A Single-Camera Gaze Tracker using Controlled Infrared Illumination

Examensarbete utfört i Datorseende vid Tekniska högskolan i Linköping
av
Marcus Wallenberg
LITH-ISY-EX--09/4199--SE
Linköping 2009
A Single-Camera Gaze Tracker using Controlled Infrared Illumination

Examensarbete utfört i Datorseende vid Tekniska högskolan i Linköping av

Marcus Wallenberg

LITH-ISY-EX--09/4199--SE

Handledare: Fredrik Larsson
ISY, Linköpings universitet

Examinator: Per-Erik Forssén
ISY, Linköpings universitet

Linköping, 18 February, 2009
Gaze tracking is the estimation of the point in space a person is “looking at”. This is widely used in both diagnostic and interactive applications, such as visual attention studies and human-computer interaction. The most common commercial solution used to track gaze today uses a combination of infrared illumination and one or more cameras. These commercial solutions are reliable and accurate, but often expensive. The aim of this thesis is to construct a simple single-camera gaze tracker from off-the-shelf components.

The method used for gaze tracking is based on infrared illumination and a schematic model of the human eye. Based on images of reflections of specific light sources in the surfaces of the eye the user’s gaze point will be estimated.

Evaluation is also performed on both the software and hardware components separately, and on the system as a whole. Accuracy is measured in spatial and angular deviation and the result is an average accuracy of approximately $1^\circ$ on synthetic data and $0.24^\circ$ to $1.5^\circ$ on real images at a range of 600 mm.

The solution implemented uses only standard off-the-shelf components, and provides a low-cost alternative to complex and costly commercial gaze tracking systems with a reasonable accuracy for some simple applications.
Abstract

Gaze tracking is the estimation of the point in space a person is “looking at”. This is widely used in both diagnostic and interactive applications, such as visual attention studies and human-computer interaction. The most common commercial solution used to track gaze today uses a combination of infrared illumination and one or more cameras. These commercial solutions are reliable and accurate, but often expensive. The aim of this thesis is to construct a simple single-camera gaze tracker from off-the-shelf components.

The method used for gaze tracking is based on infrared illumination and a schematic model of the human eye. Based on images of reflections of specific light sources in the surfaces of the eye the user’s gaze point will be estimated.

Evaluation is also performed on both the software and hardware components separately, and on the system as a whole. Accuracy is measured in spatial and angular deviation and the result is an average accuracy of approximately $1^\circ$ on synthetic data and $0.24^\circ$ to $1.5^\circ$ on real images at a range of 600 mm.

The solution implemented uses only standard off-the-shelf components, and provides a low-cost alternative to complex and costly commercial gaze tracking systems with a reasonable accuracy for some simple applications.

Sammanfattning


Metoden som används baseras på infraröd belysning av ögats olika ytor och skikt. Bilder av reflexerna från dessa används sedan för att uppskatta vart användarens användarens blick är riktad.

Utvärdering av lösningens mjukvaru- och hårdvarudelar sker både separat och i kombination. Noggrannhet mäts i spatiell avvikelse och vinkelfel. En noggrannhet på runt $1^\circ$ på syntetiska bilddata och mellan $0.24^\circ$ och $1.5^\circ$ uppnås på ett avstånd av 600 mm.
Lösningen som implementerats består enbart av allmänt tillgängliga komponenter, och kan användas som ett lågkostnadsalternativ till dyra kommersiella lösningar i vissa enklare tillämpningar.
Many people deserve thanks for their guidance and help during the course of this project. I would like to thank my examiner Per-Erik Forssén and my supervisor Fredrik Larsson, for giving me the opportunity to carry out this project and for all the help on matters both theoretical and practical along the way.

I would also like to thank Sören Hansson, Jean-Jaques Moulis, JohanWiklund and Anders Nilsson for their insights into hardware issues and electronics, and for the use of their equipment and facilities. My thanks also go to David Bäck for providing a different gaze tracking solution against which to compare my own.

Finally, Maria Magnusson for initially rousing my interest in image processing and computer vision.

Marcus Wallenberg
Linköping, February 2009
Contents

1 Introduction 3
  1.1 Background .................................................. 3
  1.2 Problem Description ......................................... 3
  1.3 Method .......................................................... 3
  1.4 Thesis Outline .................................................. 4
  1.5 Concepts and Terminology ...................................... 4

2 Visual Perception and Attention in Humans 5
  2.1 Human Vision ................................................... 5
  2.2 Visual Attention ................................................ 6
  2.3 Eye Movement ................................................... 8
  2.4 The Gullstrand Schematic Eye .................................. 9

3 Methods and Applications of Eye and Gaze Tracking 11
  3.1 Methods of Eye and Gaze Tracking .......................... 11
    3.1.1 Electro-oculography ..................................... 11
    3.1.2 Scleral Contact Lens or Search Coil ..................... 12
    3.1.3 Image-based Methods - Video Oculography ............... 13
    3.1.4 Combined Pupil and Corneal Reflection Methods .......... 13
  3.2 Applications of Eye and Gaze Tracking ....................... 14
    3.2.1 Neuroscience and Psychology .......................... 14
    3.2.2 Usability Studies ......................................... 15
    3.2.3 Advertising and Product Labelling ....................... 15
    3.2.4 Human-Computer Interaction ............................. 16

4 The Shih-Liu and Hennessey-Noureddin-Lawrence Gaze Trackers 17
  4.1 The Shih-Liu Gaze Tracker ................................. 17
    4.1.1 Background ............................................... 17
    4.1.2 Method .................................................... 18
    4.1.3 Algorithm Overview ...................................... 21
  4.2 The Hennessey-Noureddin-Lawrence Gaze Tracker ........... 22
    4.2.1 Background ............................................... 22
    4.2.2 Method .................................................... 22
5 Implementation

5.1 Hardware
5.1.1 Camera
5.1.2 Illumination Control
5.1.3 Illumination Control/Camera Interface
5.1.4 Installation and Setup Geometry

5.2 Software
5.2.1 Frame Classification and Segmentation
5.2.2 Pupil Location Algorithm
5.2.3 Glint Location Algorithm
5.2.4 Calculation of Corneal Center
5.2.5 Calculation of Pupil Center
5.2.6 Calculation of Gaze Point
5.2.7 Calibration of Gaze Point

6 Evaluation

6.1 Hardware Evaluation
6.1.1 Camera Evaluation
6.1.2 Illumination Control Evaluation

6.2 Software Evaluation
6.2.1 Generation of Synthetic Image Data
6.2.2 Software Evaluation on Synthetic Image Data

6.3 System Evaluation
6.3.1 Experimental Setup
6.3.2 Data Acquisition

7 Results and Discussion

7.1 Results
7.1.1 Camera Performance
7.1.2 Illumination Control Performance
7.1.3 Performance on Synthetic Data
7.1.4 Performance on Real Data

7.2 Discussion
7.2.1 Synthetic Data
7.2.2 Real Data
7.2.3 General
7.2.4 Comparison to the Original Henessey-Noureddin-Lawrence Gaze Tracker
7.2.5 Comparison to the Bäck Gaze Tracker

8 Conclusions and Future Work

8.1 Conclusions
8.2 Future Work

Bibliography
List of Figures

2.1 The human eye. ........................................... 6
2.2 Examples of the "pop-out" effect. .......................... 7
2.3 The Gullstrand schematic eye. ............................. 9

3.1 Features of the eye used in passive image-based methods. .... 13
3.2 Combined corneal and pupil reflections. ...................... 14
3.3 Uncertainty in gaze point introduced by the spatial extent of the fovea. ........................................... 16

4.1 Auxiliary coordinate system (Shih-Liu). ...................... 18
4.2 Second auxiliary coordinate system (Shih-Liu) ................ 20
4.3 Pupil center back-projection (Shih-Liu) ....................... 20
4.4 Auxiliary coordinate system (Hennessey-Noureddin-Lawrence) .. 23
4.5 Pupil center estimation (Hennessey-Noureddin-Lawrence) .... 24

5.1 The Point Grey Research Flea2 (FL2G-13S2M-C). ............. 28
5.2 Illumination control circuit wiring diagram. .................. 29
5.3 Layout of the HiRose HR25-7TR-8SA connector. ............... 30
5.4 Structure of the opto-isolated GPIO port. .................... 30
5.5 Structure and geometry of the hardware setup. ............... 31
5.6 Example of pupil location algorithm. ......................... 33
5.7 Example of glint location algorithm. ........................ 34
5.8 Calibration pattern. ...................................... 36

6.1 Examples of synthetic images. ................................ 39
6.2 Cornea and gaze point locations in sample set 1. .............. 41
6.3 Cornea and gaze point locations in sample set 2. .............. 42
6.4 Simplified pupil center error estimation. ..................... 43

7.1 Camera performance. ...................................... 46
7.2 Strobe control performance. ................................ 46
7.3 Performance on synthetic images, set 1 (stationary corneal center). 48
7.4 Performance on synthetic images, set 2 (mobile corneal center) ... 49
7.5 Reconstruction of calibration pattern. ........................ 50
7.6 Corrections in the x and y directions. ....................... 50
7.7 Mean error on first evaluation set vs. frames used in gaze point estimation for points within the calibration pattern. .......... 51
7.8 Reconstruction of first evaluation pattern. ................... 52
7.9 Mean error on second evaluation set vs. frames used in gaze point estimation for points within the calibration pattern. .......... 53
7.10 Reconstruction of second evaluation pattern. ................. 54
Chapter 1

Introduction

In this chapter, a description of the problem background, the methods used, and the outline of this thesis will be given. A short description of some specific technical terms used in the report is also provided.

1.1 Background

The aim of this project is the creation of a gaze tracking solution. Gaze tracking is the determination of a subject’s gaze point, that is, what the subject is "looking at". Gaze tracking has many different applications, both diagnostic and interactive. Many commercial solutions for gaze tracking exist, but their cost is often prohibitive to many potential users. The solution created in this project should ideally be simple and easy to use, but also affordable and not reliant on complicated hardware.

