
\]

Thus, the output torque required from the motor and any associated gears is given as:

\[
\tau_{frame} = 2 F \omega_0 I_{frame}
\]

Given an angular speed of \( \omega_0 = 33.3^\circ/s \) and a frame rate of \( F = 30\, fps \), we found the minimum output torque required to be \( \tau_{frame} \approx 14.1 \times 10^{-3}\, Nm \).

**Resolution**

Next we estimated the angular resolution required of both motors. We used the following methodology.

We start by formulating the field of view of the WA camera:

\[
D = \frac{d \cdot z}{f}
\]

where \( D \), \( d \), \( f \) and \( z \) are all shown in Figure 3.5. We start with the horizontal field of view.
Now, as a simplifying assumption, let us place the centre of rotation \( M \) of the NA camera at the origin (i.e. the centre of the WA camera’s image sensor) in Figure 1.5. In this case, the average distance represented by a single pixel in the WA camera image is

\[
I = \frac{D}{N}
\]

(53)

where \( I \) is the horizontal distance represented by each pixel,

\( D \) is as above, and

\( N \) is the width of the WA camera image in pixels.

From Figure C.2, for large \( z \), we can approximate the angle \( \theta \) that the motor must be turned by:

\[
\theta \approx \frac{1}{z} = \frac{D}{zN} = \frac{(d \cdot z / f)}{zN} = \frac{d}{fN}
\]

(54)

where \( \theta \) is in radians.

In our system, we know that, in the horizontal direction, \( d = 4.8 \text{mm} \), \( f \approx 8 \text{mm} \) and \( N = 640 \). Therefore, the motors must be able to turn in increments of at least

\[
\theta = \frac{4.8}{(8)(640)} = 0.0009375 \text{ radians or } 0.054^\circ.
\]
For the horizontal direction, we have $d = 3.6\, \text{mm}$, $f \approx 8\, \text{mm}$ and $N = 480$, so the angular resolution is again $0.054^\circ$. Therefore, for both the frame and mirror motors, we determined that we require an angular resolution of $0.054^\circ$.

**Required torque**

Given the torque $\tau$ that must be available to accelerate the external load (i.e., the mirror or the frame and everything attached to it), we calculated the minimum output torque of the motor/gearhead combination as follows:

$$\tau_{\text{min}} = (1 + \varepsilon) \frac{\tau / N_g^2 + 2 F \omega_0 I_{\text{rotor}}}{e}$$

(55)

where $\tau_{\text{min}}$ is the minimum torque required from the motors,

$\varepsilon$ is a safety margin,

$N_g$ is the gear ratio (where $N_g > 1$),

$I_{\text{rotor}}$ is the rotor inertia of the motor,

$e$ is the efficiency of the gearhead, and

$F$ and $\omega_0$ are as above.

For both motors, we chose the AM1524 motor from MicroMo Electronics Inc. This is a bi-polar voltage mode stepper motor with 24 steps per revolution. It weighs 0.42 oz and has a rotor inertia of $6.4 \times 10^{-6} \, \text{oz-in-sec}^2$.

For the mirror motor, we combined it with a MicroMo 15/8 (262:1 gear ratio) zero backlash gearhead. This gearhead has an efficiency of 43%. With a safety margin of 50%, we get $\tau_{\text{min}} = 5.6 \mu\text{Nm}$. According to the specifications provided by MicroMo, the AM1524 can output a torque of approximately $0.03 \, \text{oz-in} = 212 \mu\text{Nm}$ at $33.3^\circ/\text{sec}$ or 582
steps/sec. Therefore, at a speed of rotation of 582 steps/sec, this motor/gearhead combination can output enough torque to rotate the mirror. Also, this combination provides an effective angular resolution of \( \frac{360}{(24)(262)} = 0.057^\circ \), which is sufficiently close to our requirements as reported above.

