

Arto Meriläinen

Adapting a Gaze Tracking System to Mobile Environments

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of
Science in Technology.

Espoo 11.5.2012

Thesis supervisor:

Prof. Mikko Sams

Thesis instructor:

M.Sc. (Tech.) Sharman Jagadeesan

Author: Arto Meriläinen

Title: Adapting a Gaze Tracking System to Mobile Environments

Date: 11.5.2012

Language: English

Number of pages:10+65

Department of Biomedical Engineering and Computational Science

Professorship: Cognitive Science

Code: S-114

Supervisor: Prof. Mikko Sams

Instructor: M.Sc. (Tech.) Sharman Jagadeesan

Gaze tracking has traditionally been performed in controlled laboratory environments. The experiment setups have commonly been limited to study only computer-human interaction. Recently, the need to perform experiments in natural environments has emerged in different areas of science.

This thesis overviews the structure of a mobile gaze tracking system that was developed in the Ganzheit project. The goal of the gaze tracking system is to offer an open alternative for commercial systems. The developed gaze tracker utilises model-based gaze tracking. The approach is accurate and robust against movements of the head.

In order to utilise model-based gaze tracking, it is vital to identify the pupil and corneal reflections from an eye image. The constructive part of the thesis focuses on developing a method for identifying the features from the eye image. The developed method is tested with experimental data. The results show that the method for finding the pupil is accurate and robust against differences in facial features and changing lighting conditions. The goodness of recognising the corneal reflections varies between test subjects.

The mobile gaze tracking system is experimented in an ordinary office room. The results indicate that the developed method works adequately.

Keywords: mobile, gaze tracking, real time, machine vision

Tekijä: Arto Meriläinen

Työn nimi: Katseenseurantajärjestelmän sovittaminen mobiiliympäristöihin

Päivämäärä: 11.5.2012

Kieli: Englanti

Sivumäärä:10+65

Lääketieteellisen tekniikan ja laskennallisen tieteen laitos

Professuuri: Kognitiivinen tiede

Koodi: S-114

Valvoja: Prof. Mikko Sams

Ohjaaja: DI Sharman Jagadeesan

Katseenseurantajärjestelmiä on käytetty perinteisesti laboratorio-olosuhteissa. Mahdolliset koeasetelmat ovat rajoittuneet koehenkilön ja tietokoneen välisen vuorovaikutuksen tutkimiseen. Viime vuosien aikana koeasetelmia on alettu siirtää laboratorioista luonnollisiin työympäristöihin.

Tässä diplomityössä esitellään Ganzheit-projektissa kehitetyn katseenseurantajärjestelmän rakenne. Kehitetty järjestelmä tarjoaa avoimen vaihtoehdon kaupallisille järjestelmille. Katseenseurantajärjestelmä hyödyntää mallipohjaista katseenseurantaa, joka on tarkka ja sallii pään vapaan liikkumisen analyysilaitteistoon nähden.

Mallipohjaisen katseenseurannan toimimiseksi silmäkuvasta on löydettävä pupilli ja sarveiskalvosta heijastuva kuvio. Työssä kehitetään menetelmä, jolla kuvasta tunnistetaan vaaditut piirteet. Kehitettyä menetelmää testataan kuudella koehenkilöllä. Tulokset viittaavat pupillin löytyvän silmäkuvasta tarkasti. Testeissä havaitaan kasvojenpiirteiden erojen vaikuttavan merkittävästi sarveiskalvojen heijastusten löytymiseen.

Työssä havainnollistetaan kehitetyn menetelmän käyttöä katseenseurannassa. Testit osoittavat menetelmän toimivan hyvin normaalissa toimistoympäristössä.

Avainsanat: mobiili, katseenseuranta, reaaliaikainen, konenäkö

Preface

This work was done in the Brain Work Research Laboratory in the Finnish Institute of Occupational Health. The thesis is part of the Ganzheit project which is funded by the Academy of Finland.

This thesis could not have been done without the assistance of my instructor, Sharman Jagadeesan. I want to thank Kiti Müller and Andreas Henelius for their sincere interest towards my work. I wish to thank all my colleagues at the Brain and Technology team for refreshing conversations.

Finally, I want to thank my friends and family for the support they have always given me.

Otaniemi, 11.5.2012

Arto Meriläinen

Contents

Contents	v
Abbreviations	vii
List of Tables	viii
List of Figures	ix
1 Introduction	1
1.1 From Laboratories to Natural Environments	1
1.2 Structure of Gaze Tracking Systems	2
1.3 Scope of This Thesis	3
2 Structure of the Human Eye	4
2.1 Anatomical Structures of Visual Perception	4
2.2 Simplified Model of the Eye	5
2.3 Optical Properties of the Iris and the Pupil	8
3 Camera Approximations	11
3.1 The Pinhole Camera Model	11
3.2 Distortions	13
3.3 Camera Calibration	14
3.4 Mapping Image Coordinates to the Environment	15
4 Generally Applied Gaze Tracking Methodologies	16
4.1 Direct Mapping-Based Gaze Tracking	16
4.2 Model-Based Gaze Tracking	17

5	Recording Eye Movements in Mobile Environments	24
5.1	Existing Mobile Gaze Tracking Glasses	24
5.2	Construction of the Gaze Tracking Glasses in the Ganzheit Project .	27
6	A Novel Method for Recognising Corneal Reflections and the Pupil	33
6.1	Preprocessing the Eye Image	33
6.2	Determining the Region of Interest	33
6.3	Finding Corneal Reflections	38
6.4	Finding the Pupil	39
6.5	Assigning the Observed Corneal Reflections to Their Corresponding Light Sources	43
6.6	Implementation	45
7	Validating the Developed Tracking Algorithms	47
7.1	Experimental Setup for Validating the Feature Detection	47
7.2	Inspecting the Goodness of an Algorithm	48
7.3	Results of Feature Detection	49
7.4	Testing Assignment of the Corneal Reflection with Unit Tests	54
8	Discussion	56
8.1	Discussion about the Developed Method	56
8.2	The Mobile Gaze Tracking System	58
A	Parameter Values Used to Validate the Method	64

Abbreviations

Abbreviation	Meaning
CCD	Charge coupled device
CMOS	Complementary metal oxide semiconductor
CR	Corneal reflection
GEM	Gullstrand eye model
LED	Ligth emitting diode
POG	Point of gaze
ROI	Region of interest
SD	Standard deviation
USB	Universal serial bus

List of Tables

2.1	The dimensions of the Gullstrand eye model	6
6.1	The initial grouping of the observed corneal reflections	45
7.1	Sensitivity of pupil detection	51
7.2	Precision of pupil detection	51
7.3	Sensitivity of detecting corneal reflections	53
7.4	Precision of detecting corneal reflection	53
A.1	Parameter values used to validate the method	64

List of Figures

2.1	Anatomical structure of the eye	5
2.2	The simplified eye models	7
2.3	The absorption and the excitation spectra of eumelanin	9
2.4	Different gaze tracking methodologies	10
3.1	The pinhole camera model	11
3.2	The generalised pinhole camera model	12
3.3	Lenses distortions	14
3.4	Mapping image coordinates to the environment	15
4.1	Gaze tracking with direct mapping techniques	17
4.2	The law of reflection and normals of a sphere	18
4.3	Estimation of the centre of the pupil	20
4.4	κ -angle	21
4.5	Calibrating the camera positions	22
5.1	Using a conventional Universal Serial Bus (USB) camera for recording eye movements	25
5.2	Using a hot mirror allows for placing the camera out of sight	26
5.3	Using an infrared camera for recording eye movements	28
5.4	Sensitivity spectra of charge-coupled device (CCD) and complemen- tary metal oxide semiconductor (CMOS) cameras	29
5.5	Estimating the position and orientation of a mirror	31
5.6	Computing the position of a light source	32
6.1	Workflow of the feature detection	34
6.2	Preprocessing the eye image	35
6.3	The Starburst algorithm	36

6.4	Thresholding the eye image to obtain corneal reflections	38
6.5	Thresholding the eye image to locate the pupil	40
6.6	Comparison between different methods used for verifying the consistency of the pupil cluster	42
6.7	Positions of the corneal reflections	44
6.8	Determining the angle between two corneal reflections	45
6.9	Summary of assigning the observed reflections to the light sources . .	46
7.1	The experimental setup	48
7.2	Issues while finding the pupil from an image	50
7.3	An issue while finding the region of interest	50
7.4	Issues with pupil detection	52
7.5	Designing the unit tests	54
7.6	Erroneous assignments of corneal reflections	55
8.1	Utilising model-based gaze tracking for retrieving the point of gaze in the scene image	59

Chapter 1

Introduction

The study of eye movements has a long history in different areas of science. Early studies show that abnormalities in eye movements are connected to e.g. schizophrenia [1], Alzheimer's disease [2] and Parkinson's disease [3]. The studies were made using a scleral search coil, which is an accurate method for studying the movements of the eye [4]. However, the method is invasive and its usage is unpleasant for a patient or test subject.

Development in computer vision and gaze tracking algorithms has enabled researchers to use video cameras for studying eye movements. This allows much more flexible experimental scenarios, as the required equipment is not invasive. Recently, eye movements have been suggested to be used in the area of computer-human interaction [5] and web-page design [6, 7]. Ideas of basing entire user interfaces on gaze tracking techniques have often been suggested [8, 9].

Much of the research occurs in controlled laboratory environments. Early medical applications required fixing the patient to a single location. Current experimental setups do not require wiring the test subject, however, the focus on the analysis is in the interaction between the human and the computer. Despite the advancements in computer vision and gaze tracking algorithms, eye movements are currently analysed mostly in controlled static environments.

1.1 From Laboratories to Natural Environments

Eye movements are of large interest in the emerging field of naturalistic neuroscience, which studies physiologic responses of a human in natural environments [10]. This includes measuring the responses while the test subject is performing daily tasks or interacting with other people. In order to analyse eye movements in a mobile environment, the gaze tracking system must be wearable.

The limitations of a controlled environment have been widely acknowledged and many solutions – both free [11, 12] and proprietary [13, 14] – already exist. The analysis equipment consists of a wearable construction (e.g. a helmet or eye glasses)

holding several cameras. The minimal construction includes two cameras, one facing the eye and another facing the scene. There are commercial products utilising more cameras for analysing the movements of both eyes.

Despite the promising development, only SMI Eye Tracking Glasses from SensorMotoric Instruments (Germany) has been demonstrated to be reliable [15]. However, the gaze tracking system is closed and there is no evidence that the system could be extended freely. Tested gaze tracking systems [11, 14] share an inadequate characteristics: Even slight movements of the equipment during measurement can affect the results significantly. In addition, many of the systems are designed to operate in indoor environments limiting the usage scenarios. In case of commercial products, the price is usually high considering the limitations on extendability.

1.2 Structure of Gaze Tracking Systems

To address the cause for the inadequate performance of the current mobile gaze tracking systems, the structure of mobile gaze tracking systems should be studied more closely. A mobile gaze tracking system consists of five main parts:

1. A wearable device for recording eye movements and the scene image.
2. An algorithm for recognising features from the eye image.
3. An algorithm for computing the direction of the gaze vector based on the extracted features.
4. A method for computing the point of gaze in the scene image.
5. A programme for visualising and storing the results.

There is no common solution for extracting features from the eye image, however, the Starburst algorithm [16] has been widely used as a starting point in different gaze tracking systems (for example [11, 17, 18]). The algorithm is robust, but it contains adjustable parameters that depend highly on the environmental lighting and test subject. Thus, the algorithm requires parameter tuning for each test subject from time to time, which is inconvenient [16].

The extracted features are traditionally processed by applying a direct mapping between the centre of the pupil and a coordinate in the scene image. The approach is simple and computationally efficient, however, even small movements of the analysis equipment affect the results. Artificial neural networks can be utilised for creating a mapping between the coordinate of the centre of the pupil and the coordinate in the scene image [19].

Recently, a new approach for processing the extracted features has emerged. Model-based gaze tracking utilises the structure of the eye in the analysis. The idea is to estimate the centres of the cornea and the pupil from the eye image. The gaze

vector passes through those and can hence be computed. The approach is accurate and computationally efficient. The clear advantages of the method are increased accuracy and free movement of the head with respect to the analysis equipment. However, the approach requires an additional step to transform the gaze vector into a coordinate in the scene image. [20]

1.3 Scope of This Thesis

The model-based approach offers a good theoretical framework for determining the gaze vector. Sensomotoric Instruments openly states that the model-based approach is one of the key elements in their mobile gaze tracking platform [15]. The model-based approach requires finding the features accurately from the eye image and currently there are no known robust and open solutions for extracting the features.