1.2 Problem Description

The problem of tracking gaze has three main parts. In order to find a subject’s gaze point, one must determine the location of the subject’s eyes, their orientation and the point where the subject’s gaze falls on a target area.

1.3 Method

The method selected for this gaze tracking solution was the combined corneal reflection/pupil reflection method (further described in subsection 3.1.4). Two specific variants of this method were selected and are described in chapter 4. A camera was purchased, an illumination control circuit devised and gaze tracking software implemented in MATLAB. The solution was implemented for offline (non real-time) use and uses pre-captured video sequences. For evaluation purposes software for generating synthetic images was also implemented.
1.4 Thesis Outline

This report is organized into eight chapters. In chapter 2, the human visual system and human visual attention are described. The Gullstrand schematic eye is also described and illustrated. Chapter 3 gives an introduction to various methods of eye and gaze tracking and their applications. Chapter 4 explains two closely related methods of constructing a simple gaze tracking solution and the theory behind these methods. Chapter 5 describes the hardware and software components of the implemented solution and chapter 6 the evaluation procedure used to determine how well they are capable of performing. Chapter 7 contains the results of the evaluation and compares them to a few other gaze tracking solutions. This chapter also contains a discussion of the results. Chapter 8 contains conclusions and suggestions for future work. The principal lessons learned and the problems encountered will be described here.

Each chapter begins with a brief summary of its content, with references to where in the chapter specific topics are discussed. This report contains some colour images which are important to the illustration of the evaluation method used and should therefore ideally be viewed in colour.

1.5 Concepts and Terminology

This section explains some of the concepts and terms used in the report. In various publications and in general discussion, the terms eye tracking and gaze tracking are used interchangeably. Here eye tracking denotes the tracking of the eyes only, whether measured relative to the subject’s head or not, and is used to describe methods which give no explicit information about the subject’s gaze point. Gaze tracking, on the other hand, denotes methods that explicitly deliver data about gaze point, and may or may not track both the position and orientation of the eyes.

Similarly, the terms gaze point and point of gaze (sometimes abbreviated as PoG or POG) denote the same quantity in different publications. In this report, the term gaze point is used.

The terms synthetic and real/video data are also used. In this report, synthetic data and/or results imply that the data and corresponding results have been obtained using raytraced synthetic images. When data or results are described as real or video data or results, this implies that they were captured using the camera or calculated from captured video sequences.
Chapter 2

Visual Perception and Attention in Humans

In this chapter, a brief description of some central aspects of human vision will be given. A basic understanding of human vision is more or less necessary in development of eye and gaze tracking systems. Therefore, this chapter provides a simple explanation of the concepts most central to this application. First, a brief description of the human eye and visual system is given, then the concept of visual attention is described. The third section deals with the movement of human eyes, and the typical types of motion associated with visual attention and information gathering. The fourth and final section describes the schematic eye model used in the implemented system described in chapter 5.

2.1 Human Vision

Vision is perhaps the most important of the human senses and has been essential to the survival of our species. It is a sense often relied upon to make judgments and plays a large part in human communication. The visual system is the most complex of the sensory systems, encompassing sensory cells in the eyes, nerves connecting the eyes and brain and parts of the brain itself [11]. The structure of the human eye is depicted in figure 2.1.

Simply put, human vision begins when light passes through the cornea and lens of the eye and strikes the retina. On the retina two different types of photosensitive cells, commonly referred to as rods and cones, are present. The cone cells, numbering approximately seven million, are sensitive to well-illuminated stimuli and have uneven frequency response within the visible spectrum which allows for colour differentiation and recognition. The rod cells, of which there are about 125 million, are more sensitive to poorly illuminated stimuli than cone cells and are primarily responsible for perceiving differences in brightness. In reality, true colour vision is achieved through a combined use of both cell types. These cells are unevenly distributed over the retina, giving different regions in the visual field
different visual acuity. The cone cells are concentrated in a region known as the fovea, which is a small part of the retina with high visual acuity. The rod cells are distributed outside the fovea, giving a more even sense of brightness. In practice this means that high-acuity colour vision is accomplished by positioning the eye so that the image of the object or feature studied falls on the fovea (more about this in later sections). From the retina, the electrical impulses created by the photosensitive cells travel through layers of cells (called horizontal, bipolar, amacrine and ganglion cells) in front of the retina to the lateral geniculate nucleus, located in the thalamus and onward to the visual (striate) cortex of the brain. In the visual cortex, the patterns on the retina trigger different nerve cells that respond to small features such as line segments and local orientations, and these features are then combined to form the image we see and interpret [11].

Figure 2.1. The human eye.

2.2 Visual Attention

Due to the massive amount of visual stimulation constantly present in everyday situations, it would be impossible (or at the very least requiring a brain very different from our own) to pay attention to all the visual stimuli we are capable of perceiving. There exists a need to prioritize certain stimuli for further and higher-level processing, which is what defines attention. Attention can be involuntary (when some external stimulus causes the mind to focus on it) or selective (when there is a conscious choice to direct focus to some point or object). This is an important ability which allows us to interpret complex signals despite having limited processing capacity in the brain. Selective attention in particular leads to increased processing capacity devoted to stimuli perceived to be important, and moderation of information from other sources. Another aspect of selective attention is the possibility to direct attention overtly or covertly. Overt visual attention is when the stimulus or object coinciding with the line of sight is actually the one being attended to. Conversely, covert visual attention is when the object of at-
2.2 Visual Attention

tention is in fact not the one in the line of sight (a technique which can be useful in darkness, where faintly lit objects cannot be perceived through foveal vision). An example of involuntary attention is the "pop-out" effect an out-of-place object or colour, or a regional disruption of an otherwise regular pattern can have. An example of this is illustrated in figure 2.2. This in turn is important for noticing unknown, potentially harmful or otherwise important features in the environment even when not actively looking for them (feature saliency) [11].

Figure 2.2. Examples of the "pop-out" effect. A regular area within an irregular pattern (left), a change in colour (center) and a disruption of a regular pattern (right).

In humans, the concept of visual attention is linked to the anatomy of the eye and oculomotor plant. Since high-acuity vision is possible only in a small region of the retina, the entire eye must be oriented in a specific way in order to focus visual attention on a particular spot in the field of view. Also, in order to stereographically process an object, the gaze directions of both eyes usually converge on a single point. Thus, the object of visual attention is closely linked to eye orientation and thereby to gaze direction [11], [5].
2.3 Eye Movement

Early experiments investigating selective attention focused on selective visual attention. This was done partly because of the known importance of vision in cognition and partly because visual attention is among the easiest to control and study. Emile Javal observed in 1878 that reading (a common application of selective visual attention) was not performed by smoothly scanning the text as one might be inclined to assume, but rather by a series of stationary eye orientations with jumping, rapid movements in between. These rapid movements became known as saccades and the stationary orientations as fixations. Later studies determined that it was in fact during these fixations that the gathering of information (the actual reading) took place [11].

Nowadays, eye movement is typically divided into three categories. These are saccade, smooth pursuit and fixation. The saccades are rapid eye movements designed to reorient the eye and bring another part of the field of view into alignment with the fovea. Saccades can be both voluntary and involuntary and typically range in duration from 10 to 100 milliseconds. During the saccade motion, it is virtually impossible to focus on anything specific and the eye is therefore in practice blind during this time. Smooth pursuit is exhibited when the eye is following an object moving along a smooth, continuous path. The eye matches the angular change of an object in the field of view in order to follow it. Fixations occur when attention is directed at a specific, stationary point. These fixations do not in fact imply that the eye is fixed, but rather that the deviations in orientation are small and brief. Fixations typically last between 150 and 600 milliseconds and take up approximately 90% of viewing time. The small, rapid semi-random movements (known as microsaccades and typically being one to two arcminutes in magnitude) are designed to move the eye and ensure that the same receptor cells are not continuously exposed to the same stimulus as this would cause them to become 'saturated' and temporarily lose their sensitivity. If, by mechanical means or paralytic agents, the eye is rigidly fixed vision typically fades to virtual blindness within one second if the scene remains static [5].
2.4 The Gullstrand Schematic Eye

The Gullstrand Schematic Eye is a simple model of the optical properties of the human eye. It was proposed by Nobel laureate Allvar Gullstrand in his 1911 publication *Einführung in die Methoden der Doppeltrik der Augen des Menschen* [1]. The Gullstrand model is a further development of the earlier Helmholtz-Laurance model with updated values for some of the physical and optical properties of the human eye [2].

The standard version of the Gullstrand schematic eye models the human eye as three bodies of refractive media (three surfaces with associated curvatures and refractive indices). These model the anterior corneal, posterior aqueous humour/anterior lenticular and posterior lenticular/vitreous humour boundaries [2]. The Gullstrand model disregards refraction in the posterior corneal/anterior aqueous humour boundary because of the small difference in refractive index between the two media and comparatively low refractive power. The properties and geometry are described in figure 2.3 and table 2.1. Several similar schematic eyes exist, such as the above mentioned Helmholtz-Laurance eye, the Le Grand eye (closely based on the Gullstrand eye) and other models such as the Emsley, Schwiegerling and Liou-Brennan schematic eyes.

<table>
<thead>
<tr>
<th>Boundary</th>
<th>Surface</th>
<th>Radius [mm]</th>
<th>Thickness [mm]</th>
<th>Refractive Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>cornea</td>
<td>1</td>
<td>7.8</td>
<td>3.6</td>
<td>1.336</td>
</tr>
<tr>
<td>cornea/lens</td>
<td>2</td>
<td>10.0</td>
<td>3.6</td>
<td>1.413</td>
</tr>
<tr>
<td>lens/vitreous</td>
<td>3</td>
<td>6.0</td>
<td>16.97</td>
<td>1.336</td>
</tr>
</tbody>
</table>

Figure 2.3. The Gullstrand schematic eye.
Chapter 3

Methods and Applications of Eye and Gaze Tracking

In this chapter, both historical and current methods of eye and gaze tracking will be described (see section 3.1). Examples of their modern applications in different fields will also be given (see section 3.2).