For the frame motor, we combined the motor with a MicroMo 16/7 (415:1 gear ratio) planetary gearhead. This gearhead has an efficiency of 55%. With a safety margin of 50%, we get \( \tau_{\text{min}} = 4.5\mu\text{Nm} \). Again, since the AM1524 can output a torque of approximately 0.03 oz-in \( \approx 212\mu\text{Nm} \) at 33.3°/sec or 582 steps/sec, this motor/gearhead combination can output enough torque to rotate the frame and all the components attached to it. Also, this combination provides an effective angular resolution of \( \frac{369}{(24)(415)} = 0.036^\circ \), which exceeds our requirements.
Appendix D – Camera Specifications And Details

The DragonFly camera (see Figure D.1) available from Point Grey Research, Inc. in Vancouver, B.C., Canada (http://www.ptgrey.com) was used in the GTD system. The following describes some of the specifications and operational details of the DragonFly camera. All the figures were taken from the user’s manual that came with the camera.

An IEEE 1394 interface is built into the board-level camera, along with a CCD sensor (an ICX084AL from Sony). It can capture images continuously at rates up to 30 fps, and external triggering is available at up to 15 fps. The sensor is a standard 1/3” CCD sensor with a 4.8mm width and 3.6mm height.

![Figure D.1 Photograph of DragonFly camera](image)

The triggering is provided by a 6-pin header underneath the board (i.e. on the opposite side of the CCD sensor – see Figure D.2). The header gives access to +3.3V and
a ground signal. It also has three (3) general purpose digital input output (GPIO) pins (see Figure D.3), any one of which can be configured via the software API provided with the camera to accept the external triggering signal. One of the pins (pin IO2) comes equipped with a pull-up resistor, so that is the pin that has been chosen in the system implemented for this thesis as the external trigger pin.

The cameras used in the GTD system were purchased for $799.00US each, not including sales applicable sales taxes.
Figure D.3 Layout of GPIO pins on DragonFly cameras
Appendix E – System Geometry

The theory behind the operation of the GTD system requires extensive use of advanced geometry. The following sections describe the basic theory behind the relationship between image resolution and accuracy of point of gaze and how the eye is tracked in the presence of large, natural head movements. In addition, the calculation of the 3-D position of the pupil and the ultimate determination of the user’s point of gaze are presented in the following sections. Finally, a brief description of the mathematical approach to fitting ellipses to the pupil and glint in an image is presented in this appendix.

Resolution

The pixel resolution with which eye features (such as the pupil) are measured in an image of the eye obtained by an eye tracker poses an upper limit on the accuracy with which the tracker can calculate the user’s point of gaze on an external surface such as a computer monitor. That is, there is a direct relationship between the resolution and accuracy of an eye tracker. This relationship is illustrated in Figure E.1. Specifically, a relationship between the accuracy in the measurement of an eye feature (such as a pupil) and the best accuracy that can be obtained in point of gaze can be expressed as follows.

For large $z_1, z_2$:

$$r\theta \approx \frac{z_1(\hat{u}-u)}{f}$$  \hspace{1cm} (56)

$$z_2\theta \approx x$$  \hspace{1cm} (57)

$$\therefore x \approx \frac{z_1z_2(\hat{u}-u)}{rf}$$  \hspace{1cm} (58)
Assuming the eye feature can be measured to within one pixel, the image is 640 pixels wide and 480 pixels high, the image sensor is 4.8mm wide and 3.6mm high, the lens focal length is 50mm, $r = 7.7mm$ and $z_1 = z_2 = 500mm$, equation 58 implies an accuracy of approximately 4.9mm in both height and width. For a monitor that is 300mm wide and 230mm high (e.g., a typical 15” monitor), and a screen resolution of 1024x768 pixels, this means that the POG can be calculated at best to within 17 pixels horizontally and 16 pixels vertically.

Table E.1 shows the relationship between the image resolution and accuracy in determining the pupil in an image, and the best accuracy in POG that can be obtained.
Table E.1 Resolution and accuracy

<table>
<thead>
<tr>
<th>Image resolution</th>
<th>Horizontal pixel accuracy in image</th>
<th>Vertical pixel accuracy in image</th>
<th>Horizontal POG accuracy (mm)</th>
<th>Vertical POG accuracy (mm)</th>
<th>Horizontal POG accuracy (pixels)</th>
<th>Vertical POG accuracy (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>640x480</td>
<td>1</td>
<td>1</td>
<td>4.9</td>
<td>4.9</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>320x240</td>
<td>1</td>
<td>0.492</td>
<td>9.7</td>
<td>9.7</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>640x480</td>
<td>0.758</td>
<td>0.492</td>
<td>3.7</td>
<td>2.4</td>
<td>13</td>
<td>8</td>
</tr>
</tbody>
</table>