This master's thesis focuses on describing the development of a mobile and open gaze tracking system. In this thesis, a method for extracting the necessary features from the eye image is developed. The thesis is written in the Brain Work Research Laboratory in the Finnish Institute of Occupational Health. This thesis is part of the Ganzheit project, which is funded by the Academy of Finland. The Ganzheit project aims to develop an open and accurate mobile observing platform.

The thesis is divided into eight chapters. Chapters 2-4 describe the anatomy of the human eye, camera models and gaze tracking methodology. Chapter 5 presents the analysis equipment developed for recording eye movements in a mobile environment. Chapter 6 describes the developed method for analysis of the eye image. Chapter 7 describes the testing of the developed method. Finally, the results are discussed in Chapter 8.

Chapter 2

Structure of the Human Eye

This chapter is divided into three sections. First, the anatomical structure related to visual perception is shortly reviewed. The second section describes the Gullstrand theoretical model of the eye, which is utilised later in model-based gaze tracking. The model simplifies the anatomical structure and defines average dimensions for the eye. The third section describes the optical properties of the pupil and the iris. The optical properties determine how light behaves inside the eye or on the surface of the eye. Understanding the optical properties of the eye is relevant for designing the analysis equipment.

2.1 Anatomical Structures of Visual Perception

The structure of the eye is shown in Figure 2.1. The structure is complex, however, only a small fraction of the structure is required for describing the physiology of visual perception. This chapter presents a short summary of the necessary structures of visual perception. For more information about the physiology of the eye, refer to [21].

The outermost part of the eye is the *cornea*, which can be regarded as a spherical appendage on the surface of the eye. The main purposes of the cornea are protecting the inner parts of the eye, refracting incoming light and transmitting light. The radius of curvature of the cornea varies between individuals. [21]

The pupil controls the amount of light that reaches the inner parts of the eye. The pupil can be considered an aperture with a variable size determining the amount of light that passes through it. Because the light is almost completely absorbed by the inner parts of the eye, the pupil is commonly seen as a black circle. [21]

The pupil is surrounded by a thin circular structure called the *iris*, which has muscles connected to the pupil for changing the diameter of the pupil. The iris and the pupil together allow the eye to adapt to different lighting conditions. The iris is pigmented and the colour of the iris is commonly referred as “the colour of the eye”. The pigmentation of the iris is caused by a protein called *eumelanin*. [21]

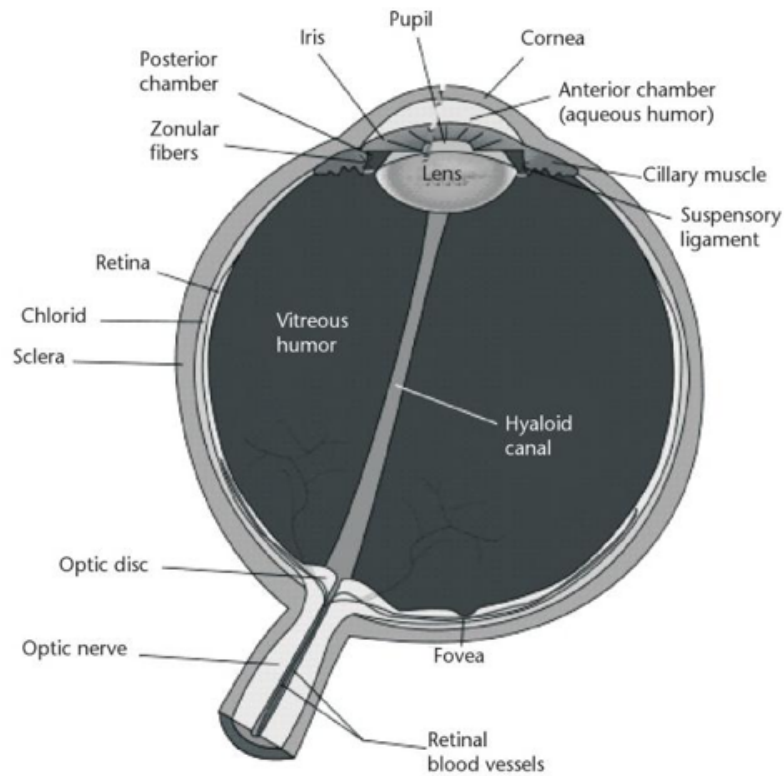


Figure 2.1: Anatomical structure of the eye. [21]

The *lens* is located behind the pupil. The function of the lens is to refract light to make the image focus correctly in the retina. The lens is surrounded by muscles that alter the shape of the lens. The changes in the shape alter the focal length of the lens, which allows the eye to see sharply at different distances. [21]

The *retina* forms the inner surface of the eye. The retina has light sensitive neurons, which are activated by rays of light hitting them. The neurons are not equally distributed, but are gathered into an area called the *fovea*, which is responsible for accurate seeing. The amount of light sensitive cells decreases significantly as the distance to the fovea increases. [21]

2.2 Simplified Model of the Eye

Although knowledge about the structure of the eye is necessary, for model-based gaze tracking this is, however, not sufficient per se. Accurate dimensions of the eye are necessary to make the anatomical model useful.

The dimensions of the eye are described in the *Gullstrand eye model* (GEM) where the dimensions are estimated from population averages. The eye model is illustrated in Figure 2.2a and the parameters of the eye model are summarised in Table 2.1.

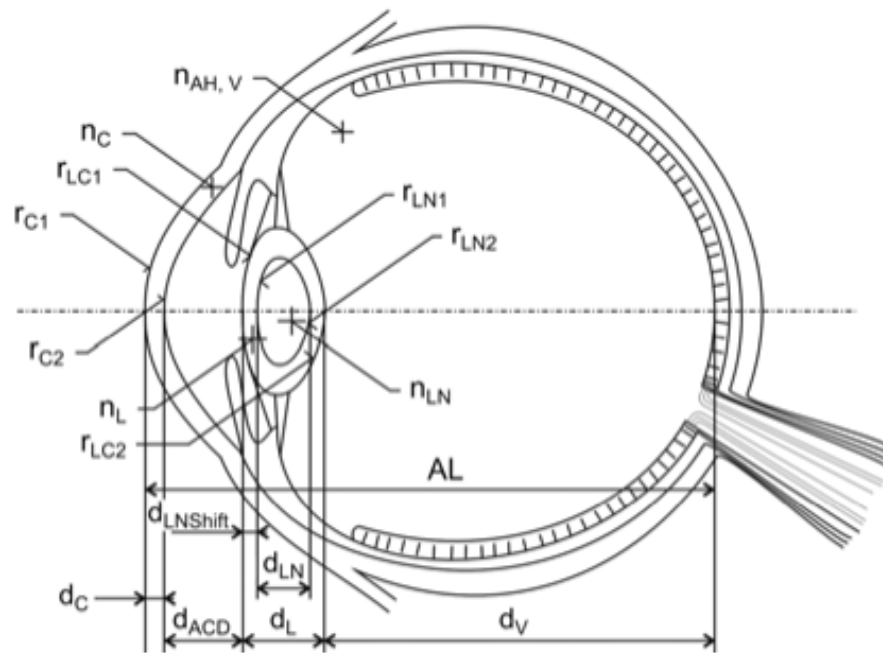
[22]

The GEM describes all necessary parameters for model-based gaze tracking. However, the model can be simplified further. The refraction index of the cornea and the aqueous humor are very similar and hence the refraction occurring on the inner surface of the cornea is small. Therefore, the cornea can be described with a single radius ρ that equals the outer radius R_{C1} . The index of refraction is assumed to be 1.336. The pupil is assumed to be a planar circle located at a distance of $r_d = 3.5\text{mm}$ from the centre of the cornea. This model is illustrated in Figure 2.2b. [20, 22]

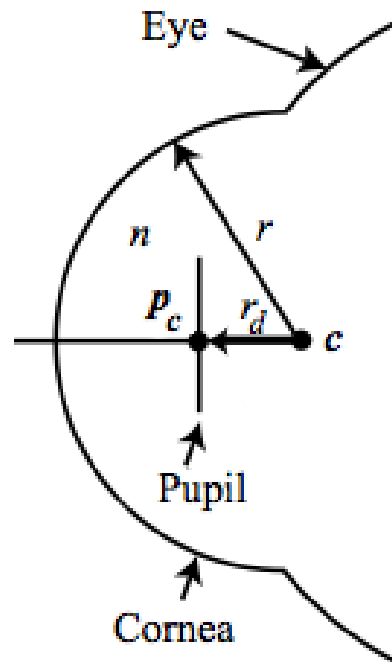
The simplified model assumes that the gaze vector passes through the centres of the pupil and the cornea. Hence, the gaze vector can be estimated by finding the centres of the pupil and the cornea. This idea forms the basis of model-based gaze tracking.

Table 2.1: The dimensions and properties of the Gullstrand eye model. The dimensions were obtained by averaging over large populations. [22]

	Variable	Symbol	Value
Index of refraction	Cornea	n_C	1.376
	Aqueous humor, vitreum	n_{AH}, n_V	1.336
	Lens	n_L	1.386
	Lens nucleus	n_{LN}	1.406
Axial position [mm]	Corneal thickness	d_C	0.50
	Anterior chamber depth	d_{ACD}	3.10
	Lens thickness	d_L	3.60
	Lens nucleus thickness	d_{LN}	2.42
	Lens nucleus distance	$d_{LNshift}$	0.55
	Vitreum length	d_V	16.80
	Axial length	AL	24.00
Radius of curvature [mm]	Front cornea surface	r_{C1}	7.70
	Back cornea surface	r_{C2}	6.80
	Front lens surface	r_{L1}	10.00
	Front lens nucleus surface	r_{LN1}	7.91
	Back lens nucleus surface	r_{LN2}	-5.76
	Back lens surface	r_{L2}	-6.00



(a) The Gullstrand eye model



(b) A simplified eye model

Figure 2.2: The simplified eye models. Figure (a) shows the Gullstrand eye model, which defines the eye dimensions with population averages [22]. Figure (b) shows a simplified eye model where the cornea is presented with a single radius and its thickness is assumed to be infinitely small [20].

2.3 Optical Properties of the Iris and the Pupil

Detecting features accurately from the eye image requires a large contrast between the features in the image. In gaze tracking, distinguishing the pupil from the iris is necessary. Therefore, an understanding of the optical properties of the eye is vital.

2.3.1 Absorption and Excitation of the Iris

There are two important curves which show how eumelanin is visually observed: the absorption and the excitation curves. The absorption curve defines the wavelengths at which eumelanin is considered dark. The excitation curve shows the wavelengths which eumelanin excites when the protein is stimulated with some other wavelength of light. The absorption curve of the protein is shown in Figure 2.3a and the excitation curve is shown in Figure 2.3b. [23]

The absorption curve clearly shows that absorption occurs mostly in the low wavelengths with a maximum in the ultraviolet region of the spectrum. The decrease of absorption is exponential and only little absorbance occurs for wavelengths above 700nm. [23]

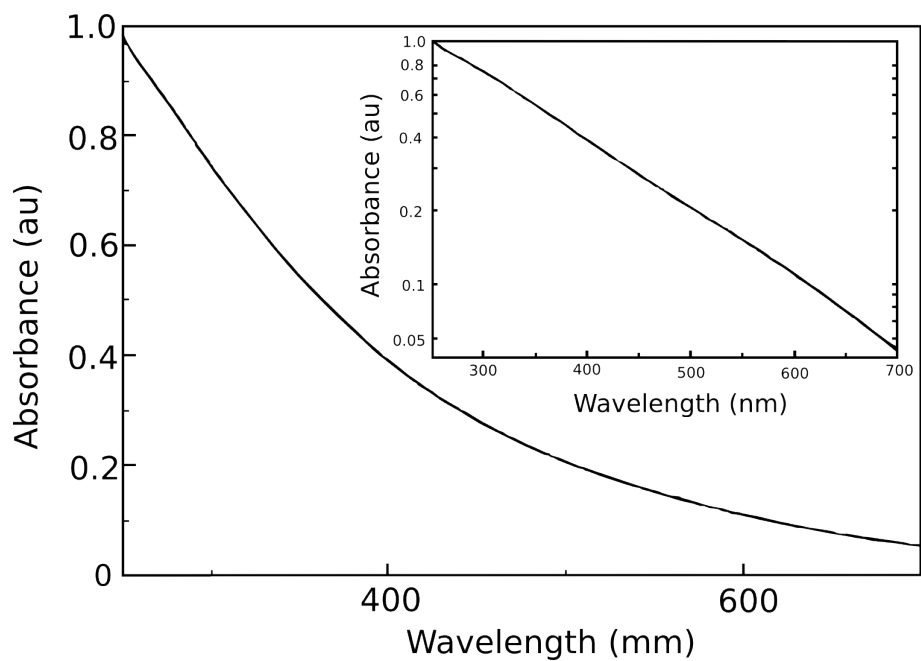
The excitation curve shows that the maximum excitation wavelength is 450 . . . 500nm depending on the stimulation wavelength. The excitation decreases significantly for longer wavelengths. In practice, no excitation occurs for wavelengths below 375nm or above 700nm. [23]