3.1 Methods of Eye and Gaze Tracking

Many different methods of eye and gaze tracking exist. Some have been used since the 19th century, some have been developed in the last decade. What they all have in common is that they, be it through electrical activity, mechanical motion or optical properties, seek to determine the orientation and motion of the eyes. In this section some of the most widely used methods are described.

3.1.1 Electro-oculography

Electro-oculography was developed in order to non-invasively measure eye orientation. It was the most widely used method for this task during the latter half of the twentieth century and is still used today, albeit on a smaller scale. The method relies on recording electric potential at specific points around the eye cavity and thus determine muscle tone and eye orientation. The recorded potential differences are small, typically in the range 15-200 $\mu$V with a nominal sensitivity of 20 $\mu$V per degree of eye movement. The method measures eye movement relative to the head, and does therefore not provide any information about the point of gaze unless the head is also tracked or fixed. Another drawback of this method is that is requires highly specialized equipment, and an environment free from electromagnetic interference [5].
3.1.2 Scleral Contact Lens or Search Coil

The scleral contact lens was the first reliable method for determining eye orientation and emerged in the late 19th century. Early methods were purely mechanical in nature and relied on a lens or ring which was mounted directly on the cornea of the subject and linked to a pen which would trace the eye movement of the subject [5].

Later developments did away with the mechanical linkage connecting the contact lens and the pen, and replaced it with a search coil in the contact lens itself. With a known external electromagnetic field applied, the changes in orientation could be calculated from the induced current in the coil. With refinement, this method became very accurate and is still used when precision is paramount. Resolutions of five to ten arc seconds are possible using this method. The drawbacks are, primarily, that the orientation of the eye is measured relative to the head, making this method unsuitable for gaze point determination. The subject’s head must also remain within the field-generating frame. Additionally, the application and use of the scleral contact lens is invasive, causes discomfort in the subject and requires highly specialized equipment in the hands of a skilled operator [5].
3.1 Methods of Eye and Gaze Tracking

3.1.3 Image-based Methods - Video Oculography

Various more or less active image-based methods for eye and gaze tracking exist. The common element for these methods is that they use visible features of the eye to determine its position and orientation. Popular features include the observed shape and location of the pupil and the limbus (the boundary between the iris and the sclera). The features are illustrated in figure 3.1. Also reflections (known as Purkinje reflections) of directed light sources in the surfaces of the eye are often used. Typically, infrared illumination is preferred in order to provide easier detection of the desired reflections and less of a distraction to the subject. The illumination hardware and camera are typically either separate from the subject or mounted on spectacles or a headset [5].

In image-based methods of gaze tracking it is often the optical axis of the eye which is determined and used to estimate gaze direction. This is not entirely accurate, since the actual line of sight can deviate from the optical axis. This deviation, however is usually small and can be compensated for by calibration [5].

![Figure 3.1. Features of the eye used in image-based methods.](image)

3.1.4 Combined Pupil and Corneal Reflection Methods

A category of methods that has gained popularity in recent years is the use of image-based methods combined with a more advanced illumination setup. This subgroup of the image-based methods uses different types of reflections and features combined to determine both the location of the eye in three dimensions and its orientation. The advantage of this method is that, used correctly, it provides gaze point determination without special head tracking. The typical approach is to use the above mentioned Purkinje reflections to determine the location of the eye, and the pupil reflection to determine its orientation (see figure 3.2). This method can be adapted for use with very little specialized hardware and provide reasonable accuracy at a significantly lower cost than many other methods [5]. The systems described in chapters 4 and 5 are of this type.
3.2 Applications of Eye and Gaze Tracking

The applications of eye and gaze tracking span a wide variety of areas. In general, these applications can be characterized as either diagnostic or interactive. That is to say that they record eye movement and/or gaze either for the sake of studying the subject or to obtain an input to respond to.

3.2.1 Neuroscience and Psychology

Eye tracking studies are widely used in the field of attentional neuroscience. Even though it has been demonstrated that it is possible to attend to stimuli outside the fovea, the existence of a link between visual attention and eye movement has been well established in many situations. In order to better understand the mechanisms of attention, eye tracking studies have been helpful. In recent years there have also been studies combining eye tracking with brain imaging techniques such as fMRI, where the combined results can be used to better map specific functions related to vision, visual attention and oculomotor function [5].

One of the oldest applications of eye tracking is the study of reading behaviour. The studies concern the visual search patterns involved in gathering information from text, sheet music and other forms of written material. In a broader context, the study of scene perception also makes use of this method in trying to quantify what features or attributes are salient, important or interesting in a scene and how much information can be gathered in a given length of time. This in turn can be used to create computational models of visual search patterns and salient features that mimic the human information gathering process [11], [5].

It has also been shown that the study of eye movement while performing natural tasks can show what actions demand attention and what actions are performed with little conscious involvement. This has been demonstrated when comparing overt attention to actions the subject has previous experience performing and new tasks the subject is unfamiliar with. It has, for example, been shown that if a sequence of actions is known beforehand, the subject’s attention will shift to the next task or action before the current one is completed [5].

Figure 3.2. Composite image showing both bright pupil and Purkinje reflections on corneal surface.
3.2 Applications of Eye and Gaze Tracking

3.2.2 Usability Studies

Eye tracking can be used to evaluate the usability of tools, interfaces, controls and other equipment. Examples of situations where this technique has been used include aircraft, motor vehicles and software graphical user interfaces. Knowledge of the link between eye movement, thought processes and motor functions can determine whether the instruments or controls are intuitively usable and can be understood by the operator. The goal is to create more easily used, less confusing tools and interfaces and avoid problems or potential hazards created by the operator being confused, distracted or inattentive [5].

3.2.3 Advertising and Product Labelling

In advertising, eye tracking has been used to determine the impact of different types of advertisements, as well as trying to establish which features are most important for a successful advertisement. From studies of non-selective visual attention it seems likely that most humans have the same attention triggers, which are important for noticing an advertisement. Properties such as size, sharp contrasts in colour and spatial location have been shown to influence the amount of time spent reading an advertisement, and the amount of information remembered from it. A person is more likely to notice a large advertisement than a smaller one, similarly, an advertisement in colour is more noticeable than the same advertisement in greyscale. It also seems that the placement of an advertisement high on the page makes it more likely to be read and remembered. This has not only affected the layout and placement of advertisements, but also the pricing of ad space in print and electronic publications [5].

In the business of product labelling it is important not only to attract attention, but also to convey relevant aspects of the product. Bojko et al. conducted a study of labelling of pharmaceutical products in 2005 which indicates that a generic layout of product labels makes the identification of a specific product easier and faster if the labels have not been seen before [5].
3.2.4 Human-Computer Interaction

An important and growing field in the application of eye tracking technology is the development of control systems with eye-based interaction. The classic application is the use of gaze as a pointing tool, similar to a mouse pointer. The user’s gaze point can then be used for selection and interaction within the user interface [5].

One of the problems associated with eye-based interaction is the Midas Touch problem, as described by Jacob in 1990. The problem lies in the fact that there is no simple way to differentiate between studying a feature of the interface and wanting to activate it. One solution to this is to create a fixation/dwell-time threshold and activate the feature only after a preset amount of time spent fixating it. Another problem lies in the inherent inaccuracy of an eye tracker as an indicator of visual attention because the spatial extent of the fovea limits actual gaze point accuracy to (approximately) $\pm 1 - 2^\circ$ visual angle and the line of sight can deviate from the optical axis of the eye by several degrees [5]. The uncertainty introduced by the spatial extent of the fovea is illustrated in figure 3.3.

![Figure 3.3. Uncertainty in gaze point introduced by the spatial extent of the fovea.](image-url)

A major application for this type of system is eye typing and accessibility features for the disabled, where eye tracking is used to select letters or words which the subject wishes to communicate. Another is the use of gaze-contingent displays and attentive user interfaces, where the user’s gaze determines the level of lighting, zoom, processing power or bandwidth allocation [5].
Chapter 4

The Shih-Liu and Hennessey-Noureddin-Lawrence Gaze Trackers

In this chapter, a description of two specific methods of gaze tracking will be described. The methods are those presented by Shih and Liu in the 2004 article *A Novel Approach to 3-D Gaze Tracking using Stereo Cameras* [10] and by Hennessey, Noureddin and Lawrence in *A Single Camera Eye-gaze Tracking System with Free Head Motion* [7] in 2007. These methods were selected for use in the implementation of the system described in chapter 5 due to their simplicity and potential for use with simple, low-cost hardware.

4.1 The Shih-Liu Gaze Tracker

This section summarizes the Shih-Liu method of gaze tracking described by Shih and Liu in [10]. The information contained herein is from this source unless otherwise stated.

4.1.1 Background

In 2004 Shih and Liu proposed a gaze tracking system utilizing a schematic eye, multiple light sources, a pinhole camera model and ray tracing. The techniques described were implemented in a stereo camera system and evaluated. A theoretical single camera version was also described and was later implemented by Hennessey et al. [7] (this is described further in section 4.2). The Shih-Liu implementation uses first-order Purkinje reflections combined with refracted pupil reflections and the Le Grand schematic eye. The Shih-Liu method requires two basic assumptions.
1. The cornea has a constant curvature and radius $\rho$.

2. The cornea and aqueous humour can be modeled as a single spherical lens with uniform refractive index corresponding to the simplified Le Grand schematic eye.

4.1.2 Method

The method described by Shih and Liu is based on first computing the location of the corneal center, and then that of the pupil center. These two points are then used to determine the optical axis of the eye and its intersection with the displayed screen (the gaze point). The corneal center is computed using the first-order Purkinje reflections of at least two light sources, as a single light source would only define the corneal location to a curve in space.