We assume here the use of a 640x480 image, 4.8mm x 3.6mm image sensor, a focal length of 50mm, \( r = 7.7\text{mm} \), \( z_1 = z_2 = 500\text{mm} \), a 300mm x 230mm monitor and a screen resolution of 1024x768 pixels. The first row indicates the accuracy that can be obtained by a system that uses a 640x480 image and pixel level accuracy. The second row indicates the same system but using half the image resolution. This is the case for several existing eye trackers, such as the one described in [52]. Note that we are assuming a focal length of 50mm, but a shorter focal length which allows for a wider field of view would result in even poorer POG accuracy. The third row indicates the case of the GTD eye tracker presented in this thesis, which achieves sub-pixel accuracy. Clearly, the ability of the GTD system to measure the location of the pupil and glint accurately to within less than a pixel in an image allows it to achieve a much higher degree of accuracy in POG than any other complete VOG-based eye tracking device that allows for large head movements.

NA camera orientation

In order to successfully track the eye so that a good image of the pupil can be obtained, the system must first provide a means of rotating the NA camera so that the user's eye falls within its field of view. The pixel representing the centre of the pupil as
measured in the WA camera represents the ray of light reflected off the actual pupil. Suppose we want to orient the NA camera so that the ray of light reflected off the same pupil, off the surface of the mirror, and through the lens to the camera sensor of the NA camera corresponds to the centre of the image. For any given depth, there is a unique set of two angles (horizontal and vertical) that orients the NA camera in this way.

These angles can be calculated as follows (see Figure E. 2). First, we assume the point $P$ is calculated using the pixel coordinates of the pupil centre in the WA camera's image and the depth estimate calculated using the separation between the two pupils. We can trace the ray of light reflecting off the pupil by first defining two vectors:

$$ \hat{v}_r = \frac{M - P}{\|M - P\|} $$  \hspace{1cm} (59)
\[ \hat{v}_r = \frac{\mathbf{N} - \mathbf{M}}{\|\mathbf{N} - \mathbf{M}\|} \]  

(60)

where:

- \( \mathbf{M} \) is the centre of rotation of the mirror (see Figure E.2)
- \( \mathbf{N} \) is the centre of NA camera’s image sensor (see Figure E.2)

These represent the incident (\( \hat{v}_r \)) and reflected (\( \hat{v}_r \)) rays from the surface of the mirror. The points \( \mathbf{N} \) and \( \mathbf{M} \) are known from the geometry of the system, and hence the values for \( \hat{v}_r \) and \( \hat{v}_r \) can be calculated. Assuming perfect reflection, the normal \( \hat{n} \) to the plane of the mirror can be calculated using the formula:

\[ \hat{v}_r = \hat{v}_r - 2 \cdot (\hat{n} \cdot \hat{v}_r) \cdot \hat{n} \]  

(61)

Note that equation 61 can also be expressed as:

\[ \hat{v}_r = \hat{v}_r - 2 \cdot (\hat{n} \cdot \hat{v}_r) \cdot \hat{n} \]  

(62)

Generally, this is a fairly complex non-linear system of three (3) equations in three (3) unknowns. However, if we design the apparatus so that \( v_y = v_z = 0 \) where \( \hat{v}_r = [v_x \quad v_y \quad v_z]^T \), then equation 62 is simplified to:

\[ v_x - 2n_x^2v_x = x \]  

(63)

\[ -2n_xn_yv_x = y \]  

(64)

\[ -2n_xn_zv_x = z \]  

(65)

where:

\[ \hat{n} = [n_x \quad n_y \quad n_z] \]

\[ \hat{v}_r = [x \quad y \quad z] \]

Equations 63, 64 and 65 represents a system of equations with two solutions. Physically, the one that makes sense for calculating the angles of rotation is the one with
\( n_z > 0 \) since the mirror would never face away from the user. We can therefore solve for \( \hat{n} \) directly and efficiently without resorting to non-linear approximation methods.