2.3.2 Absorption of the Pupil

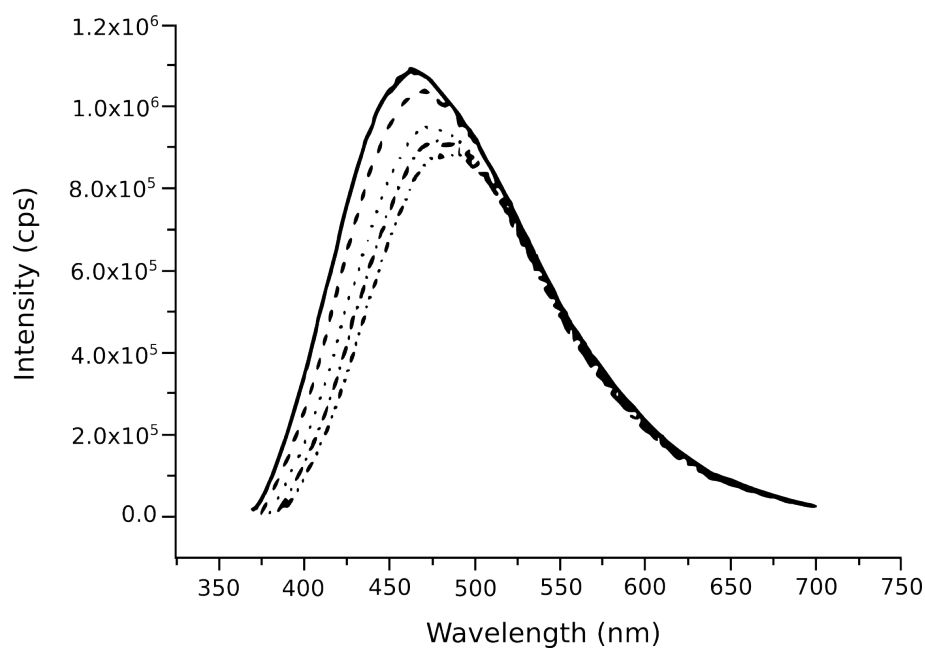
The pupil is an aperture transmitting light into inner parts of the eye and therefore the pupil itself does not absorb light. The transmitted light is absorbed by the retina and the absorption spectrum depends on the relative position between the camera and the light source. Depending on the relative position, gaze tracking methodologies are divided into two categories: *Bright pupil* gaze tracking and *dark pupil* gaze tracking. Both approaches are proper, however, the phenomenon must be noted while positioning the light sources. [24]

In bright pupil gaze tracking the light source is positioned near the camera. The retina reflects the light back to the camera making the pupil seem bright. In case of ordinary cameras, this phenomenon is called the *red-eye effect*. The phenomenon appears if the angle between the camera and the light source is below 5 degrees with respect to the eye. The phenomenon is illustrated in Figure 2.4a. [24]

If the angle between the camera and the light source is large, the retina does not reflect the light back to the camera and the pupil will be seen as black. Dark pupil gaze tracking utilises this phenomenon. Figure 2.4b illustrates this approach. [24]

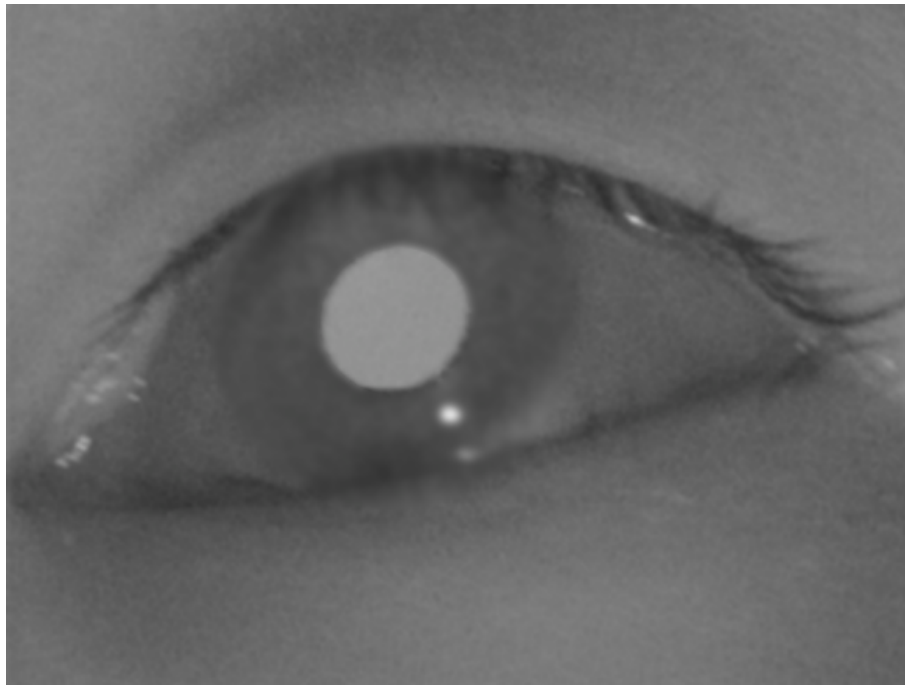


(a) The absorption spectrum of eumelanin

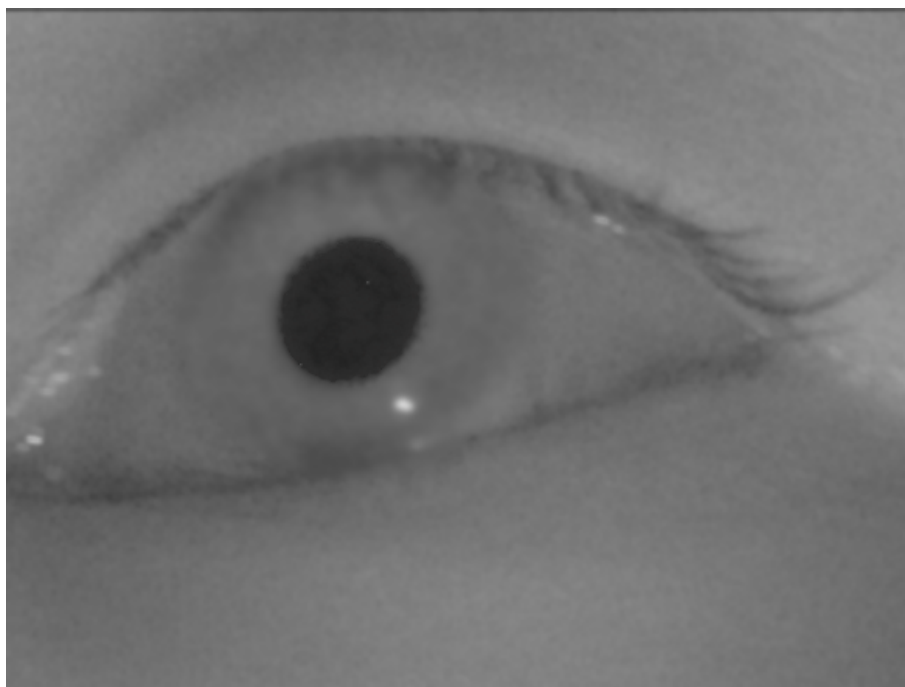


(b) The excitation spectrum of eumelanin

Figure 2.3: The absorption and the excitation spectra of eumelanin. Figure (a) shows the absorption of eumelanin as a function of wavelength. The insert in the figure shows the data in a semi-logarithmic plot. Figure (b) shows the excitation curve of eumelanin with different excitation wavelengths starting from 360nm (outer solid line) to 380nm (inner dashed line). [23, edited]



(a) Bright pupil gaze tracking



(b) Dark pupil gaze tracking

Figure 2.4: Different gaze tracking methodologies. Figure (a) illustrates bright pupil gaze tracking and Figure (b) illustrates dark pupil gaze tracking. [24]

Chapter 3

Camera Approximations

Cameras are used for recording eye movements in video-based gaze tracking. Therefore, it is important to understand their operation principle and limitations. The interest in the current work is in Universal Serial Bus (USB) cameras due to their low price, availability and configurability. In addition, the non-idealities of USB cameras are known and modellable.

3.1 The Pinhole Camera Model

Regular USB cameras can be modelled with the *pinhole camera model*. Figure 3.1 illustrates the basic idea of the pinhole camera. Light from the environment travels through a small hole and projects to an image plane inside the camera. In reality, the image plane is a sensor, which converts light into electric impulses. [25]

Let us consider Figure 3.1 in more detail. The figure presents an object of height X at distance Z from the pinhole plane. The image is projected onto the image plane at distance f behind the pinhole plane. The distance f is usually called the *focal*

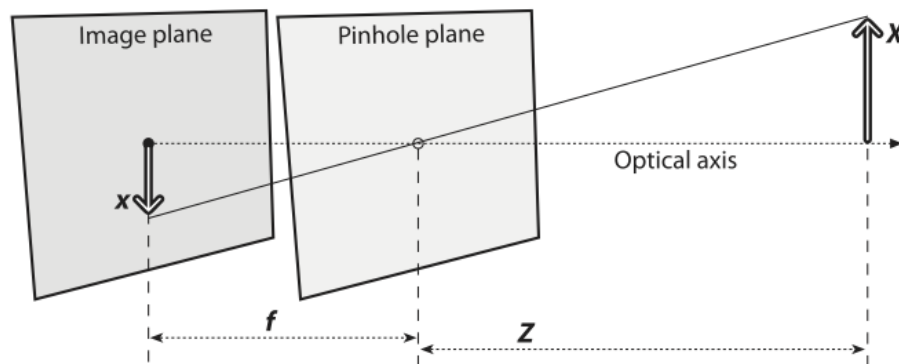


Figure 3.1: The pinhole camera model. Light passes through a small aperture and projects to the back of the camera upside down. [26]

length. The relation between the real height and the height of the projected image can be expressed with the following equation [26]:

$$-x = f \frac{X}{Z}. \quad (3.1)$$

The equation can be generalised to a common case where both x - and y -coordinates are considered. Figure 3.2 illustrates the idea where a three-dimensional point (X,Y,Z) is projected onto a planar point (x,y) . In the figure the imaging plane is moved in front of the pinhole plane at distance f in order to simplify the mathematical notation. Because the imaging plane is in front of the pinhole plane, the image is not flipped upside down. If this notation is used, the transformation for the three-dimensional coordinate is described by equations [26]:

$$x = f_x \frac{X}{Z} \quad (3.2)$$

$$y = f_y \frac{Y}{Z}. \quad (3.3)$$

It is important to note that f_x and f_y are equal only if an ideal pinhole camera is considered. However, this never holds for regular USB cameras and f_x and f_y should not be combined into a single variable. [25]

The origin of the image is commonly in the upper left corner, however, the previous model describes the coordinate with respect to the centre of the sensor. Therefore,