Figure 4.1. Auxiliary coordinate system for corneal center estimation.
Corneal Center Estimation

For each light source, an auxiliary coordinate system is defined. These coordinate systems place the origin in the optical center of a camera, and the Purkinje reflection of the light source in the $x$-$y$ plane (see figure 4.1). The origin is denoted $O$ and the position of the light source $Q$. $P_{img}$ is the 2D image coordinates of the Purkinje reflection. Each point in this system can be expressed as its $x$ and $y$ coordinates, so $P_k = (x_k, y_k)$. The coordinate system is defined such that the $x$ axis is parallel to $\vec{OQ}$ and the $x$-$y$ plane contains the vector $\vec{OP}_{img}$. The law of reflection states that the incident angle of the $\vec{QP}_0$ on the corneal sphere is equal to the reflected ray $\vec{OP}_0$ with respect to the surface normal at $P_0$. Since

$$P_0P_1 = \left[ \rho \cos \left( \frac{\pi + \alpha - \beta}{2} \right), \rho \sin \left( \frac{\pi + \alpha - \beta}{2} \right) \right]^T,$$  \hspace{1cm} (4.1)

the $x$-$y$ coordinates of the corneal center $P_1$ can be expressed as

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_0 - \rho \sin \left( \frac{\alpha - \beta}{2} \right) \\ x_0 \tan (\alpha) + \rho \cos \left( \frac{\alpha - \beta}{2} \right) \end{bmatrix}.$$  \hspace{1cm} (4.2)

When using two cameras and four off-axis light sources placed on the $x$ and $y$ axes of the cameras the corneal location can be determined by nonlinear optimization even if the actual radius of the corneal sphere is unknown.

Pupil Center Estimation

In order to establish 3D line of sight, the image coordinates of the refracted pupil as seen by the cameras are used. Shih and Liu introduce another auxiliary coordinate system (illustrated in figure 4.2) where

- the origin is defined by the corneal center described above.
- the $x$ axis is defined as pointing from the corneal center toward the optical center of the camera.
- the $y$ axis is defined so that the $x$-$y$ plane contains the optical axis of the eye.
- the refracted pupil (as seen through the cornea) is called the virtual pupil.
- the projection of this virtual pupil in the image captured by the camera is called the virtual pupil image.

The angle between the optical axis and the $x$ axis is denoted $\varphi$, the radius of the pupil to $r_p$ and the distance between the corneal center and pupil center to $d$. The pupil edge can then be expressed as

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} d \cos (\varphi) - r_p \cos \theta \sin \varphi \\ d \sin (\varphi) - r_p \cos \theta \cos \varphi \\ r_p \sin \theta \end{bmatrix}.$$  \hspace{1cm} (4.3)
The coordinates of the virtual pupil edge ($\mathbf{P}_{vp}$) are described in terms of the pupil coordinates using Snell’s law of refraction as

$$
\mathbf{P}_{vp} = \begin{bmatrix}
p'_x \\
p'_y \\
p'_z
\end{bmatrix} = \begin{bmatrix}
\rho - \frac{n'l}{R \cdot l + n} \\
m \cdot p_y \\
m \cdot p_z
\end{bmatrix}
$$

(4.4)

where $l = \rho - p_x$, $R = |(n' - n)/\rho|$ and $m = n/(R \cdot l + n)$. Here, $n$ is the refractive index of the cornea and aqueous humour and $n' \approx 1$ is the refractive index of air.

The virtual pupil on the corneal surface is assumed to be planar, and the center of this virtual pupil $\mathbf{P}'_c$ can be obtained by allowing $r_p$ to approach zero in equation 4.4. This yields
4.1 The Shih-Liu Gaze Tracker

\[ P'_c = \lim_{r_p \to 0} P_{vp} = \begin{bmatrix} 
\rho - \frac{n'(\rho - d \cos \varphi)}{n + ((n - n')(\rho - d \cos \varphi))/\rho} \\
\frac{dn \sin \varphi}{n + ((n - n')(\rho - d \cos \varphi))/\rho} \\
0 
\end{bmatrix}. \] (4.5)

This point is then found in 3D by averaging the 2D image points of the pupil edge (the edge points of the virtual pupil image) and back-projecting to the corneal surface (see figure 4.3). The estimation of the real pupil center \( P_c \) is then given as

\[ P_c = [d \cos \varphi, d \sin \varphi]^T. \] (4.6)

The pupil center location is then used in combination with the corneal center to establish the optical axis of the eye and the line of sight.

4.1.3 Algorithm Overview

The main parts of the Shih-Liu algorithm are listed in algorithm 1 below.

**Algorithm 1 Overview of the Shih-Liu algorithm**

1: Locate Purkinje reflections in image.
2: Locate pupil edge in image.
3: Estimate location of corneal center from Purkinje reflections.
4: Estimate pupil center from back-projection of average virtual pupil image edges and corneal sphere location.
5: Calculate optical axis of eye from corneal center and pupil center.
6: Calculate intersection of optical axis and screen.

A per-user calibration is also performed in order to reduce errors caused by deviation of the line of sight from the optical axis of the eye.
4.2 The Hennessey-Noureddin-Lawrence Gaze Tracker

This section summarizes the modified Shih-Liu method of gaze tracking described by Hennessey, Noureddin and Lawrence in [7]. The information contained herein is from this source unless otherwise stated.

4.2.1 Background

The Hennessey-Noureddin-Lawrence gaze tracking method utilizes the principles laid out by Shih and Liu with some modifications. By adding the additional assumption that the curvature of the cornea is known, a single-camera setup is sufficient for gaze point determination. The schematic eye used is the Gullstrand schematic eye and the method of pupil center estimation has been modified. The Hennessey-Noureddin-Lawrence method has basic assumptions similar to those set out by Shih and Liu.

1. The cornea has a constant known curvature and radius (this allows for a single-camera setup).
2. The cornea and aqueous humour can be modeled as a single spherical lens with uniform refractive index corresponding to the Gullstrand schematic eye.
3. The angular deviation of the line of sight from the optical axis of the eye is small enough to be disregarded in the gaze point calculation.

4.2.2 Method

The method described by Hennessey et al. is very similar to the Shih-Liu method on which it is based. It works by first computing the corneal center from Purkinje reflections of two light sources in the anterior corneal surface, and then the pupil center from the image coordinates of the virtual pupil image.

Corneal Center Estimation

The auxiliary coordinate system created using the optical center of the camera and the location of the light sources is the same one described in subsection 4.1.2, though the corneal center now lies in the x-z plane (see figure 4.4). To distinguish between values in the world coordinate system and the auxiliary systems, all values in the auxiliary coordinate systems are marked with a \( \hat{\cdot} \). The representation of the corneal center \( \hat{c} \) remains the same, but expressed in 3D with a \( y \) coordinate which is zero. This gives \( \hat{c} \) as

\[
\begin{bmatrix}
\hat{c}_{ix} \\
\hat{c}_{iy} \\
\hat{c}_{iz}
\end{bmatrix}
= \begin{bmatrix}
\hat{g}_{ix} - r \cdot \sin \frac{\hat{\alpha}_i - \hat{\beta}_i}{2} \\
0 \\
\hat{g}_{ix} \cdot \tan \hat{\alpha}_i + r \cdot \cos \frac{\hat{\alpha}_i - \hat{\beta}_i}{2}
\end{bmatrix}
\] (4.7)
where the angles $\hat{\alpha}_i$ and $\hat{\beta}_i$ are

$$\hat{\alpha}_i = \cos^{-1} \left( \frac{-\hat{I}_i \cdot \hat{Q}_i}{\|\hat{I}_i\| \|\hat{Q}_i\|} \right)$$

(4.8)

$$\hat{\beta}_i = \tan^{-1} \left( \frac{\hat{g}_{ix} \cdot \tan \hat{\alpha}_i}{\hat{l}_i - \hat{g}_{ix}} \right)$$

(4.9)

and $\hat{g}_{ix}$ is an unknown parameter corresponding to $x_0$ in equation 4.2. The coordinate transformation for each of the auxiliary coordinate systems with respect to the world coordinate system is designated $R_i$. This gives the estimated corneal center $c_i$ expressed in world coordinates as the product of the auxiliary coordinates for $\hat{c}_i$ and its transformation matrix,

$$c_i = R_i^{-1} \hat{c}_i.$$  

(4.10)

Figure 4.4. Auxiliary coordinate system for corneal center estimation.
Since both estimates of the corneal center correspond to the same point in world space, the additional constraint
\[ c_1 = c_2 \] (4.11)
applies when this is done for two light sources. This is solved numerically by means of a nonlinear least-squares optimization.

**Pupil Center Estimation**

In estimating the pupil center, the Hennessey-Noureddin-Lawrence method is slightly different than the Shih-Liu method. Rather than calculating the center of the virtual pupil image and back-projecting, pairs of opposing edge points are selected from an ellipse fit to the virtual pupil image, separately back-projected to the estimated distance of the pupil edge from the corneal center and then averaged to provide the 3D location of the pupil center. The back-projection \( u_i \) of each virtual pupil image edge point \( k_i \) can be expressed as a line
\[ u_i = k_i + s_i \cdot K_i \] (4.12)
where \( K_i \) is a unit vector defining the direction from \( k_i \) to the optical center of the camera and \( s_i \) is the displacement along this line. The pupil edge points can then be expressed as refracted rays starting in \( u_i \). This yields the expression for the pupil edge points \( \hat{u}_i \)
\[ \hat{u}_i = u_i + w_i \cdot \hat{K}_i \] (4.13)
Here, the rays emanate from the previously computed corneal intersections. The unit vector \( \hat{K}_i \) is the previous direction \( K_i \) refracted in the corneal surface and \( w_i \) is the displacement along this line (see figure 4.5).

![Figure 4.5. Pupil center estimation through back-projection of virtual pupil image edge points.](image)

The optical axis of the eye is established using the estimated corneal center and pupil center locations, and the gaze point \( p \) is given as the intersection between the optical axis and the screen
\[ p = c + t \cdot L \] (4.14)
with \( L \) as a unit vector along the optical axis.
4.2.3 Algorithm Overview

The main parts of the Henessey-Noureddin-Lawrence algorithm are listed in algorithm 2 below. The principal difference between the two methods lies in the pupil back-projection, which in this case is averaged after the actual projection.