Once the normal \( \hat{n} \) is known, then the angles needed to rotate the mirror can be calculated by using the following formulae:

\[
\rho_H = \cos^{-1}(n_y) - \pi / 2
\]

\[
\rho_V = \cos^{-1}(n_x) - \pi / 2
\]

(66)

(67)

where \( \rho_H \) and \( \rho_V \) represent the horizontal and vertical angles required to rotate the mirror, respectively. Note that if there is an error in the depth of the calculated position of the pupil (\( P \)), the actual pixel location of the pupil in the NA camera’s image will not be in the centre of the image. At start-up, if the user is looking straight into the screen (i.e., the head is not tilted about the y-axis), the depth estimate should be close enough to ensure the pupil still falls within the field of view of the NA camera. Once the pupil has been found by the NA camera, other measures can be used to calculate a more accurate and robust estimate of the depth of the pupil, and hence decrease the chance that the tracking algorithm would fail to place the pupil in the field of view of the NA camera.

**Calculation of 3-D pupil position**

Equations 4 and 5 present an over-determined system of 6 equations with 5 unknowns (\( P, \alpha_s \) and \( \alpha_o \)). We can compute the other four vectors in equations 4 and 5 as follows.

We can calculate the 3-D position of the focal point of the WA camera using:

\[
W_H = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}
\]

(68)
\[ W = R_w (W_w - T_w) \]  \hspace{1cm} (69)

where:
- \( W_w \) is the focal point in the WA camera’s coordinate system
- \( R_w \) is the rotation matrix used to transform points in the WA camera’s coordinate system to the world coordinate system
- \( T_w \) is the translation vector used to transform points in the WA camera’s coordinate system to the world coordinate system

Now we can compute the direction of the vector from the image of the pupil centre in the WA camera’s image plane to the pupil centre itself:

\[ \vec{Q}_w = \begin{bmatrix} u_w / f_w \\ v_w / f_w \\ 1 \end{bmatrix} \]  \hspace{1cm} (70)

\[ \vec{Q} = R_w (\vec{Q}_w - T_w) \]  \hspace{1cm} (71)

where:
- \( u_w \) is the x-coordinate of the pupil centre image in the WA camera’s image plane
- \( v_w \) is the y-coordinate of the pupil centre image in the WA camera’s image plane
- \( f_w \) is the focal length of the WA camera
- \( \vec{Q}_w \) is the vector \( \vec{Q} \) in the WA camera’s coordinate system

Next we trace backward along the light ray resulting in the image of the pupil centre in the NA camera’s image plane:

\[ F_w = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \]  \hspace{1cm} (72)
\[ A_N = \begin{bmatrix} u_N / f_N \\ v_N / f_N \\ 1 \end{bmatrix} \]  
(73)

\[ A_M = R_{NM} (A_N - T_{NM}) \]  
(74)

\[ F_M = R_{NM} (F_N - T_{NM}) \]  
(75)

\[ \overline{A}_M = A_M - F_M \]  
(76)

\[ S_M = F_M + \alpha \overline{A}_M \]  
(77)

\[ S_{M'} = 0 \]  
(78)

\[ V_A = \frac{\overline{A}_M}{||\overline{A}_M||} \]  
(79)

\[ \overline{S}_M = \begin{bmatrix} V_{Ax} \\ V_{Ay} \\ -V_{Az} \end{bmatrix} \]  
(80)

\[ S = R_{MW} (S_M - T_{MW}) \]  
(81)

\[ \overline{S} = R_{MW} (\overline{S}_M - T_{MW}) \]  
(82)

where:

- \( f_N \) is the focal length of the NA camera
- \( u_N \) is the x-coordinate of the pupil centre image in the NA camera's image plane
- \( v_N \) is the y-coordinate of the pupil centre image in the NA camera's image plane
- \( F_N \) is the focal point in the NA camera's coordinate system
- \( F_M \) is the focal point in the mirror's coordinate system
\( \bar{A}_N \) is the intersection of the light ray from the pupil with the NA camera plane in the NA camera’s coordinate system