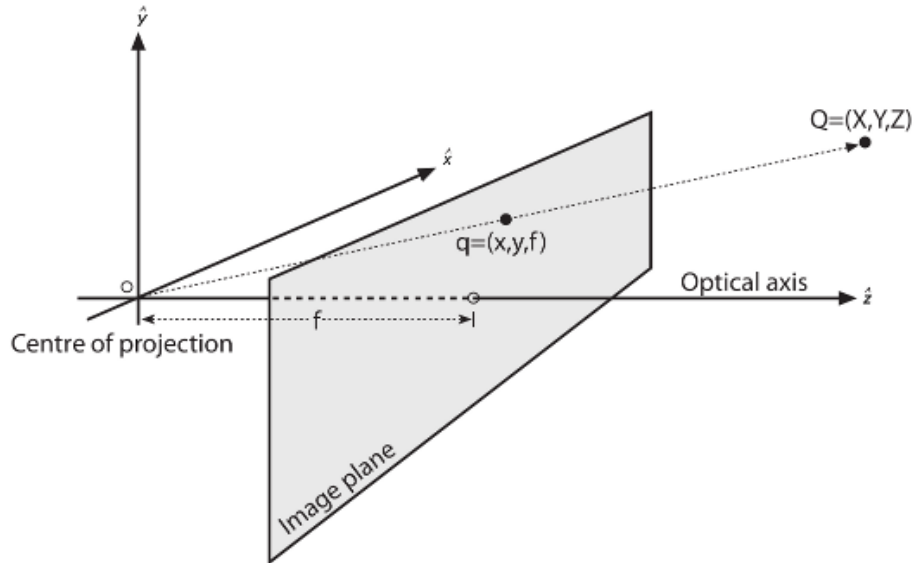


Figure 3.2: The generalised pinhole camera model. A three-dimensional coordinate (X,Y,Z) is projected onto a planar point (x,y) . The imaging plane is moved in front of the pinhole at distance f in order to simplify the mathematical notation. The coordinate axes are denoted by \hat{x} , \hat{y} and \hat{z} [26, edited]

it is necessary to introduce offset terms to the coordinates [26]:

$$x = f_x \frac{X}{Z} + c_x, \quad (3.4)$$

$$y = f_y \frac{Y}{Z} + c_y. \quad (3.5)$$

The previous projection is generally expressed in matrix form as [26]:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{M} \begin{pmatrix} X/Z \\ Y/Z \\ Z/Z \end{pmatrix}, \quad (3.6)$$

where \mathbf{M} denotes a *camera matrix*:

$$\mathbf{M} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}. \quad (3.7)$$

It is important to note the unit. If the focal length f and the sensor centre coordinates are in the metric system, the two-dimensional coordinate will be a metric position of the projection on the image plane. While working with images, pixels are preferred. Hence, the focal length and the sensor centre coordinates should be presented as pixels. [25]

3.2 Distortions

An ideal camera does not distort the image and the mapping works in all cases. However, manufacturing a perfect camera is not possible, which causes distortions to the image. The distortions can be divided into two categories: radial and tangential distortions. The first class of distortions is caused by the lens. The effect of the radial distortions is demonstrated in Figure 3.3a. The second class of distortions is caused by the misalignment of the sensor with respect to the lens. The effect of tangential distortions is demonstrated in Figure 3.3b. [25, 27]

If accurate results are required, the model must be extended to compensate for the distortions of the camera. Let us assume that the distorted coordinates x and y have already been calculated. The radial distortions can be compensated using these equations:

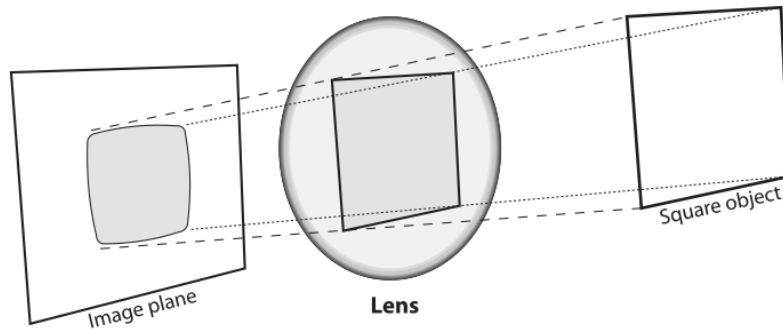
$$x' = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (3.8)$$

$$y' = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \quad (3.9)$$

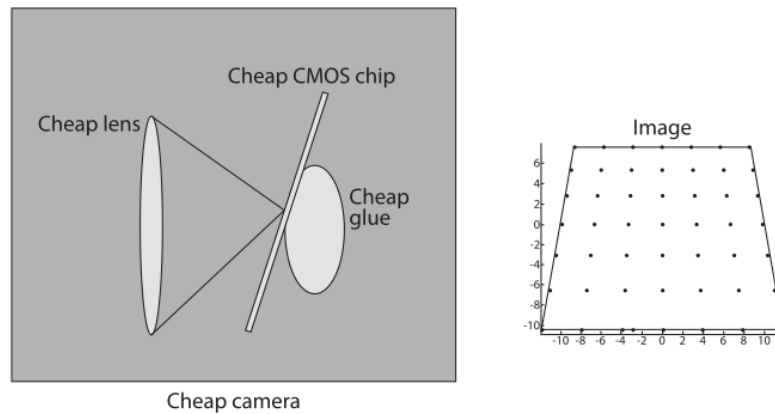
where k_1 , k_2 and k_3 refer to the radial distortion coefficients and $r^2 = x^2 + y^2$. The tangential distortion can be compensated using these equations:

$$x'' = x' + (2p_1 * y + p_2(r^2 + 2x^2)) \quad (3.10)$$

$$y'' = y' + (2p_1 * x + p_2(r^2 + 2y^2)) \quad (3.11)$$



(a) Radial distortions



(b) Tangential distortions

Figure 3.3: Figure (a) illustrates the radial distortions. The radial distortions cause rays farther from the centre of the lens to bend more compared to the rays near the centre. The tangential distortions are illustrated in Figure (b). The tangential distortions are caused by the misalignment of the lens with respect to the imaging plane. [26]

where p_1 and p_2 refer to the tangential distortion coefficients. The coordinate (x'', y'') represents the corrected coordinate in the image. [27]

3.3 Camera Calibration

The previous chapter described the compensation procedure when the compensation parameters are available. Commonly USB cameras are not designed to be used in computer vision applications and the parameters are not available in advance. Therefore, the required parameters must be determined with a calibration procedure.

The calibration is performed by showing an object (or a pattern) to the camera. It is important to know the relative dimensions of the calibration object. The distortion coefficients are estimated by comparing the projected shape to the known relative

dimensions of the object. [26]

3.4 Mapping Image Coordinates to the Environment

The mapping from real coordinates to projected coordinates was shown in the previous section. The inverse operation – determining the position of an object from an image – is commonly more interesting in computer vision. This is illustrated in Figure 3.4.

Depth information is lost in the projection and only the direction vector pointing towards the object can be determined. There is no closed form solution to the transformation and the direction vector must be estimated using numerical methods. [26]

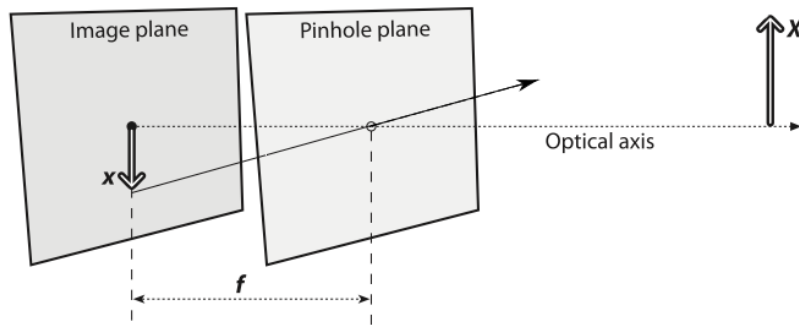


Figure 3.4: Mapping image coordinates to the environment. It is possible to determine a direction vector pointing towards the origin of the ray of light, however, no depth information is available and hence the accurate position cannot be estimated. [26, edited]

Chapter 4

Generally Applied Gaze Tracking Methodologies

Gaze tracking has traditionally required a controlled environment and a fixed head position. When the head position is fixed, gaze tracking can be performed with simple direct mapping techniques. Despite the fact that these algorithms can be used in mobile gaze tracking systems, fixing the head position with respect to the eye camera requires attaching the equipment in an unpleasant fashion. Therefore, the movement of the head should be regarded as a requirement.

This chapter reviews some commonly used direct mapping methods and the properties of the methods. The main focus of the chapter is in model-based gaze tracking. The approach allows free movement of the head with respect to the analysis equipment making it an obvious choice for mobile gaze tracking systems.

4.1 Direct Mapping-Based Gaze Tracking

Traditionally, direct mapping techniques have been applied in gaze tracking. Direct mapping-based approaches are accurate (the error is commonly less than a degree) and computationally efficient. [16]

In direct mapping techniques the problem is approached by tracking the centre of the pupil from the eye image. The located centre coordinate is then mapped to the scene image by applying a transformation for the pupil coordinate (see Figure 4.1a). The transformation is generally estimated with a calibration sequence before or after the measurement. [16, 28]

The mapping becomes incorrect if the measurement device moves even slightly. In order to overcome the issue, a light source is commonly used to create a bright reflection on the cornea. These reflections are called *corneal reflections* (CRs). The movements of the camera do not cause abrupt changes to the position of the CR. Therefore, the the effect of the movements of the measurement device can be decreased by using the distance from the CR to the centre of the pupil (see Figure

4.1b). [16]

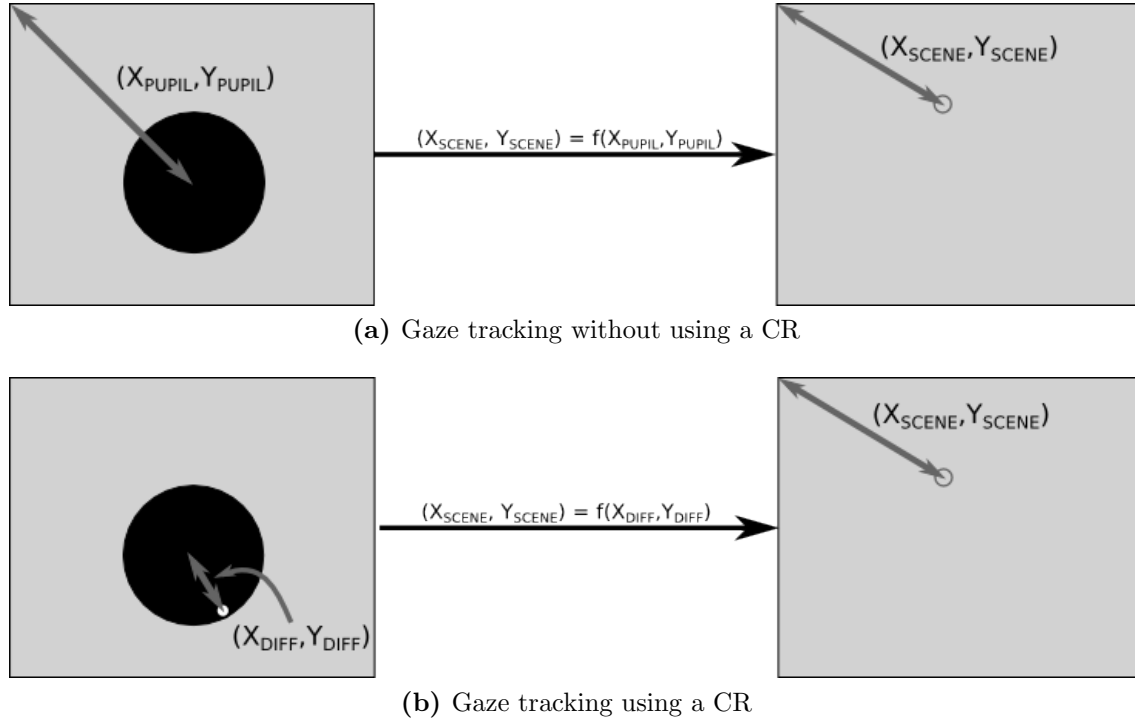


Figure 4.1: Gaze tracking with direct mapping techniques. Figure (a) illustrates direct mapping without using a CR and Figure (b) illustrates the mapping using a CR. The figures on left show the eye image and the figures on right show the scene image.

4.2 Model-Based Gaze Tracking

The approaches based on direct mapping are computationally efficient and the mapping techniques provide reasonable accuracy [16]. However, the mapping techniques are extremely vulnerable to movements of the measurement device even when the CR is used. If the device moves even slightly, the mapping becomes incorrect. Hence, more sophisticated techniques are required when working in mobile environments. [20]

Figure 2.1b in Chapter 2 illustrates the functional structure of the eye. The figure shows that the gaze vector passes through the centres of the cornea and the pupil. The central idea in model-based gaze tracking is to determine the centres of the pupil and the cornea. The gaze vector is mapped to the scene image or the model of the environment. [20]

4.2.1 Computing the Centre of the Cornea

The cornea can be modelled as a sphere with a radius of $\rho = 7.70\text{mm}$ [22]. To be more precise, the cornea can be modelled as a spherical mirror that follows the law of reflection [22, 29]. The law of reflection describes that the angle between the incoming light and the surface normal equals the angle between the reflected light and the surface normal [29]. These observations are essential for model-based gaze tracking [20]. The law of reflection is illustrated in Figure 4.2a.