Algorithm 2 Overview of the Henessey-Noureddin-Lawrence algorithm

1: Locate Purkinje reflections in image.
2: Locate pupil edge in image.
3: Estimate location of corneal center from Purkinje reflections.
4: Estimate pupil center from average back-projection of virtual pupil image edges and corneal sphere location.
5: Calculate optical axis of eye from corneal center and pupil center.
6: Calculate intersection of optical axis and screen.

A per-user calibration is also performed in order to reduce gaze point errors by bilinear interpolation between corrections at preset calibration points in screen space.
Chapter 5

Implementation

In this chapter a description of the implementation will be given. Section 5.1 describes the hardware used in the implementation, the function of the various components and their places in the system. Section 5.2 contains a description of the gaze tracking algorithm as a whole, as well as its main parts and the principles upon which they function. Examples are also given of the image and tracking data at various stages of processing.

5.1 Hardware

The hardware used in the implementation of the gaze tracking system consists of a camera, an illumination control circuit, a power supply and a desktop computer. The computer used is equipped with an Intel® Core2 Duo E6750 CPU running at 2.66GHz with 2GB of RAM.

5.1.1 Camera

The camera used in the implementation is a Point Grey Flea2 FLG2F-13S2M with a FireWire interface [8]. The camera is equipped with a Sony ICX445 1/3" progressive scan CCD at a resolution of 1288x966 pixels. The camera is fitted with a C-mount objective lens with focal distance 12 mm. In order to reduce interference from ambient visible light, the objective lens is fitted with a Heliopan optical long-pass filter with a cutoff frequency of 780 nm. The camera, objective lens and long-pass filter are shown in figure 5.1.
Figure 5.1. The Point Grey Research Flea2 (FL2G-13S2M-C), objective lens and Heliopan optical long-pass filter (left). Objective lens and filter mounted on camera (right).

5.1.2 Illumination Control

The illumination control is responsible for providing the Purkinje reflections used to determine the corneal center location and the pupil reflection used to find the pupil center. The illumination control triggers two different groups of Agilent Technologies HSDL-4220 High-Performance TS AlGaAs Infrared LED Lamps to provide on- and off-axis illumination in alternating frames. Switching of illumination is done using the *Integration Enable* signal from the GPIO port on the Flea2 camera (described in subsection 5.1.3). This signal is led to an NTE Electronics 4095B Gated J-K Master/Slave Flip-Flop on which the J and K terminals have been wired together, making it function as a toggle flip-flop. The toggled signal and its inverse are then led to the gates of two SGS-Thomson Microelectronics IRF620/STM N-Channel Enhancement Mode Power MOS Transistors which control the larger currents needed to drive the infrared LED lamps. In parallel with each group of infrared LED lamps, an Everlight 363GD GaP LED Lamp is connected. This green LED lamp lights up to indicate when the corresponding group of infrared LED lamps are lit. The wiring of the illumination control is illustrated in figure 5.2. The circuit is powered by a Farnell L30BT Stabilised Power Supply at a voltage varying from 5 to 15V depending on desired brightness. The geometric layout and structure of the illumination control is described in subsection 5.1.4.
5.1.3 Illumination Control/Camera Interface

Flea2 FL2G cameras are equipped with a HiRose HR25-7TR-8SA connector for General Purpose I/O (GPIO) functionality [8], [9]. The connector allows for access to internal signals such as the Integration Enable signal which is active during image integration, or external triggering of the camera. The FL2G models are equipped with an opto-isolated output signal, which can be connected to an external voltage (see figure 5.4). The function of this output defaults to the Integration Enable signal. The layout of the HiRose connector is illustrated in figure 5.3.

In this implementation, the Integration Enable signal is fed into the illumination control described in subsection 5.1.2.

Figure 5.2. Illumination control circuit wiring diagram.
Figure 5.3. Layout of the HiRose HR25-7TR-8SA connector.
1: GPIO0, opto-isolated input
2: GPIO1, opto-isolated open-collector output
3: GPIO2, bi-directional input/output
4: GPIO3, bi-directional input/output
5: GND, ground pin for bi-directional IO, $V_{ext}$, +3.3V
6: GND, ground pin for opto-isolated IO pins
7: $V_{ext}$, external camera power
8: +3.3V, external device power

Figure 5.4. Structure of the opto-isolated GPIO port, Terminal numbers correspond to pins on the HiRose connector in figure 5.3.

5.1.4 Installation and Setup Geometry

The camera and illumination control is mounted on a metal frame which is placed in front of the user. The illumination control is divided into four separate pods, numbered one through four in figure 5.5. The first pod contains the illumination control circuitry, logic and green status LED lamps, the second and third contain the infrared off-axis LED lamps in groups of four and the fourth, which is mounted coaxially on the objective lens contains the on-axis infrared LED lamps.
Figure 5.5. Structure and geometry of the hardware setup. Schematic image with labels (top), photograph of the actual hardware (bottom).
5.2 Software

The gaze tracking algorithm consists of a number of major steps, which must be performed in order to locate the gaze point. The initial steps involve extracting information from the captured images in order to obtain the image locations of the tracked features. The later steps consist of calculations based on these image locations, which yield the gaze point information. The steps are listed in algorithm 3.

Algorithm 3 Overview of the implemented gaze tracking algorithm
1: Capture image/read image from video file
2: Classify as on-axis or off-axis image (see subsection 5.2.1)
3: Detect pupil and set ROI (see subsection 5.2.2)
4: Get pupil edge points, parameterize and find center (see subsection 5.2.2)
5: Get glint positions on sensor (see subsection 5.2.3)
6: Calculate corneal center (see subsection 5.2.4)
7: Calculate pupil center (see subsection 5.2.5)
8: Calculate screen plane intersection of optical axis (see subsection 5.2.6)

5.2.1 Frame Classification and Segmentation

In order to correctly process a new frame, it must first be classified as an on-axis (bright pupil) or off-axis (corneal Purkinje reflection) frame. This is accomplished by measuring the average intensity of each frame and comparing it to the average intensity of all frames. The on-axis frames generally have a much higher average intensity than the off-axis frames due to the coaxial placement of the light source and can therefore be distinguished from the off-axis frames with relative ease. Naturally the same principle also applies to the off-axis frames. Before the actual gaze tracking algorithm can be run, one on-axis and one off-axis frame must be identified.

The frames must also be divided such, that each image contains only one eye. This is because the gaze tracking algorithm in its current form only tracks one eye at a time. The segmentation is achieved by creating a binary image of the bright pupils in a difference image between an on-axis and an off-axis frame by thresholding. The image is then divided along a line placed vertically between the bright areas.

5.2.2 Pupil Location Algorithm

The pupil location algorithm is called each time a new on-axis frame is processed. The Region of Interest containing the pupil is identified by finding the maximum intensity in the on-axis image. A Region of Interest around this point is then extracted from both the on-axis (pupil) image and the off-axis (cornea) image. Through a series of thresholding and filtering operations, a difference image of the ROI in the pupil and cornea images is produced and thresholded to find pixels
belonging to the pupil. A selection is then made among these pixels and those most likely to belong to the pupil perimeter are extracted. The coordinates of these pixels are parameterized to an ellipse using the method described by Halíř and Flusser [6]. The center of this ellipse is considered the virtual pupil image center. The steps are listed in algorithm 4. Examples of images at the various stages of the algorithm are shown in figure 5.6.

Algorithm 4 Steps of the pupil location algorithm

1: Find intensity maximum in pupil image.
2: Cut ROI from pupil image.
3: Cut ROI from cornea image.
4: Perform thresholding of corneal image to minimize effect of glints on the pupil contour.
5: Create difference image of regions cut from pupil and cornea images.
6: Perform 2D Wiener filtering to reduce noise.
7: Threshold to find possible pupil pixels.
8: Perform binary closing to reduce noise.
9: Get perimeter points of pupil area.
10: Remove points inside the probable ellipse contour.
11: Parameterize to ellipse and find center.
12: Convert image coordinates to sensor coordinates.
5.2.3 Glint Location Algorithm

The glint location algorithm is called each time a new off-axis frame is processed. The ROI determined to contain the eye is cut from the off-axis image and a 2D Wiener filter is applied. The image is then thresholded to contain only pixels with an intensity value at least twice the mean intensity of the filtered image. After this, non-maximum-suppression filtering is performed in 3x3 pixel patches so that only local maxima remain. A circular mask is then applied to eliminate all pixels in the corners of the ROI. Areas of 5x5 pixels around the remaining maxima are then extracted and the peaks found using a second degree polynomial approximation. The glint locations are then transformed into world coordinates to be used in the corneal center estimation. The steps are listed in algorithm 5 below. Figure 5.7 contains examples of the image at different stages of glint location.
Algorithm 5 Steps of the glint location algorithm

1: Cut ROI from image.
2: Apply 2D Wiener filter to reduce noise.
3: Threshold to remove low-intensity pixels.
4: Perform non-maximum-suppression filtering.
5: Apply mask to remove corners.
6: Extract patches around remaining maxima.
7: Estimate peaks.
8: Convert to world coordinates.

5.2.4 Calculation of Corneal Center

The estimation of the corneal center location is carried out in accordance with the method described in subsection 4.2.2. Equations 4.7, 4.10 and 4.11 are solved numerically using a non-linear least-squares approximation. This step yields one of the points defining the optical axis of the eye and is also used in the calculation of the pupil center (see subsection 5.2.5).

5.2.5 Calculation of Pupil Center

In order to find the location of the pupil center in world space the virtual pupil image center (as computed in subsection 5.2.2) is back-projected to the estimated location of the corneal surface. The back-projection ray is then refracted in the corneal surface. The intersection of the refracted ray and a sphere centered on the corneal center with a radius equal to the estimated distance between the corneal and pupil centers is found. This intersection is considered the pupil center location. This step yields the second point needed to determine the optical axis of the eye. This method is the same as the method described in subsection 4.1.2.