\( \bar{A}_M \) is the vector \( \bar{A}_N \) in the mirror’s coordinate system

\( S_M \) is the intersection of \( \bar{A}_M \) with the mirror plane in the mirror’s coordinate system

\( \alpha_A \) is a scale factor used to solve for \( S_M \) using equations 77 and 78

\( \bar{V}_A \) is the unit vector representing the incident light ray from the pupil off the surface of the mirror (in the mirror’s coordinate system)

\( \bar{S}_M \) is the reflection of \( \bar{V}_A \) off the surface of the mirror (or the inverse of the incident light ray from the pupil) in the mirror’s coordinate system

\( \mathbf{R}_{NM} \) is the rotation matrix used to transform points in the NA camera’s coordinate system to the mirror’s coordinate system

\( \mathbf{T}_{NM} \) is the translation vector used to transform points in the NA camera’s coordinate system to the mirror’s coordinate system

\( \mathbf{R}_{MN} \) is the rotation matrix used to transform points in the mirror’s coordinate system to the world coordinate system

\( \mathbf{T}_{MN} \) is the translation vector used to transform points in the mirror’s coordinate system to the world coordinate system

We will assume a right-handed coordinate system, and define the rotation matrices and translation vectors above as follows:
\[ \mathbf{R}_w = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta_T) & \sin(\beta_T) \\ 0 & -\sin(\beta_T) & \cos(\beta_T) \end{bmatrix} \begin{bmatrix} \cos(\beta_H) & 0 & \sin(\beta_H) \\ 0 & 1 & 0 \\ -\sin(\beta_H) & 0 & \cos(\beta_H) \end{bmatrix} \]  
(83)

\[ \mathbf{T}_w = \begin{bmatrix} 0 \\ 0 \\ -f_w \end{bmatrix} \]  
(84)

\[ \mathbf{R}_{NM} = \begin{bmatrix} \cos(\theta_N) & 0 & \sin(\theta_N) \\ 0 & 1 & 0 \\ -\sin(\theta_N) & 0 & \cos(\theta_N) \end{bmatrix} \]  
(85)

\[ \theta_N = (\rho_H - 90^\circ) \]  
(86)

\[ \mathbf{T}_{NM} = \begin{bmatrix} 0 \\ 0 \\ D_m - f_N \end{bmatrix} \]  
(87)

\[ \mathbf{R}_{MIR} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\rho_T) & \sin(\rho_T) \\ 0 & -\sin(\rho_T) & \cos(\rho_T) \end{bmatrix} \begin{bmatrix} \cos(\rho_H) & 0 & \sin(\rho_H) \\ 0 & 1 & 0 \\ -\sin(\rho_H) & 0 & \cos(\rho_H) \end{bmatrix} \]  
(88)

\[ \mathbf{T}_{MIR} = \begin{bmatrix} D_m - D_s \\ 0 \\ 0 \end{bmatrix} \]  
(89)

where:

- \( \beta_T, \beta_H \) are the vertical and horizontal angles of rotation of the WA camera with respect to the world coordinate frame (in the system reported here these are system constants).

- \( \rho_T, \rho_H \) are the vertical and horizontal angles of rotation of the mirror with respect to the world coordinate frame (in the system reported here, these are measured values).
$D_c$ is the distance between the centres of the WA camera and NA camera sensors (in the system reported here, these is a system constant)

$D_m$ is the distance between the centre of the NA camera sensors and the rotation point $M$ of the mirror (in the system reported here, these is a system constant)

**Gaze calculation using single light source**

Once the 3-D pupil location has been determined, the second step in calculating the angle of gaze is to use the light ray from the glint to the NA camera’s image plane as shown in Figure 3.3. If we treat the light source (array of infrared LEDs) as a single point source, then the glint $G$ is the reflection of that light off the surface of the cornea. In this case, $G$, $C$ and $R$ lie on the same line. Specifically,

$$R + (\alpha_g + r)\bar{R} = C$$

(90)

where:  $\alpha_g$ is a scale factor

$r$ is the radius of curvature of the cornea (and also the distance between $G$ and $C$)