The key idea in computing the centre of the cornea is finding multiple surface normals of the cornea [20]. The surface normals of a sphere pass through the centre of the sphere. Therefore, the intersection of these normals define the centre of the cornea. Figure 4.2b illustrates this idea.

The normals of the cornea sphere are estimated at the observed CRs of individual light sources. Estimating the centre of the cornea requires knowing accurately the position of a light source with respect to the camera. It is also necessary that the camera observes the CR of the light source, because the direction vector towards the CR is required. Providing that the position of the light source and the direction vector towards the CR are available, the law of reflection and the idea of intersections can be used for determining the centre of the cornea (\mathbf{c}_1) as a function of a single unknown variable. [20]

Because the centre of the cornea is a three-dimensional coordinate, the centre is presented by three equations all of which depend on the unknown variable.

Introducing another light source provides another estimate for the centre of the cornea (\mathbf{c}_2) that also depends on another unknown variable. Because both esti-

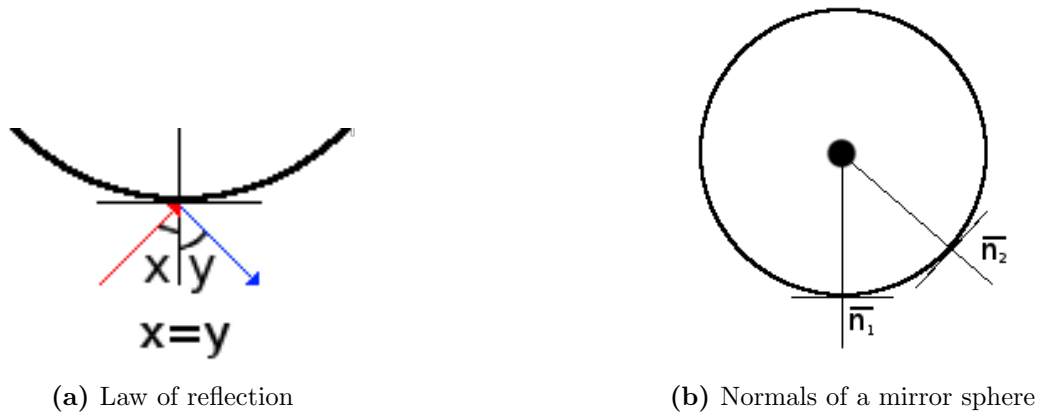


Figure 4.2: Figure (a) illustrates the law of reflection. The figure shows how light reflects from the surface of a mirror sphere: The angle between the incoming light and the surface normal (angle x) equals the angle between the reflected light and the surface normal (angle y). Figure (b) illustrates that the normals of a mirror sphere intersect at the centre of the sphere.

mates denote the same centre by three equations (i.e. $\mathbf{c}_1 = \mathbf{c}_2$) and both estimates introduce only a single unknown variable, the group of equations becomes overdefined and hence the unknown variables can be solved. [20]

The group of equations cannot be solved analytically. Therefore, the problem must be solved by defining an error function to be minimised with numerical methods.

The centre of the cornea can be estimated using two light sources, however, adding light sources increases the number of normals, which can be utilised for increasing the estimation accuracy. Each light source pair gives an estimate for the centre of the cornea and the total number of pairs is computed using the equation:

$$P(n) = \sum_{i=1}^{n-1} i, \quad (4.1)$$

where $P(n)$ denotes the number of pairs formed from n light sources. Each pair produces three equations and the total number of equations is obtained using the equation:

$$L(n) = 3 \sum_{i=1}^{n-1} i, \quad (4.2)$$

whereas the number of unknown parameters follow the formula:

$$U(n) = n. \quad (4.3)$$

For example, three light sources produce nine limiting equations and three unknown variables.

4.2.2 Computing the Centre of the Pupil

The second point needed for defining the gaze vector, the centre of the pupil, is determined after estimating the centre of the cornea. Figure 4.3 illustrates the idea of finding the three-dimensional coordinate of a pupil contour point.

An ellipse estimating the pupil contour is extracted from the eye image using a method presented in Chapter 6. A predefined number of pupil contour point pairs with one point at one edge and another point at the opposite edge are selected. The three-dimensional coordinates of these points are estimated and the centre of the pupil is estimated by computing the centroid of the obtained three-dimensional pupil contour coordinates. [20]

Let us denote a selected point with index i . For each i , the direction vector $\hat{\mathbf{K}}_i$ can be computed. Because the pupil is inside the cornea, the direction vector intersects with the cornea sphere at point \mathbf{u}_i . The light is refracted into the cornea towards the pupil perimeter point \mathbf{u}'_i .

Several direction vectors $\hat{\mathbf{K}}'_i$ towards the pupil contour are computed. Their depths are estimated using the information about the eye structure. The distance between

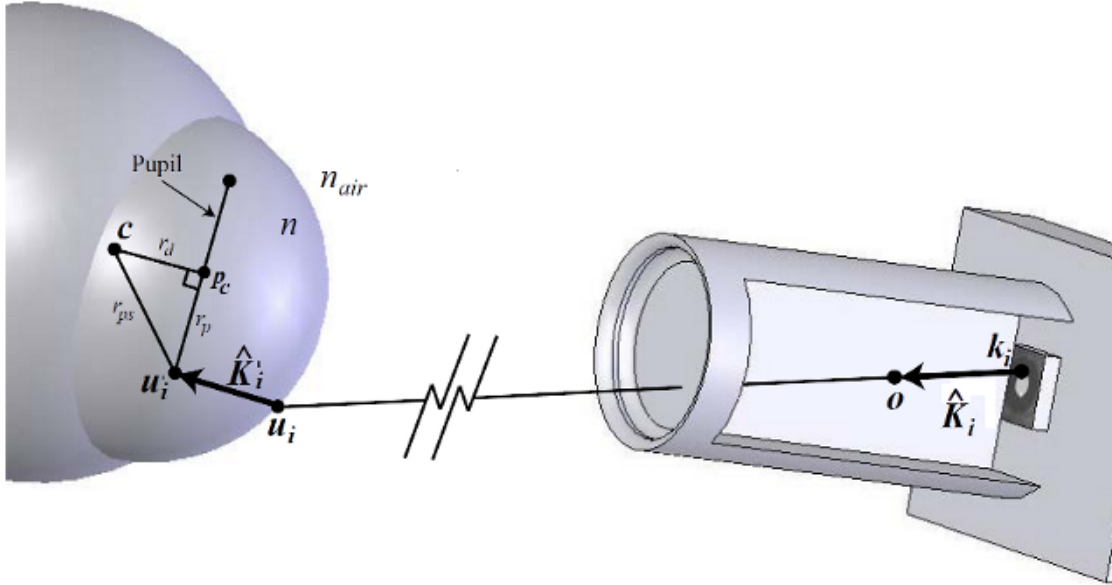


Figure 4.3: Estimation of the centre of the pupil. The position of the pupil perimeter is estimated using knowledge of the centre of the cornea \mathbf{c} and the population statistics (variables r_d and n). [20, edited]

the centres of the cornea and the pupil, r_d , is approximately 3.75mm (refer to Section 2.2). The distance between the centres cannot be used as a restriction, however, if the radius of the pupil (r_p) can be estimated, the distance from the centre of the cornea to the pupil perimeter (r_{ps}) can be computed. By introducing the restriction that the distance between the pupil perimeter point \mathbf{u}'_i and the centre of the cornea must be r_{ps} , it is straightforward to compute the pupil contour point \mathbf{u}'_i . [20]

The end points of the major axis of the pupil ellipse are used to estimate the radius of the pupil. The direction vectors towards the end points, $\hat{\mathbf{K}}_{MA1}$ and $\hat{\mathbf{K}}_{MA2}$, are determined. Obviously, the direction vectors intersect with the cornea at points \mathbf{u}_{MA1} and \mathbf{u}_{MA2} . The distance between these points is assumed to equal the pupil diameter, $2r_p$.

4.2.3 κ -correction

The retina is not homogeneous, but the concentration of photoreceptors is highly centralised near the fovea, which is responsible for sharp vision. Because the fovea is used for accurate vision, the gaze vector should also hit the fovea. So far the gaze vector is defined as a vector passing through the centres of the cornea and the pupil. However, the fovea does not generally reside on the estimated gaze vector. This issue is illustrated in Figure 4.4. Compensating for this issue is known as κ -correction. [30]

The κ -angle can be estimated with the help of the person wearing the gaze tracking

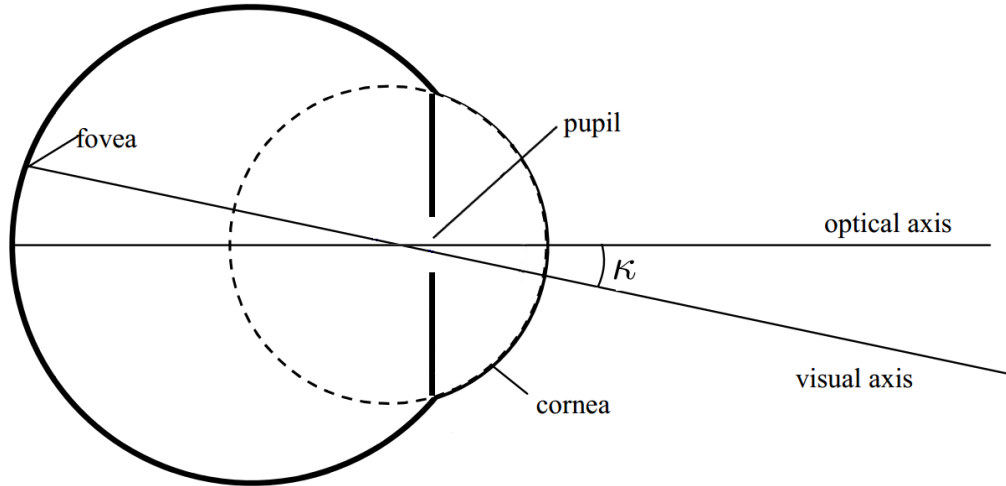


Figure 4.4: κ -angle. The optical axis rarely equals the visual axis. The angle between the visual axis and optical axis is marked by κ in the figure. [31, edited]

apparatus. The position of the calibration point with respect to the eye is estimated using some measurement technique. The person is then asked to look at the calibration point. κ -angles can be retrieved by computing the difference between the real gaze vector and the estimated gaze vector.

4.2.4 Mapping the Gaze Vector to the Scene Image

The gaze vector only describes the direction in which the test subject is looking. However, the interest is usually in the *point of gaze* (POG) in the scene image rather than in the gaze vector itself. Determining the POG in the scene image is not trivial, because two-dimensional mapping on the scene image is dependent on the depth information, which is not available for the gaze vector. The issue can be solved by making assumptions about the gaze vector.

Mapping the gaze vector to the scene image or the environment is beyond the scope of this thesis and hence this topic is treated briefly.

4.2.4.1 Calibrating the Camera Positions

If the direct mapping techniques described in Section 4.1 are used in gaze tracking, the cameras can be moved without any issues. The calibration procedure is performed before or after each measurement and the mapping is always be correct. However, model-based gaze tracking is not that simple.

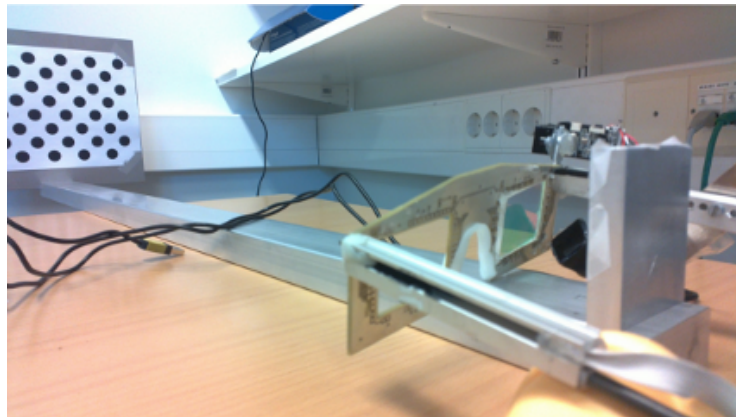
The gaze tracking glasses are equipped with two cameras and both of them have their own coordinate systems. The gaze vector is computed in the coordinates of the eye camera, but the mapping to the scene image is performed in the coordinates of

the scene camera. Therefore, a method for transforming the eye camera coordinates to the scene camera coordinates must be developed.