5.2.6 Calculation of Gaze Point

After both corneal center and pupil center have been estimated, the intersection of a line through these points and a plane containing the screen is calculated. This intersection is considered the gaze point.

5.2.7 Calibration of Gaze Point

In order to reduce errors in the gaze points calculated, a calibration pattern is created and images of the eyes gazing at points in this calibration pattern collected. The deviation of the mean gaze point from the true value at each of these points is then used to adjust the calculated gaze points. The correction is calculated for new gaze points by means of a bilinear interpolation of the corrections needed at nearby calibration points. The calibration pattern is shown in figure 5.8.
Figure 5.8. Calibration pattern in screen coordinates. All axes in [mm].
Chapter 6

Evaluation

In this chapter, the evaluation criteria and procedures used to test the gaze tracking solution will be described. The first section will describe evaluation of the hardware used and the second evaluation of the gaze tracking software on both synthetic and real images. The results obtained using these evaluation procedures can be found in chapter 7.

6.1 Hardware Evaluation

The hardware, composed primarily of the illumination control and the camera, was evaluated in order to determine whether it is capable of performing well enough to allow capture of images suitable for gaze tracking. The evaluation procedure for the camera is described in subsection 6.1.1 and the evaluation procedure for the illumination control in subsection 6.1.2.

6.1.1 Camera Evaluation

The evaluation of the camera simply consisted of assessing whether the camera could acquire images of a sufficient quality. The images had to be clear and the necessary features for the gaze tracking visible when properly lit. In order to assess whether this was possible, a set of images was captured and examined for these features. If the features were found, the camera was considered to be performing as required. If performance was not satisfactory, adjustments to the optics were made and the procedure was repeated.
6.1.2 Illumination Control Evaluation

The function of the illumination control is to provide on-axis and off-axis illumination in alternating frames. In order to assess whether it is functioning properly measurements of the input to the illumination control circuit and the resulting output were measured. Three series of measurements were done, two comparing the input to each of the two output signals, and one comparing the output signals to each other. If it was found that the outputs were toggled correctly, the illumination control circuit was considered to be operating properly. The intensity of the lighting must also be high enough to produce clearly visible reflections in the eyes. The camera evaluation images described in subsection 6.1.1 were examined for these reflections. If they were found, the lighting intensity was considered adequate. If performance was not satisfactory, the voltage supplied to the circuit was adjusted and the procedure was repeated.
6.2 Software Evaluation

6.2.1 Generation of Synthetic Image Data

Generation of synthetic image data was done with a simple raytracing program created specifically for this purpose. The program creates synthetic images of the corneal and pupil reflections given a corneal center location and a gaze point. The setup geometry used in the simulation is the same as that used in the gaze tracking algorithm. Examples of relevant regions from the two kinds of images produced are displayed in figure 6.1. The raytracer also applies a low-pass filter to the images in order to make them more similar in appearance to the real images the camera provides.

![Synthetic and Real Images](image)

*Figure 6.1.* Examples of synthetic images (top row) and real images (bottom row). Corneal reflections (left), pupil reflection (right).

6.2.2 Software Evaluation on Synthetic Image Data

In order to evaluate the gaze tracking algorithm two sets of 441 synthetic image pairs (each consisting of an on-axis and an off-axis illuminated image) were generated. The first set models a stationary cornea and a sweeping gaze point (see figure 6.2). The second set models a mobile cornea with the same sweeping gaze points (see figure 6.3). Along with both sets of images, ground truth data is also generated. Each of these image pairs were then fed into the gaze point algorithm and the result compared to the true values. The gaze point function allows the input of true values for the corneal center and gaze point locations. Thus, errors arising from the corneal center location or the pupil center location can be calculated independently. Errors resulting from the quality of the images themselves cannot be seen in this way. Due to numerical errors and quantization effects in the images, and the inherent inaccuracy of the methods used, it is not possible to obtain a perfect result, even with synthetic images. Performance on synthetic images is described in subsection 7.1.3.
In order to illustrate and estimate the potential errors arising from a miscalculation of the pupil center (which is the most crucial factor in gaze point determination as the short distance between corneal center and pupil center leads to large angular deviations when the pupil center is miscalculated), a simplified model of the connection between miscalculation of pupil center and gaze point error was devised. This model is illustrated in figure 6.4. Approximate numerical values computed from this model are listed in table 6.1.

<table>
<thead>
<tr>
<th>$\epsilon_{pc}$ [px]</th>
<th>$\alpha_{pc}$ [$^\circ$]</th>
<th>$d_{pc}$ [mm]</th>
<th>$\beta_{gp}$ [$^\circ$]</th>
<th>$d_{gp}$ [mm]</th>
<th>$d_{gp}$ [px]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.00448</td>
<td>0.04688</td>
<td>0.53716</td>
<td>5.9</td>
<td>20</td>
</tr>
<tr>
<td>0.5</td>
<td>0.00895</td>
<td>0.09375</td>
<td>1.0749</td>
<td>11.8</td>
<td>42</td>
</tr>
<tr>
<td>1</td>
<td>0.01790</td>
<td>0.18750</td>
<td>2.1491</td>
<td>23.6</td>
<td>101</td>
</tr>
<tr>
<td>2</td>
<td>0.03581</td>
<td>0.37540</td>
<td>4.3012</td>
<td>47.3</td>
<td>201</td>
</tr>
<tr>
<td>3</td>
<td>0.05371</td>
<td>0.56250</td>
<td>6.4594</td>
<td>71.1</td>
<td>303</td>
</tr>
<tr>
<td>4</td>
<td>0.07162</td>
<td>0.75000</td>
<td>8.6269</td>
<td>94.9</td>
<td>405</td>
</tr>
<tr>
<td>5</td>
<td>0.08952</td>
<td>0.93750</td>
<td>10.807</td>
<td>118.9</td>
<td>507</td>
</tr>
</tbody>
</table>

Table 6.1. Approximate errors arising from pupil center miscalculation. In this table $\epsilon_{pc}$ denotes the pixel error in the estimation of the virtual pupil image center and $\alpha_{pc}$ the resulting angular deviation of the virtual pupil center as seen from the origin (camera). $d_{pc}$ is an approximation of the resulting spatial error in the pupil center (refraction not taken into account) and $\beta_{gp}$ the angular deviation of the pupil center as seen from the corneal center. The error in gaze point is denoted $d_{gp}$. The numerical values shown are approximation based on the simplified model shown in figure 6.4.
Figure 6.2. Cornea (blue) and gaze point (magenta) locations in sample set 1. The cornea is stationary at $(0, 0, 600)$ [mm] in the world coordinate system (corresponding to a position 600mm directly in front of the camera and along its optical axis). The gaze point uniformly sweeps an area with corners at $(-50, 50, 0)$, $(50, 50, 0)$, $(50, -50, 0)$, $(-50, -50, 0)$ [mm] in the screen coordinate system, centered on the center of the screen with a sampling distance of 5mm.
Figure 6.3. Cornea (blue) and gaze point (magenta) locations in sample set 2. The cornea sweeps an area with corners at \((-30, 30, 600), (30, 30, 600), (30, -30, 600), (-30, -30, 600) [mm]\) in the world coordinate system (corresponding to a plane 600mm directly in front of the camera and perpendicular to its optical axis). The gaze point uniformly sweeps an area with corners at \((-50, 50, 0), (50, 50, 0), (50, -50, 0), (-50, -50, 0) [mm]\) in the screen coordinate system, centered on the center of the screen.
Figure 6.4. Simplified pupil center error estimation.
6.3 System Evaluation

In order to evaluate the performance of the entire solution, a calibration was performed as described in subsection 5.2.6. A reconstruction of the calibration pattern was performed using the calculated corrections. Then, video sequences were captured using the camera while the user was looking at points in an evaluation pattern located within the calibration pattern. These gaze points were reconstructed using the calibration pattern and a varying number of frames for gaze point estimation. Also, a second calibration and evaluation set was captured, where half of the points in the evaluation set were located outside the calibration pattern. These gaze points were then reconstructed in the same way.

6.3.1 Experimental Setup

The setup used for evaluation is the same as that described in subsection 5.1.4. A manual measurement of the setup geometry was made and input into the gaze tracking software. Camera parameters were taken from the camera documentation (see [8]) and, where applicable, verified using the Camera Calibration Toolbox [4].

6.3.2 Data Acquisition

The image data which was to be processed was captured using the Point Grey FlyCapture application provided with the camera drivers. Video sequences were captured and stored uncompressed on disk for later processing. The video sequences were then read into MATLAB using the `aviread` function.

When video data was captured for the calibration pattern, the user was seated at a "normal" distance (approximately 600 mm) directly in front of the monitor and video for all points was captured in sequence. When video data was captured for the evaluation pattern, data for a single point was captured. The user then left the room, returned and sat down again at the workstation. This was done in order to introduce typical variations in head and eye positions resulting from varying posture and distance. This evaluation procedure was also used in [7].
Chapter 7

Results and Discussion

In this chapter the results of the evaluation procedures described in chapter 6 will be presented. The results will, where applicable, also be compared to results from two other gaze tracking solutions (see subsections 7.2.4 and 7.2.5). A discussion of the results obtained can be found in section 7.2.

7.1 Results

In this section the results of the evaluation procedures described in chapter 6 will be presented. Evaluation of both hardware and software performance will be presented. Comparisons to Hennessey et al. gaze tracker [7] and the Bäck [3] gaze tracker will also be made.

7.1.1 Camera Performance

An on-axis and an off-axis illuminated image were captured using the camera. These images were then manually examined for the features necessary for the gaze tracking algorithm. Both images were found to contain the necessary features. The relevant regions of the images are shown in figure 7.1.