The radius $r$ is constant for any given user. The Gullstrand model uses an approximate value of 7.7mm. The system can be additionally calibrated for $r$ using a single calibration point (i.e., if $\bar{E}$ is known, the computation can be reversed to find a suitable value for $r$ for any given user). Also, multiple calibration points could be used to provide a more robust estimate for $r$. Such calibration could be done once per user.
To arrive at values for $R$ and $\overline{R}$, we trace backward along the light ray resulting in the image of the glint in the NA camera’s image plane:

$$\begin{align*}
B_N &= \begin{bmatrix} u_G / f_N \\ v_G / f_N \\ 1 \end{bmatrix} \\
B_M &= R_{NM} (B_N - T_{NM}) \\
\overline{B}_M &= B_M - F_M \\
R_M &= F_M + \alpha M \overline{B}_M \\
R_{Mf} &= 0 \\
V_R &= \frac{\overline{B}_M}{\| \overline{B}_M \|} \\
\overline{R}_M &= \begin{bmatrix} V_{Rx} \\ V_{Ry} \\ -V_{Rz} \end{bmatrix} \\
R &= R_{Mf} (R_M - T_{Mf}) \\
\overline{R} &= R_{Mf} (\overline{R}_M - T_{Mf})
\end{align*}$$

where: $u_G$ is the x-coordinate of the glint image in the NA camera’s image plane

$v_G$ is the y-coordinate of the glint image in the NA camera’s image plane

$B_N$ is the intersection of the light ray from the glint with the NA camera plane in the NA camera’s coordinate system

$\overline{B}_M$ is the vector $B_N$ in the mirror’s coordinate system

$R_M$ is the intersection of $\overline{B}_M$ with the mirror plane in the mirror’s coordinate system
\( \alpha_s \) is a scale factor used to solve for \( R_M \) using equations 94 and 95

\( V \) is the unit vector representing the incident light ray from the glint off the surface of the mirror (in the mirror's coordinate system)

\( R_M \) is the reflection of \( V \) off the surface of the mirror (or the inverse of the incident light ray from the glint) in the mirror's coordinate system

\( R_{NM} \), \( T_{NM} \), \( R_{MP} \), and \( T_{MP} \) are as above

Equation 90 gives a system of 3 equations and 4 unknowns (\( \alpha_s \) and \( C \)). Having calculated the 3-D position of the pupil \( P \) above, using the Gullstrand model and the concave spherical refractive lens formula for paraxial rays, we can introduce another constraint as follows.

Figure E. 3 Ray diagram for computing location of image of pupil
Figure E. 3 shows a ray diagram that is useful for visualizing where the image of the pupil will appear, given that we are modeling the cornea as a concave lens. We start by computing the dipotric power of the cornea as a lens:

\[ F = \frac{n' - n}{r} \]  

(100)

where \( n' \) and \( n \) are the indices of refraction for air and the interior of the cornea, respectively. Next we calculate the vergence on the outside of the cornea:

\[ L = \frac{n}{l} \] 

(101)

Now we combine equations 100 and 101 to calculate the vergence on the inside of the cornea:

\[ L' = F + L + \] 

(102)

which we then use to calculate the distance of the pupil’s image from the corneal surface:

\[ l' = \frac{n'}{L'} \] 

(103)

From Gullstrand's model (see Figure 3.2), we know that \( n' = 1, \) \( n = 1.376, \) \( r = .0077 \) and \( l = 0.004146 \), which gives values of \( F = -48.83D \) and \( L = +331.9D \). From the above equations, we find that \( L' = -48.83 + 331.9 = +283D \) and \( l' = 3.53mm \).

Finally, we compute the “radius” of the sphere defining the locus on which the pupil’s image (as seen by the cameras) will be located as:

\[ r_p = r - l' \] 

(104)

From the values calculated above, we get \( r_p = 4.17mm \).