The mapping matrix ($\mathbf{A}_{\mathbf{EC} \rightarrow \mathbf{SC}}$) from the eye camera to the scene camera is determined with a precision built calibration scaffold (see Figure 4.5a). The calibration scaffold has separate calibration patterns for both cameras. As the scaffold is precision built, the transformation from one calibration pattern to the other is known accurately. Therefore, a transformation matrix from the eye camera coordinates to the scene camera coordinates can be computed when both cameras can see their patterns (see Figure 4.5b).



(a) The calibration scaffold



(b) Using the calibration scaffold

Figure 4.5: Calibrating the camera positions. Figure (a) shows the precision built calibration scaffold that is used for calibration. The scaffold has a small pattern for the eye camera and a large pattern for the scene camera. Figure (b) demonstrates the usage of the scaffold for determining the transformation matrices.

4.2.4.2 Assuming a Constant Depth

The gaze vector can be mapped to the scene image by assuming that the POG has always the same distance from the centre of the cornea. The three-dimensional coordinate is written as:

$$\mathbf{POG}_{\mathbf{EC}} = R \cdot \frac{\mathbf{c}_p - \mathbf{c}_c}{\|\mathbf{c}_p - \mathbf{c}_c\|} + \mathbf{c}_c, \quad (4.4)$$

where \mathbf{c}_p denotes the centre of the pupil, \mathbf{c}_c the centre of the cornea and R the distance from the centre of the cornea to the POG. The coordinate can be transformed to the scene camera coordinate system:

$$\mathbf{POG}_{\mathbf{SC}} = \mathbf{A}_{\mathbf{EC} \rightarrow \mathbf{SC}} \mathbf{POG}_{\mathbf{EC}}. \quad (4.5)$$

Providing that the gaze vector has been mapped to the scene camera coordinate system, the POG can be estimated using the pinhole camera model (see Section 3.4 for details).

This approach requires no additional information about the gaze vector. However, the approach fails if the approximated distance differs much from the real distance of the gaze.

4.2.4.3 Using the Properties of Stereo View

The eyes do not share the same gaze vector, because the eyes look towards the same point in space. It is possible to find the POG in three-dimensional space by determining the intersection point of the gaze vectors. The approach offers an automatic method for determining the POG.

4.2.4.4 Utilising a Model of the Environment

If a model of the environment and the position of the user are available, the POG can be estimated accurately. The idea is finding the point at which the gaze vector intersects with an object in the model.

Clearly, the approach has several drawbacks. Finding the position of the user with respect to the environment is not trivial. Creating a model of the environment is possible, however, no unmapped objects or persons should exist in the environment.

Despite the difficulties in implementing the method, a clear advantage exists. Because the POG is mapped directly to the model of the environment, the method allows naming individual objects at which the user has looked while wearing the gaze tracking glasses.

Chapter 5

Recording Eye Movements in Mobile Environments

Recording eye movements in mobile environments is challenging. The greatest difficulties are varying lighting conditions and confounding reflections seen on the cornea. The issues can be prevented in controlled environments, but in mobile environments the issues are challenging. This chapter overviews selected reference designs from the literature and introduces the gaze tracking glasses developed in the Ganzheit project.

5.1 Existing Mobile Gaze Tracking Glasses

All gaze tracking glasses previously presented in the literature share a similar design. The glasses are equipped with a camera for recording the scene and another camera for recording the eye.

Eye movements can be observed with a visible light camera. This approach allows retrieving the eye image with an ordinary camera. Figure 5.1a shows the openEyes gaze tracking glasses that is equipped with an ordinary camera for recording the eye. Figure 5.1b shows an image that was captured with a visible light camera in controlled conditions. The image shows clearly a reflection of the monitor that is placed 1 meter away from the eye. [11]

The human eye cannot see infrared light, but cameras detect those wavelengths. There are also optical components that act differently for different wavelengths. A hot mirror is an optical component which passes visible light but reflects infrared light. If infrared light is utilised, the properties of the selective optical components can be utilised while designing the hardware. The eye camera can be placed out of sight by placing a hot mirror in front of the eye. The hot mirror reflects infrared light from the environment and therefore this design solves also the issue of unwanted reflections from the environment. The idea is illustrated in Figure 5.2a and a reference hardware implementation is shown in Figure 5.2b. [32]

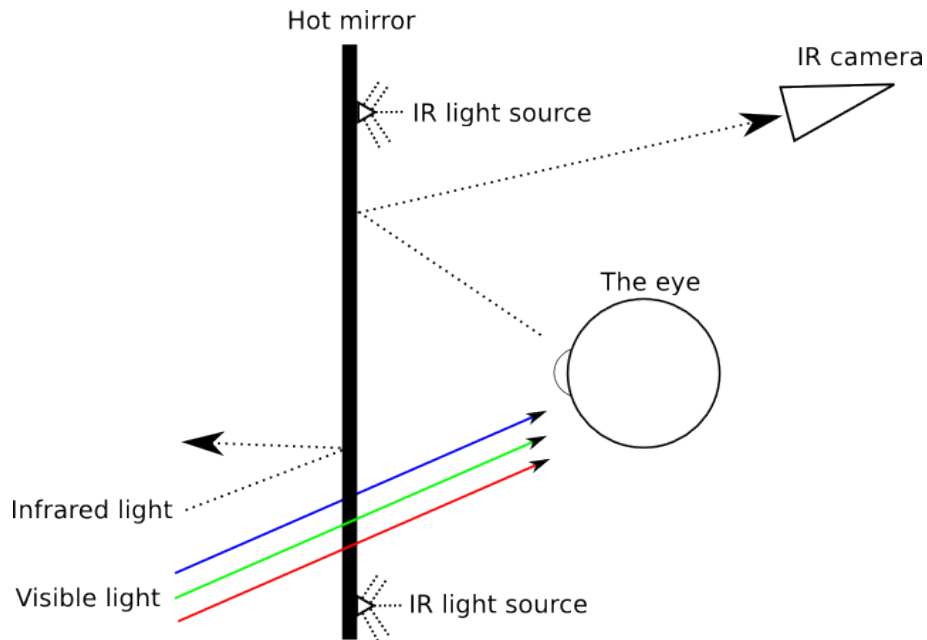


(a) Gaze tracking glasses used in the openEyes project



(b) An eye image captured using an ordinary USB camera

Figure 5.1: Using a conventional USB camera for recording eye movements. Figure (a) shows the glasses utilised in the openEyes project (image source [11]). The camera on the glasses observes visible light and the camera is placed in front of the eye. Figure (b) shows an eye image that was taken in controlled conditions with an ordinary USB camera. The reflection of a light source can be clearly observed. However, also a reflection of a monitor placed 1m away can be seen (marked with a red circle in the image).



(a) Schematic idea of using a hot mirror



(b) Eye tracking glasses using hot mirrors

Figure 5.2: Using a hot mirror allows for placing the camera out of sight. Figure (a) shows the key idea of using infrared light and a hot mirror. Visible light passes through the hot mirror whereas infrared light reflects from it. This reduces artefacts from the environment and allows for moving the camera to a less disturbing position. Figure (b) shows a reference implementation utilising the idea [32, edited].

5.2 Construction of the Gaze Tracking Glasses in the Ganzheit Project

The previous section reviewed some gaze tracking glasses from the literature. The glasses offer interesting starting points for developing gaze tracking glasses in the Ganzheit project. This chapter describes the constructed gaze tracking glasses and a procedure for estimating positions of the light sources.

The glasses developed in the openEyes project observe visible light with a conventional USB camera, which causes two significant issues. First, reflections from the environment will be seen in the eye image. Second, controlling the lighting inside the glasses while observing the visible spectrum requires blocking all light from the environment, which disturbs the test subject and affects the measurement result.

Visible light may not be ideal for gaze tracking considering the optical properties of the eye. Section 2.3.1 described that the iris excites light at the visible part of the spectrum, which may affect the measurement. In addition, the section noted that absorption occurs for visible light and hence the iris may be observed dark if the wavelength of the light is too short. If near-infrared light (wavelength above 750nm) is considered, the iris should appear bright as only little excitation and absorption occurs.

As noted in the previous chapter, the usage of infrared light allows for a more flexible design. A hot mirror reduces artefacts from the environment and allows positioning the eye camera out of sight. Most notably, the infrared light allows for controlling the lighting near the eye without disturbing the test subject. The gaze tracking glasses are built using these ideas. Figure 5.3a shows the constructed glasses and an image captured with the eye camera is shown in Figure 5.3b.

The glasses are built using a printed circuit board (PCB), which has surface mounted infrared *light emitting diodes* (LEDs) as light sources. The eye camera is placed next to the eye. The scene camera is placed near the right temple.

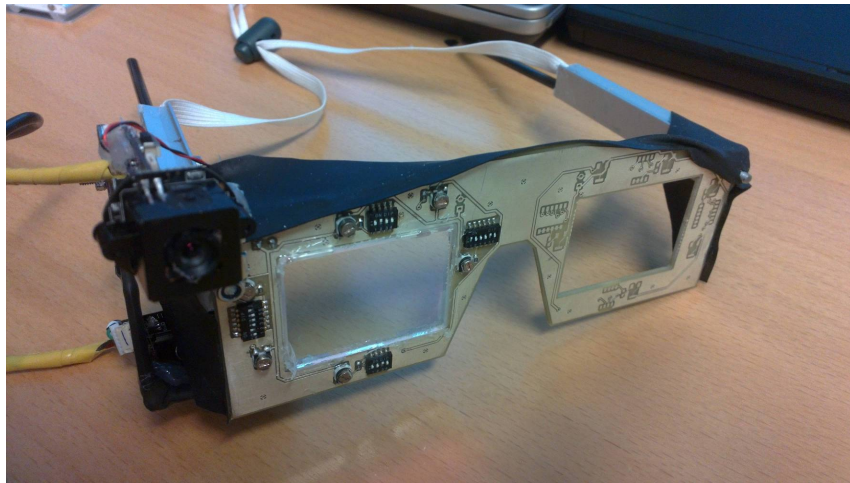
5.2.1 Selecting an Infrared Camera

Manufacturers of USB cameras rarely publish the sensitivity spectrum. However, it is common knowledge that cameras use either *complementary metal oxide semiconductor* (CMOS) or *charge-coupled device* (CCD) to detect light. CCD cameras are similar, hence, the sensitivity spectrum of any CCD camera applies to every CCD camera. The same applies to CMOS cameras. Figure 5.4a shows the sensitivity spectrum of a CCD camera and Figure 5.4b shows the sensitivity spectrum of a CMOS camera. The figures show clearly that CCD and CMOS cameras are adequate for detecting infrared light. Therefore, any USB camera can be used to detect infrared light. [33]

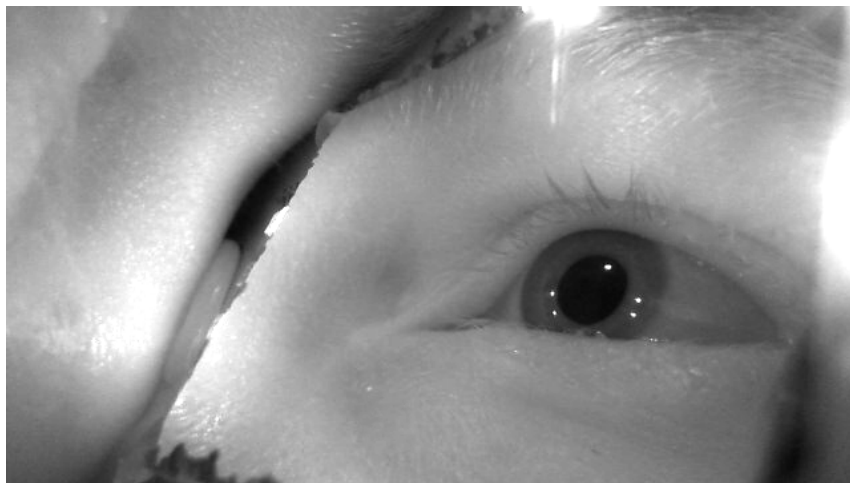
Ordinary USB cameras are intended to observe visible light and near-infrared light is generally regarded as noise in the image. Therefore, an infrared filter is commonly

placed in the optical path of a camera. In order to modify an ordinary USB camera to an infrared camera, the infrared filter must be replaced with a visible light filter. Removing the filter from a USB camera is usually a simple operation. [33]

The frame rate of the camera should not vary. In order to obtain a stable frame rate, the exposure time should be set to some predefined constant value. This feature is available only in some USB camera models.