7.1.2 Illumination Control Performance

The illumination control was connected to an oscilloscope and the outputs to the three lighting pods was measured. The results of this measurement can be seen in figure 7.2. The toggling of the lighting was found to be performed correctly by the illumination control.

The images captured and used for camera evaluation were found to contain all the necessary features for gaze point estimation of both eyes (see figure 7.1). The reflections were clearly visible and the lighting intensity is therefore considered adequate.
Figure 7.1. Camera performance. On-axis illuminated images with bright pupil reflections (top), off-axis illuminated images with corneal reflections (bottom).

Figure 7.2. Strobe control performance. Integration enable signal from camera and output to on-axis lighting (left), Integration enable signal from camera and output to off-axis lighting (center), both output signals (right).
7.1 Results

7.1.3 Performance on Synthetic Data

The evaluation of the algorithm was carried out according to the evaluation procedure described in subsection 6.2.2. The two image series were generated and given as input to the gaze tracking software. The results on image set 1 (see figure 6.2) are shown in table 7.1 and figure 7.3. The results on image set 2 (see figure 6.2) are shown in table 7.2 and figure 7.4. The values listed are average values over these image sets.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Mean Spatial error, (standard dev.) [mm]</th>
<th>$X$ [mm]</th>
<th>$Y$ [mm]</th>
<th>$Z$ [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corneal center</td>
<td>0.5130 ($3 \times 10^{-15}$)</td>
<td>0.0031</td>
<td>0.0092</td>
<td>0.5130</td>
</tr>
<tr>
<td>Pupil center</td>
<td>0.5182 (0.0195)</td>
<td>0.0402</td>
<td>0.0482</td>
<td>0.5126</td>
</tr>
<tr>
<td>Gaze point</td>
<td>9.2533 (4.3229)</td>
<td>5.2054</td>
<td>6.7391</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7.1. Errors on synthetic images, set 1 (stationary corneal center). Mean angular error was 0.88018°. $Z$ errors in gaze point are not possible due to the line-plane intersection calculation (the gaze point will always lie on the screen plane).

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Mean Spatial error, (standard dev.) [mm]</th>
<th>$X$ [mm]</th>
<th>$Y$ [mm]</th>
<th>$Z$ [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corneal center</td>
<td>3.4019 (3.4801)</td>
<td>0.0819</td>
<td>0.0850</td>
<td>3.3989</td>
</tr>
<tr>
<td>Pupil center</td>
<td>3.4084 (3.4768)</td>
<td>0.1072</td>
<td>0.1114</td>
<td>3.3997</td>
</tr>
<tr>
<td>Gaze point</td>
<td>10.2276 (4.8718)</td>
<td>5.7826</td>
<td>7.2137</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7.2. Errors on synthetic images, set 2 (mobile corneal center). Mean angular error was 0.97578°. $Z$ errors in gaze point are not possible due to the line-plane intersection calculation (the gaze point will always lie on the screen plane).
Figure 7.3. Performance on synthetic images, set 1 (stationary corneal center). Gaze point locations in screen coordinates (top), pupil center locations (center), corneal centers (bottom). True values correspond to blue circles, calculated values to red crosses. All axes in [mm].
Figure 7.4. Performance on synthetic images, set 2 (mobile corneal center). Gaze point locations in screen coordinates (top), pupil center locations (center), corneal centers (bottom). True values correspond to blue circles, calculated values to red crosses. All axes in [mm].
7.1.4 Performance on Real Data

Performance on real data is evaluated in two ways. First, a calibration is performed and verified. Then, video is captured while the user gazes at an evaluation pattern. The full gaze tracking and reconstruction is then performed on this data.

Calibration

The calibration described in subsection 5.2.7 was performed using the mean gaze point of both eyes averaged over 50 frames to estimate the location of each calibration point. The calibration pattern was then reconstructed by interpolation of the correction values calculated at the calibration points. The results of the reconstruction of the calibration pattern are shown in figure 7.5. The mean reconstruction error was 1.22 mm for the calibration points. Interpolated maps of the corrections calculated within the gaze tracker output at the calibration points are shown in figure 7.6.

Figure 7.5. Reconstruction of calibration pattern. Mean spatial error was 1.22 mm. True calibration points (black circled crosses), gaze tracker output (red crosses), reconstructed calibration points (magenta squares). All axes in [mm].

Figure 7.6. Corrections in the $x$ direction (left) and in the $y$ direction (right). All axes in [mm].
Gaze Point Estimation

Evaluation of gaze point estimation was first performed on a set of four points located within the bounds of the calibration pattern. Video sequences of 95 frames were captured while the eyes were fixating on each of the evaluation points. The resulting gaze point data was then reconstructed using the interpolated corrections from the calibration pattern. Depending on how many frames were used to estimate the gaze point, varying levels of accuracy were achieved. The mean errors when using an increasing number of frames is shown in figure 7.7. The evaluation pattern and its reconstruction using 95 frames for each evaluation point are shown in figure 7.8.

![Figure 7.7. Mean error on evaluation set vs. frames used in gaze point estimation for points within the calibration pattern.](image)

A calibration and evaluation on a second set of points was also performed. The calibration pattern was removed from the monitor and reattached, and a new calibration was performed. A set of evaluation points located both inside and outside the calibration pattern was then processed in the same way as described above. The reconstruction of the second evaluation set, consisting of points both inside the calibration pattern and outside it is shown in figure 7.10. Errors for points inside and outside the calibration pattern using an increasing number of frames for estimation are shown in figure 7.9.
Figure 7.8. Reconstruction of first evaluation pattern using 95 frames per evaluation point. Mean spatial error was 2.49 mm. Calibration points (black circled crosses), true evaluation points (blue squares), gaze tracker output (red crosses), reconstructed evaluation points (magenta diamonds). All axes in [mm]. Evaluation points were selected to lie within the calibration pattern, but to not coincide with any calibration point.
Figure 7.9. Mean error on second evaluation set vs. frames used in gaze point estimation for points inside (top) and outside (bottom) of the calibration pattern.
Figure 7.10. Reconstruction of second evaluation pattern using 95 frames per evaluation point. Mean spatial error for points inside the calibration pattern was 2.85 mm. Mean spatial error for points outside the calibration pattern was 5.50 mm. Calibration points (black circled crosses), true evaluation points (blue squares), gaze tracker output (red crosses), reconstructed evaluation points (magenta diamonds). All axes in [mm]. Evaluation points were selected to lie both inside and outside the calibration pattern, but to not coincide with any calibration point.
7.2 Discussion

In this section the results of the evaluation will be discussed. The first subsection deals with the results on synthetic image data and the second with results on real data. The third subsection discusses the results in general.

7.2.1 Synthetic Data

The results on synthetic data serve to verify that the software part of the solution is of sound construction. Since errors on the synthetic data set can arise only from errors or inaccuracies in the algorithm and numerical errors in the synthetic image data, this evaluation method should be appropriate for testing the software.

The results obtained from the synthetic image sets indicate that the algorithm is in fact performing its task satisfactorily. The angular and spatial errors which are present in the results are most likely due to numerical errors and quantization effects in the synthetic image. Moreover, the method used to estimate the pupil center location is likely to have an inherent inaccuracy due to the fact that it uses a more simplified model of the eye geometry than is used in the generation of the synthetic images. Uncalibrated results on synthetic images should therefore give a best-case estimate of the accuracy of the uncalibrated system based on the assumptions that:

- the camera is a perfect pinhole camera
- the light sources are ideal point light sources
- the human eye corresponds exactly to the Gullstrand schematic eye
- the manually measured setup geometry corresponds exactly to the actual setup geometry.

The results indicate that the software is capable of calculating the gaze point of a single eye with an accuracy of around $1^\circ$.

7.2.2 Real Data

The results on real data before calibration indicate that the above mentioned assumptions are in fact incorrect when working with real images. The inaccuracy inherent in the algorithm is combined with the deviations from the camera, eye and geometry models (as well as the simplified optics and algebra applied to these) to produce a much larger error in the uncalibrated state. However, after calibrating on real video sequences, most of these errors can be eliminated and the resulting error is comparable to the synthetic image results (it is in fact even lower at most of the evaluation points). For reliable gaze point data, more frames are needed than in the case of synthetic images (typically, one pair of synthetic images is needed for a reliable gaze point estimation, whereas many frame pairs are needed when using real images as seen in figure 7.7). Though this means that the camera must capture more frames to provide reliable gaze point estimates, this amounts
Results and Discussion

to fairly short capture times in reality as the camera is capable of acquiring images at 30 frames per second. On the points in the second evaluation set which were located outside the calibration pattern, a much larger error was present. This was not unexpected, but illustrates one of the difficulties arising from using this method of correcting the gaze point estimation.

Additionally, it was found that large angular deviations of the pupil centers from the optical axis of the camera (that is, when the gaze point falls far away from the origin at the edges of the screen) caused reflections in the pupil to become less visible, thereby making reliable measurements more difficult. The same problem was present in the corneal reflections in the off-axis image, which became faint when the user moved as far as possible to either side of the image. These effects are likely due to the fact that the LED lamps used for illumination have a fairly narrow illumination angle (17 – 30° according to the manufacturer).

7.2.3 General

In total, the gaze tracking solution seems to perform well on both synthetic and real image data, and may be suitable for applications where real-time gaze point determination is not required and user movement and field of view are limited.

7.2.4 Comparison to the Original Hennessey-Noureddin-Lawrence Gaze Tracker

The implemented gaze tracker is very similar to the one constructed by Hennessey et al. since it is based on this construction. Although the operating principles of the original tracker are described in [7], many implementational details are not described. It is, however, a reasonable guess that the implementations are similar with regard to both hardware and software components. In evaluation, Hennessey et al. used a similar procedure to the one used to test the implemented system, but used only a single pair of frames for each gaze point estimation. Accuracy in the calibrated state was determined to be 0.46° to 0.90° at a range of 750 mm for the original tracker, whereas the implemented system has an accuracy of approximately 1.5° at 600 mm when using one pair of frames (estimated using the calculated distance to the user and the spatial deviation of the gaze point). The reason for the higher accuracy of the original tracker may be the result of a more accurate setup geometry, more accurate feature detection or a better calibration procedure.