Therefore, we can apply the constraint:
\[ \|P - C\| = r_p \]  
\[ \text{and together with equation 90, solve for the centre of the cornea } C. \text{ The angle of} \]
\[ \text{gaze is then given by:} \]
\[ \overline{E} = \frac{P - C}{\|P - C\|} \]  

**Gaze calculation using multiple light sources**

We have already shown how to calculate \( R \) and \( \overline{R} \) above. Now, as shown in Figure E.4, we can trace backward along the light ray resulting in the image of the secondary light source in the NA camera’s image plane:

\[ B'_N = \begin{bmatrix} u' / f_N \\ v' / f_N \\ 1 \end{bmatrix} \]  
\[ B'_M = R_{NM} (B'_N - T_{NM}) \]  
\[ \overline{B}'_M = B'_M - F_M \]  
\[ R'_M = F_M + \alpha B'_M \]  
\[ R'_{M_2} = 0 \]  
\[ V'_R = \overline{B}'_M / \|B'_M\| \]  
\[ \overline{R}'_M = \begin{bmatrix} V'_{Rx} \\ V'_{Ry} \\ -V'_{Rz} \end{bmatrix} \]  
\[ R' = R_{MW} (R'_M - T_{MW}) \]
\[ \overrightarrow{R'} = R_{MIP} (\overrightarrow{R_M'} - T_{MIP} ) \] (115)

where:

- \( u' \) is the x-coordinate of the image of the reflection of the secondary light source off the cornea in the NA camera's image plane
- \( v' \) is the y-coordinate of the image of the reflection of the secondary light source off the cornea in the NA camera's image plane
- \( \overrightarrow{B_M'} \) is the intersection of the light ray from the reflection of the secondary light source off the cornea with the NA camera plane in the NA camera's coordinate system
- \( \overrightarrow{B_N'} \) is the vector \( \overrightarrow{B_M'} \) in the mirror's coordinate system
- \( \overrightarrow{B_M'} \) is the intersection of \( \overrightarrow{B_M'} \) with the mirror plane in the mirror's coordinate system
- \( \omega_n' \) is a scale factor used to solve for \( R_M' \) using equations 110 and 111
- \( V_r' \) is the unit vector representing the incident light ray from the corneal reflection of the secondary light source off the surface of the mirror (in the mirror's coordinate system)
- \( \overrightarrow{R_M'} \) is the reflection of \( V_r' \) off the surface of the mirror (or the inverse of the incident light ray from the corneal reflection of the secondary light source) in the mirror's coordinate system
- \( R_{NM} \), \( T_{NM} \), \( R_{MIP} \) and \( T_{MIP} \) are as above
Figure E.4 Calculation of corneal centre using multiple light sources
In addition, we define $L_0$ as the location of the secondary light source we wish to track. Then $L$ is the intersection of the light ray from the secondary light source (we treat it as a single ray parallel to the x-axis). We now have enough information to compute $C$ as follows:

$$C = L + (\alpha_L + r)\overline{L}$$

$$G' = L + \alpha_L \overline{L}$$

$$G' = R' + \alpha' \overline{R}'$$

where: $\alpha_L$ and $\alpha'$ are scale factors

$\overline{L}$ is unit vector of ray from $L$ to $G'$

$G'$ is the location of the corneal reflection of the secondary light source

Together with equation 90, the three equations above form a system of 12 equations with 12 unknowns. Therefore, we can solve for $C$.

Now we can use equations 72 - 82 as before to calculate $S$ and $\overline{S}$. Then we can calculate $P$ up to a scale factor $\alpha_s$ using equation 4, and together with equation 105 (which provides an additional constraint on the value for $P$) calculate the 3-D location of the pupil. Note that with the constraint in equation 105 we get two possible values for $P$. In this case, we are looking for the value with the smaller $z$ value, since we assume the person is facing the computer monitor. Finally, we can use equation 106 as before to calculate the gaze direction $\overline{E}$.

Although the theory above has been presented for tracking one additional light source, since the system designed for this thesis has several LEDs, more than one additional light source could be used simultaneously. This would lead to an over-
constrained system of equations, and simple approximation techniques could be used to arrive at a more robust solution for the corneal centre of curvature and, ultimately, the gaze calculation.