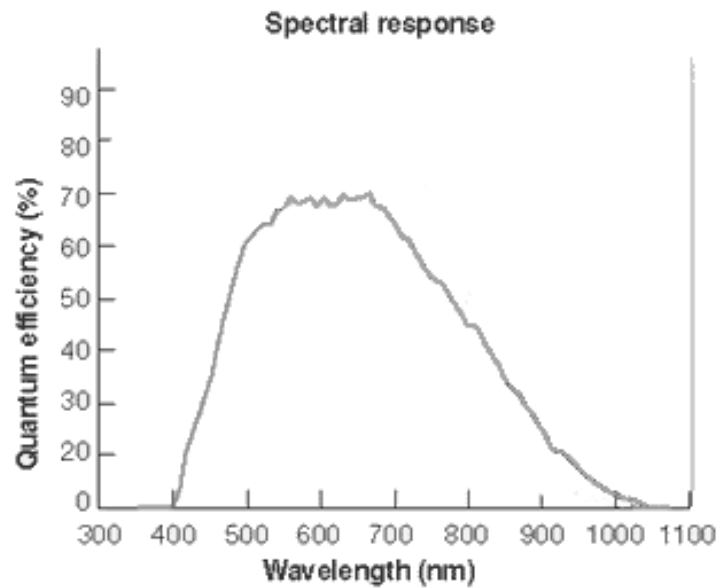


(a) Gaze tracking glasses used in the Ganzheit project

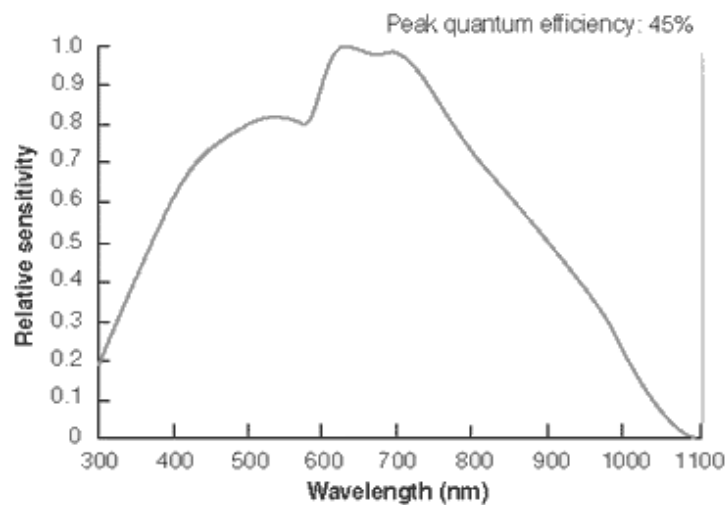


(b) An eye image captured using infrared light

Figure 5.3: Using an infrared camera for recording eye movements. Figure (a) shows the gaze tracking glasses developed in the Ganzheit project. The eye camera detects infrared light and the camera is placed behind the glasses. A hot mirror is placed in front of the eye. The camera observes a reflection of the eye in the hot mirror. Figure (b) shows an image captured with the glasses in an office room. No unwanted reflections are seen in this image.



(a) Sensitivity spectrum of CCDs



(b) Sensitivity spectrum of CMOSes

Figure 5.4: Sensitivity spectra of CCD (a) and CMOS (b) cameras. The spectra show that regular CCDs and CMOSes are sensitive to infrared light if no infrared filter is present. [33, edited]

5.2.2 Selecting the Light Sources

Light sources are used to generate CRs that are used for estimating the centre of the cornea. In addition, light sources are required for creating general lighting inside the gaze tracking glasses.

The wavelength of the light sources should be in the near-infrared band. The sensitivity spectra shown in Section 5.2.1 show that the cameras are commonly more sensitive to short wavelengths. Section 2.3 describes the absorbance of the pupil and the iris. In short, the pupil absorbance spectrum is broad in the near-infrared band given that the distance between each light source and the camera is large. The absorbance of the pupil is always higher than the absorbance of the iris, thus, the pupil is distinguishable.

The constructed gaze tracking glasses are equipped with surface mountable LEDs whose emission spike is at $850nm$. The prototype allows for testing multiple different configurations and setting the brightness of the LEDs.

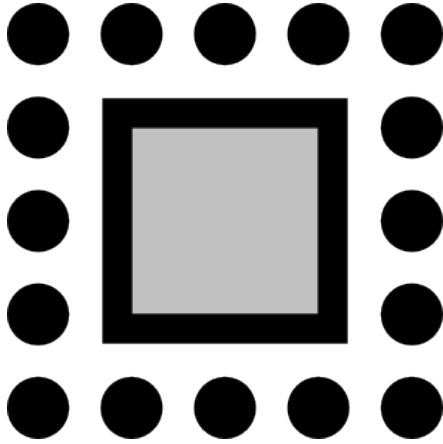
While selecting the light sources and setting the brightnesses of the LEDs, it is important to remember infrared safety. In order to avoid risks caused by the light sources the analysis device must be validated adhering to Directive 2006/25/EC of the European Parliament on infrared safety [34].

5.2.3 Calibrating the Positions of the Light Sources

As discussed in Section 4.2, the accurate positions of the light sources must be known for estimating the centre of the cornea. Because the position of the focal point is unknown, the exact position of a LED with respect to the focal point cannot be measured directly. However, the position can be computed indirectly by utilising a first surface mirror.

The method consists of two parts. First, the position and orientation, i.e. transformation, of the mirror with respect to the camera is estimated. The estimation is accomplished by placing a pattern on the mirror. If the camera detects the pattern and the exact dimensions of the pattern are known, the transformation of the mirror can be estimated. The pattern is shown in Figure 5.5.

Second, the mirror is utilised for determining a direction vector towards a light source. This operation is accomplished by utilising the law of reflection, which states that the angle between the normal of the mirror and arriving light is the same as the angle between the normal and the reflected light [29]. Because the transformation of the mirror is known, the normal of the mirror can be estimated. If the camera can detect the reflection of the light source in the mirror, the direction vector towards the point in the mirror can be determined (see Chapter 3.4 for details). When both the direction vector towards the mirror and the normal of the mirror are known, the direction vector from the mirror towards the light source can be estimated. Estimating the direction vector is demonstrated in Figure 5.6.



(a) The calibration pattern



(b) The calibration pattern on a mirror

Figure 5.5: Estimating the position and orientation of a mirror. Figure (a) shows the calibration pattern placed on the mirror. The grey area inside the rectangle represents the surface of the mirror. Figure (b) shows the calibration pattern on a mirror.

The position of the light source cannot be determined with a single direction vector. However, with multiple direction vectors obtained from different orientations of the mirror, the intersection point of the direction vectors should be the position of the light source.

In theory, two samples are enough. However, the result can be improved by collecting multiple samples and searching point \mathbf{P} which is closest to all direction vectors. This is done by minimising the error function E :

$$E = \sum_{n=0}^N (d_n(\mathbf{P}))^2, \quad (5.1)$$

where N denotes the number of obtained vectors and $d_n(\mathbf{P})$ denotes the smallest distance between point \mathbf{P} and the n th direction vector. The function $d_n(\mathbf{P})$ is defined as

$$d_n(\mathbf{P}) = \frac{\|\hat{\mathbf{S}}'_n \times (\mathbf{P} - \mathbf{S}_n)\|}{\|\hat{\mathbf{S}}'_n\|} = \|\hat{\mathbf{S}}'_n \times (\mathbf{P} - \mathbf{S}_n)\|, \quad (5.2)$$

where $\hat{\mathbf{S}}'_n$ denotes the direction vector from the mirror towards the light source and \mathbf{S}_n denotes the vector from the camera to the point of the reflection in the mirror.

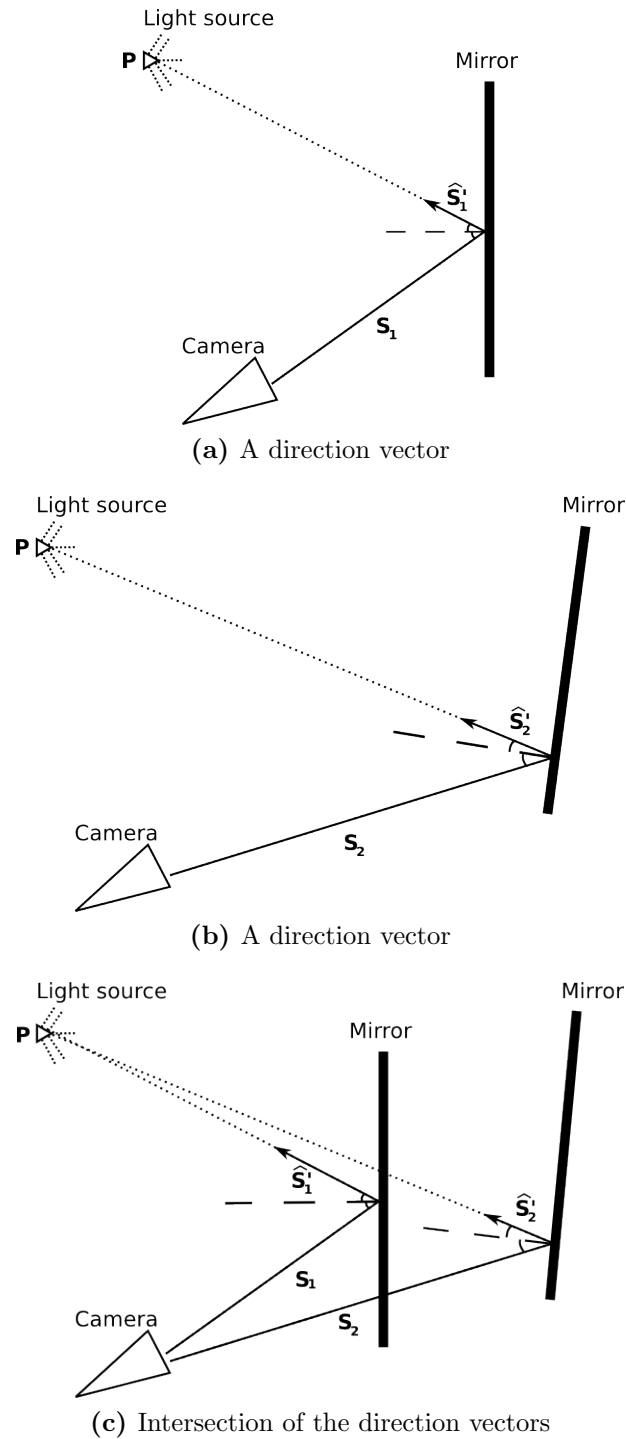


Figure 5.6: Computing the position of a light source. A mirror is positioned in front of the camera allowing the camera to see the reflection of the light source. A direction vector towards the light source is computed using the law of reflection (Figures (a) and (b)). By combining multiple direction vectors, the position of the light source can be estimated (Figure (c)). The solid lines represent the reflection. The dotted lines represent the direction vectors towards the light source. The dashed lines represent the normals of the mirror.

Chapter 6

A Novel Method for Recognising Corneal Reflections and the Pupil

This chapter introduces a novel method for recognising the necessary features for model-based gaze tracking. The workflow is illustrated in Figure 6.1. This chapter first presents a method for finding the approximate position of the pupil from an image. Second, a method for finding the CRs from the eye image is introduced. The CRs are unwanted artefacts in estimation of pupil ellipse and are hence removed before finding the pupil. Third, the estimation of the pupil contour is studied closely. Finally, a method for assigning the observed CRs to the corresponding light sources is presented.

6.1 Preprocessing the Eye Image

The quality of the eye image is improved by preprocessing. Its size is reduced to the area of interest. If the gaze tracking glasses utilise the mirror technique described in Chapter 5, the eye camera may observe parts of the gaze tracking glasses. The analysable area is manually selected and its size and position are related only to the gaze tracking glasses.

The noise in the image is decreased by Gaussian filtering [26]. The Gaussian filter computes a moving average over neighbouring pixels removing rapid changes. Figure 6.2a shows a raw image and Figure 6.2b shows the image after preprocessing.