7.2.5 Comparison to the Bäck Gaze Tracker

The gaze tracking solution implemented by Bäck [3] is a passive, image-based gaze tracking solution. The system uses a simple web camera to capture images of the user. Facial features are then extracted by matching colour and intensity values in the face of the user. Gaze point determination is achieved using sets of training images with extracted features combined with true gaze point information. The
gaze point is then approximated using a neural net and the error back-propagation (EBP) learning paradigm.

In comparing the gaze tracking solution described in this report to Bäck’s solution, several factors must be considered. First, the Bäck system is completely passive in terms of lighting and does not rely on any hardware besides the camera and computer it is connected to. It also extracts features using colours and intensities, making it sensitive to lighting changes, skin colour and background. It also does not explicitly calculate the gaze point, but learns it which requires vast amounts of training data.

The system described in this report requires its own illumination control to function, but through this is also less sensitive to outside changes in ambient lighting and shadows. It does not use any colour information to detect and classify the tracked features. The features themselves are based on reflections of the light sources in the surfaces of the eye, and can therefore be expected to vary less between users and head positions.

The accuracy of the Bäck gaze tracker is between two and four degrees, corresponding to an average spatial error of 21-42 mm at a range of 600 mm on new images. This was achieved when training the neural net with a set of 500 training images, with varying gaze points. In contrast, the implemented system does not have any learning capability, and the number of frames necessary for a good gaze point estimation is roughly the same for each new point. However, when averaging gaze points over even a few frames, accuracy is substantially better than with the Bäck gaze tracker (approximately 2.5-15 mm at a range of about 600 mm, corresponding to an angular error of $0.24^\circ$ to $1.5^\circ$ after calibration, depending on how many frames are used in estimation).
Chapter 8

Conclusions and Future Work

In this chapter, the main conclusions drawn will be presented. Suggestion of future improvements to the solution implemented will also be proposed.

8.1 Conclusions

The broadly stated aim of this project was to implement a simple gaze tracking solution with reasonable performance. In most aspects, this has been achieved. The implemented system has proven itself on both synthetic and real image data, and has an accuracy comparable to similar non-commercial solutions. Gaze tracking itself is not a simple problem to deal with, and therefore most solutions require many constraints on the problem to produce workable solutions. Considering what resources went into the implementation of this system, the cost is low when compared to commercial solutions. The performance and accuracy of the solution does not compare to state-of-the-art solutions in any way, mostly due to the time consuming gathering of a large number of frames needed to obtain a stable gaze point estimate, and the slow execution of the calculations. This is partly due to implementational choices such as the decision to use MATLAB (which alone rules out real-time performance, and has compatibility issues with the chosen camera) and inefficient implementation of image processing functions. The best accuracies achieved are also sub-par for commercial solutions, although single-camera solutions are uncommon in most high-level applications.

Despite this, a working gaze tracking solution has in fact been created, and will be made available for use within the department.

8.2 Future Work

Many improvements upon this solution can be made, given time. Some major potential improvements include:
• integration of video capture into gaze tracking software
• optimization of image processing functions
• implementation of real-time gaze tracking software
• automatic calibration of geometry
• automatic calibration of gaze point

The hardware used could also be fitted with its own power supply, rather than relying on an external one as is presently the case.

Also, applications such as the empirical evaluation of feature saliency measures (which was one of the original ideas for this project) or the implementation of a HCI system based on this gaze tracker are possible future developments.
Bibliography


Appendix A

List of Parameters

This appendix contains a list of the parameters used in gaze tracking and simulation. The parameters are initialized in the file `load_parameters`. The functions which require these parameters will load this file when needed.
Table A.1. List of parameters used for simulation and gaze point estimation

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>phi</td>
<td>Angle of camera from horizontal plane (degrees)</td>
</tr>
<tr>
<td>screen_T</td>
<td>Matrix describing translation from camera to screen</td>
</tr>
<tr>
<td>screen_R</td>
<td>Matrix describing rotation from camera to screen</td>
</tr>
<tr>
<td>screen_matrix</td>
<td>Total transformation matrix, product of the above</td>
</tr>
<tr>
<td>COS</td>
<td>Center of screen in world coordinates</td>
</tr>
<tr>
<td>screen_normal</td>
<td>Screen normal in world coordinates</td>
</tr>
<tr>
<td>q1</td>
<td>Position of off-axis light source 1</td>
</tr>
<tr>
<td>q2</td>
<td>Position of off-axis light source 2</td>
</tr>
<tr>
<td>f</td>
<td>Focal distance of camera</td>
</tr>
<tr>
<td>xres</td>
<td>Number of image columns</td>
</tr>
<tr>
<td>yres</td>
<td>Number of image rows</td>
</tr>
<tr>
<td>pixel_size_x</td>
<td>Width of a pixel</td>
</tr>
<tr>
<td>pixel_size_y</td>
<td>Height of a pixel</td>
</tr>
<tr>
<td>r</td>
<td>Corneal radius</td>
</tr>
<tr>
<td>rd</td>
<td>Distance from corneal center to pupil center</td>
</tr>
<tr>
<td>n_cornea</td>
<td>Refractive index of cornea and aqueous humour</td>
</tr>
<tr>
<td>n_air</td>
<td>Refractive index of air</td>
</tr>
<tr>
<td>c_sim</td>
<td>Simulated corneal center (used in simulations)</td>
</tr>
<tr>
<td>rp_sim</td>
<td>Simulated pupil radius (used in simulations)</td>
</tr>
<tr>
<td>POG_sim</td>
<td>Simulated gaze point (used in simulations)</td>
</tr>
<tr>
<td>pc_sim</td>
<td>Simulated pupil center (used in simulations)</td>
</tr>
<tr>
<td>ROI_size</td>
<td>Simulated pupil center (used in simulations)</td>
</tr>
<tr>
<td>cornea_im_path</td>
<td>Default save path for rendered images of cornea</td>
</tr>
<tr>
<td>pupil_im_path</td>
<td>Default save path for rendered images of pupil</td>
</tr>
<tr>
<td>supersample</td>
<td>Supersampling factor for raytracing of images</td>
</tr>
<tr>
<td>calibration_POGs</td>
<td>Gaze points for calibration</td>
</tr>
</tbody>
</table>
Appendix B

Function Reference

The MATLAB functions needed to run the simulations and gaze tracking algorithm are listed and briefly described below. The functions are divided by category depending on their use in the solution, and where they can be located within the directory structure.

B.1 Gaze Tracking Functions

These functions are directly connected to the gaze tracking algorithm. They can be found in the root directory of the implementation.

- **center_of_cornea_PE**: Objective function for corneal center estimation.
- **find_center_of_cornea_PE**: Function for running the corneal center estimation.
- **findPOG**: Function which calculates gaze point given a pair of images consisting of an on-axis and an off-axis frame.
- **get_video_gaze**: Function which performs gaze tracking on a video sequence given a file name.
- **load_parameters**: Setup script which loads parameters, see table A.1.

B.2 Vector Tools

These functions are used for various vector operations and algebra. They can be found in the `vector_tools` directory.

- **crop_vector**: Function which crops a vector given an input vector and a binary mask.
• **image2world_coords**: Function which converts image coordinates to world coordinates on the sensor.

• **interpolate_correction**: Function which performs bilinear interpolation of gaze point corrections given a measured point and four calibration points with corresponding correction values.

• **intersectplane**: Function which calculates the intersection of a ray and a plane.

• **intersectsphere**: Function which calculates the intersection of a ray and a sphere.

• **refract**: Function which calculates the refraction of a ray given a surface normal and refractive indices.

• **screen2world**: Function which transforms screen coordinates into world coordinates.

• **vecnorm**: Function which normalizes and calculates the original length of a vector.

• **world2screen**: Function which transforms world coordinates into screen coordinates.

### B.3 Calibration Tools

These functions are used for calibration of the gaze point. They can be found in the `calibration_tools` directory.

• **correct_computed_POG**: Function which applies corrections to a gaze point calculated using synthetic data.

• **correct_video_gp**: Function which applies correction to a gaze point calculated using video data.

• **generate_calibration_images**: Function which generates synthetic calibration images using the raytracing functions.

• **get_gp_calfac**: Function which calculates the gaze point corrections using the calibration video sequences.

### B.4 Raytracing Tools

These functions are used for generation of synthetic images. They can be found in the `raytrace_tools` directory.

• **generate_image_pair**: Function which generates a pair of images consisting of an on-axis and an off-axis illuminated image.
B.5 Image Tools

- **limited_raytrace_cornea**: Function which, given a probable image location of the eye, raytraces a region of the off-axis image.

- **limited_raytrace_pupil**: Function which, given a probable image location of the eye, raytraces a region of the on-axis image.

- **raytrace_cornea**: Function which raytraces the entire off-axis image.

- **raytrace_pupil**: Function which raytraces the entire on-axis image.

B.5 Image Tools

These functions are used for image manipulation and analysis. They can be found in the `image_tools` directory.

- **crop_image**: Function which crops an image given a binary mask.

- **findpupiledge**: Function which searches for the pupil contour.

- **generate_circle_mask**: Function which generates a circular binary mask.

- **get_ellipse_center**: Function which, given edge pixels, fits an ellipse and returns its center point.

- **get_glint_positions**: Function which searches for the corneal reflections in the off-axis image.

- **get_nonzero_coordinates**: Function which returns indices of all non-zero elements in an image.

- **maxcoords**: Function which returns the indices of the intensity maximum in an image.

- **pick_points**: Function which, given edge pixels, fits an ellipse and calculates its center and vertices.

- **reconstruct_image**: Function which uses normalized convolution to reconstruct a sparsely sampled image.

- **separate_eyes**: Function which divides a pair of video frames along a vertical line to separate the images of the left and right eye.