**Ellipse fitting algorithm**

The algorithm used to fit an ellipse to the contour points found is the one described in [60]. This algorithm treats the contour points as the points on a conic section with some noise, and uses a direct least squares approach to solve for the parameters of the conic section. This approach has three advantages over other ellipse-fitting algorithms:

a) It always returns an ellipse, even with very noisy data. Most other algorithms solve for a general conic, and may return a parabola or hyperbola as the best fit to the data. Since we are only interested in ellipses, this is an important strength of this approach.

b) It can be solved naturally by a generalized eigensystem, making it very attractive computationally.

c) It is fairly robust and very efficient, which is important in a real-time application such as ours.

The following is a brief description of the algorithm in [60]. First, we represent a general conic with an implicit second order polynomial:

\[ F(\mathbf{a}, \mathbf{x}) = \mathbf{a} \cdot \mathbf{x} = a x^2 + bxy + cy^2 + dx + ey + f = 0 \]  \hspace{1cm} (119)

where: \[ \mathbf{a} = [a \ b \ c \ d \ e \ f]^T \] is a vector representing the conic parameters.
\[ x = [x^2 \ xy \ y^2 \ x \ y \ 1]^T \] is a vector containing the polynomial terms for a point \((x, y)\) on the conic.

Given a point \((x_i, y_i)\) in our dataset, we can define the "algebraic distance" of the point to the conic in (119) as \(F(a, x_i)\). The fitting of a general conic can then be performed by minimizing the sum of squared algebraic distances of \(N\) points as follows:

\[
\sum_{i=1}^{N} F(a, x_i)^2
\]

(120)

The optimal parameter vector \(a\) is thus found. To find specific types of conics and to avoid the trivial solution \(a = 0\), various constraints are placed on the parameter vector \(a\). For an ellipse, the fitting method described in [60] makes use of the constraint that, for an ellipse, \(b^2 - 4ac < 0\), or more specifically, \(4ac - b^2 = 1\). This quadratic constraint can be expressed as:

\[
\mathbf{a}^T \mathbf{C} \mathbf{a} = 1
\]

(121)

where:

\[
\mathbf{C} =
\begin{bmatrix}
0 & 0 & 2 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

We can define a design matrix \(\mathbf{D} = [x_1 \ x_2 \ \cdots \ x_N]^T\) and a scatter matrix \(\mathbf{S} = \mathbf{D}^T \mathbf{D}\) and, using a Lagrange multiplier \(\lambda\), arrive at the following equation which, when combined with equation (121), can be solved for the optimal parameter vector \(\mathbf{a}\):

\[
\mathbf{S} \mathbf{a} = \lambda \mathbf{C} \mathbf{a}
\]

(122)

The ellipse parameters are recovered from \(\mathbf{a}\) as follows. We first define a matrix:
\[
\mathbf{A} = \begin{bmatrix}
\frac{a}{b/2} & \frac{b}{c/2}
\end{bmatrix}
\] 

(123)

such that \[ \mathbf{A} \begin{bmatrix} x_o \\ y_o \end{bmatrix} + \begin{bmatrix} d/2 \\ e/2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \] where \((x_o, y_o)\) is the centre of the ellipse. In this way, the centre of the ellipse can be found directly. The length and angle of rotation of the axes are calculated as follows. We first scale all the conic parameters as follows:

\[
a' = s_a, b' = s_b, c' = s_c, d' = s_d, e' = s_e, f' = s_f
\]

(124)

where \(s_1 = \sqrt{\frac{\det(A)}{2}}\). We then define a second matrix:

\[
\mathbf{A}' = \begin{bmatrix}
a's_2 & b's_2/2 \\
b's_2/2 & c's_2
\end{bmatrix}
\]

(125)

where \(s_2 = -1/(f + a'x_0^2 + b'y_0^2 + c'y_0^2 + d'x_0 + e'y_0)\). If \(\lambda_1\) and \(\lambda_2\) represent the eigenvalues of \(\mathbf{A}'\) such that \(\lambda_1 > \lambda_2\), and \(\mathbf{v}_2 = [e_1, e_2]^T\) is the eigenvector corresponding to \(\lambda_2\), then the length of the ellipse axes can be defined as \(l_{\text{width}} = 2\sqrt{1/\lambda_2}\) and \(l_{\text{height}} = 2\sqrt{1/\lambda_1}\), and the angle of rotation as \(\theta_{\text{ellipse}} = \arctan(e_2/e_1)\).