6.2 Determining the Region of Interest

The preprocessed eye image does usually not include only the eye, but also several artefacts. These artefacts include skin and eyelashes. In order to enhance the performance and robustness of the analysis, it is necessary to define the *region of interest* (ROI). This operation excludes artefacts that might disturb the next stages

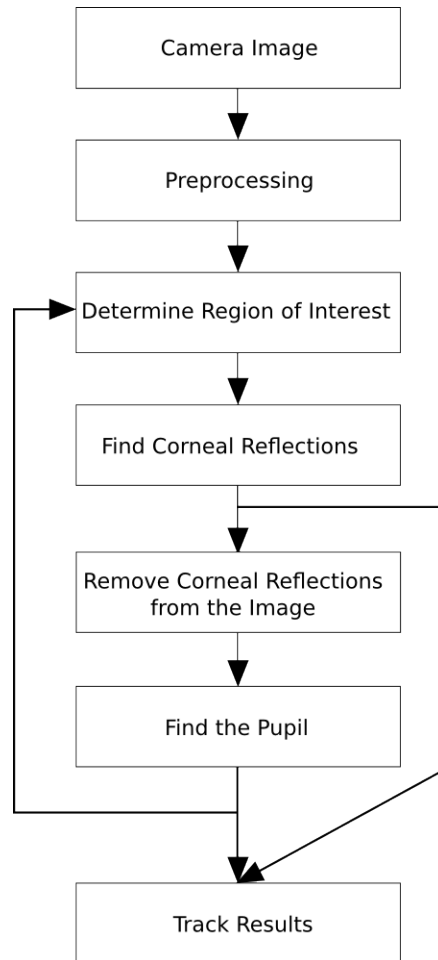


Figure 6.1: Workflow of the presented feature detection. The region of interest is determined using simple heuristics. CRs are searched inside this area and the CRs are removed from the original image. The position of the pupil is estimated. The results are reutilised in the analysis of subsequent frames.



Figure 6.2: Preprocessing the eye image. Figure (a) shows the original image. The image shows parts of the gaze tracking glasses. Figure (b) shows the preprocessed image. Preprocessing reduces the size of an image by selecting only the area of interest. In addition, noise is reduced by Gaussian filtering.

of the analysis algorithm. The ROI is found using the Starburst algorithm [16], which has been developed for finding the pupil contour. The algorithm itself is useful, but inadequate. The algorithm requires manual tuning of parameters, which is not an option when the algorithm should automatically adapt to the environment. The algorithm is modified to support automatic parameter selection in this work.

6.2.1 The Starburst Algorithm

The Starburst algorithm utilises the gradient of the pupil contour. Let us start with a simple case where the approximate position of the pupil is known. If we travel to any direction from a starting point inside the pupil, a significant change from black to white will be detected on the pupil contour. This is illustrated in Figure 6.3a. [16]

The approximate position of the pupil is not known while determining the ROI for the first time. Figure 6.3b shows the behaviour of the algorithm when the starting point is selected outside the pupil. Note that the algorithm tracks transitions from black to white and thus a ray can pass through the pupil. The accuracy of the algorithm is inadequate because only few points are detected on the pupil contour. [16]

Nevertheless, the algorithm has an interesting feature. If even a single pupil contour point is detected, reapplying the algorithm to that particular point will find several other pupil contour points (see Figure 6.3c). On the other hand, if the algorithm is applied to a point that is not on the pupil contour, only few points will be detected (see Figure 6.3d). [16]

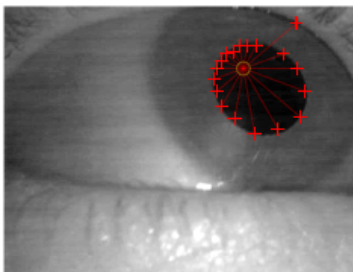
Initial contour points reside on one side of the pupil. Therefore, the remaining contour points must reside on the the other side of the pupil contour. While reapplying

the algorithm to an initial contour point, the gradient changes are tracked only between ± 50 degrees around the vector from the initial contour point towards the starting point. This idea is illustrated in Figures 6.3c and 6.3d. [16]

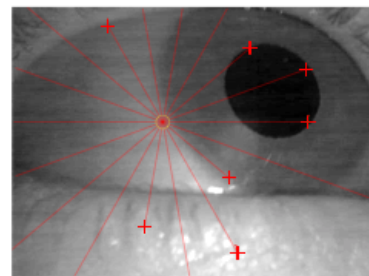
Because the result of the algorithm depends on the starting point, the algorithm is iterated. The iteration cycle includes two phases: Seeking gradient changes radially away from the starting point (1) and seeking gradient changes towards the starting point from the points detected in the first phase (2). After each cycle the centre of the ROI is estimated by computing the centroid of all detected contour points. The centre of the ROI is used as a starting point in the following iteration cycle. The iteration is finished in three cases [16]:

1. The previous and current centroids differ less than 10 pixels.
2. Convergence does not occur in 10 iteration cycles.
3. Too few detected significant gradient changes.

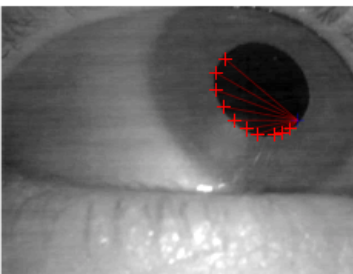
A list of the detected contour points is sustained for later use. In addition, outlier points are removed from the list. A point is considered an outlier if its distance from the estimated centroid is greater than 1.5 times the standard deviation of the points. [16]



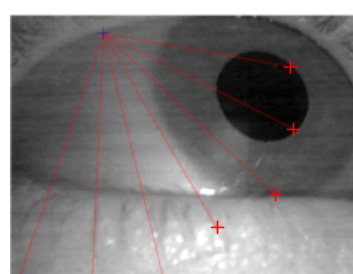
(a) The starting point is inside the pupil



(b) The starting point is outside the pupil



(c) Reapplying the Starburst algorithm for a contour point



(d) Reapplying the Starburst algorithm for a point outside the pupil

Figure 6.3: The Starburst algorithm. The red points mark the starting point and the red crosses mark the detected contour points. [35, edited]

6.2.2 Determining the Edge Threshold

Although the Starburst algorithm finds the pupil reliably, the algorithm is very sensitive to bad edge threshold values. The threshold is used to determine whether the gradient is large enough to be considered a potential contour point. If the threshold is too low, the Starburst algorithm will not detect only the edge points of the pupil, but also the iris and other features. On the other hand, if the threshold is too high, only few points will be found. [16]

The original work [16] uses a predefined estimate for the edge threshold. The method tries to find contour points using the threshold and if a sufficient amount of contour points is not available, the threshold is decremented by one. Although this approach can be considered dynamic with respect to the image quality, it is sensitive to bad values. The goodness largely depends on the quality of the image, the test subject, lighting and other factors. [16]

In order to adapt the Starburst algorithm to varying environments, the edge threshold must be determined dynamically from the image. There is no direct way to compute the threshold, but the best threshold can be estimated using a simple procedure. If the goodness of an edge threshold can be measured, it is straightforward to test different thresholds. It is not necessary to repeat the threshold estimation for each image as the lighting of the environment stays relatively constant. If the Starburst algorithm fails to find the ROI, the edge threshold is re-estimated.

This work presents a novel approach for finding the optimal threshold. It was noted that the Starburst algorithm finds several outlier points if too small threshold values are used, and hence the variance of the edge point coordinates can be used as an indicator for the goodness of the threshold. If the variance is small, the points are distributed in a small area, which is common for the pupil. If the variance is large, the edge points are distributed in a large area indicating that the current threshold is inadequate.

A similar approach has been applied in [17] where different thresholds were tested and the error from ellipse fitting was used as the goodness measure of the threshold. However, this approach may falsely select a threshold that is more sensitive to other circular shapes (e.g. the iris) in the image.

6.2.3 Recovering from Errors

If an error occurs while determining the ROI, the current frame is dropped and a reset operation is initiated. The reset operation resets the starting point to the centre of the image and initiates re-estimation of the edge threshold. The reset is initiated also if the pupil contour is not available (refer to Section 6.4.5 for more information).

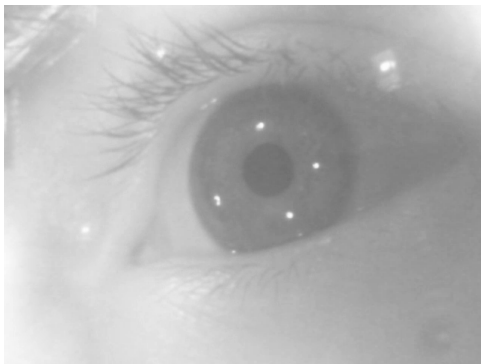
6.3 Finding Corneal Reflections

Many gaze tracking methodologies utilise CRs. If direct mapping techniques are used, the CRs are used to reduce the effect of head movements [16]. If model-based algorithms are used, locating the CRs is necessary for computing the centre of the cornea [20, 36]. The CRs are seen as bright points on the pupil and the iris. Figure 6.4a shows an image of the eye where four CRs are clearly observable.

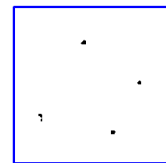
In order to simplify the analysis, the image is thresholded. This operation categorises pixels into black or white pixels according to their intensities. All values below a given threshold are changed to 0 (black) and all values above the threshold are changed to 255 (white) [26]. If the threshold is appropriately selected, the CRs will appear as bright pixels whereas the image mainly remains black. In this work, the threshold is regarded as a constant coefficient dependent on the gaze tracking glasses. Figure 6.4b shows the image after thresholding. Note that the colours in the figure are inverted.

After thresholding, potential CRs are determined by finding continuous bright areas from each row. The potential CRs are compared to a CR template and the differences are used for determining the goodnesses of the areas. The error and location of each area is stored.

The potential CRs are sorted according to their goodnesses. The error of each CR candidate is compared to the error of the best fitting candidate. The candidate is discarded if the error of the candidate is significantly higher than the error of the best fitting CR candidate. The intensity weighted centroids of the valid candidates are computed and the centroids are added to the list of CRs.



(a) The original image



(b) The thresholded image

Figure 6.4: Thresholding the eye image to obtain CRs. Figure (a) shows the original image. Figure (b) shows the image after thresholding. The colours in the thresholded image are inverted.

6.4 Finding the Pupil

Despite the fact that the Starburst algorithm finds the ROI, the method cannot be applied to determining the shape of the pupil. The main limitation comes from the eyelashes. If the eye is even partly closed, the Starburst algorithm will find points that are part of the eyelashes. If these points are used in ellipse fitting, the estimate of the pupil contour is distorted.

The results must be refined in a second step. The Starburst algorithm can be applied twice for a limited number of points in order to find a better estimate for the pupil contour [37]. However, this approach does not define a scheme for validating whether the contour points actually are from the pupil contour and hence this approach is not utilised in this work.

This section describes the developed method for locating the pupil. The method consists of four phases: thresholding the image (1), finding pupil candidates from the image (2), fitting an ellipse to each candidate (3) and selecting the pupil from the candidates (4).

6.4.1 Determining the Optimal Threshold for the Pupil Image

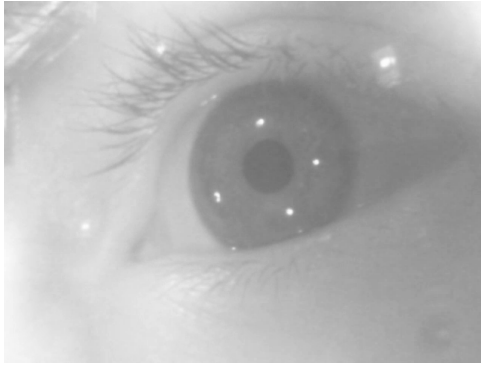
Thresholding is useful for finding the pupil. If the threshold is appropriately selected, the pupil will appear black in the thresholded image whereas other parts of the image will appear white.

Selecting the correct threshold is vital. If the threshold is too high, the iris and other artefacts will be observed as parts of the pupil. If the threshold is too low, the pupil becomes fragmented. A good estimate for the threshold is obtained using the results from the Starburst algorithm. The Starburst algorithm gives potential pupil contour points. Although not all points are on the pupil contour, most points are. Therefore, the threshold is determined by computing the average intensity value. Figure 6.5 demonstrates this procedure.

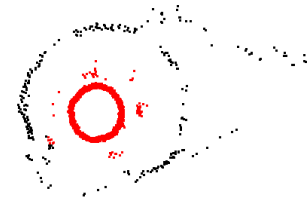
6.4.2 Clustering the Data and Finding the Pupil Contour Points

Providing that a good threshold has been selected, all groups of black pixels form clusters, each of which is regarded as a candidate for the pupil. The clusters are commonly presented using their contours, because analysis of the shape is easier in this format. The contours are searched from the thresholded image using the Teh-Chin algorithm [26, 38]. The algorithm searches the contours and creates a list of contour points for each contour.

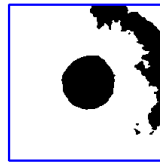
After the contours are available, the holes inside contours are filled in order to remove holes caused by CRs. Clearly spurious clusters are removed and neither too large



(a) The original image



(b) The Starburst points



(c) The thresholded image

Figure 6.5: Thresholding the eye image to locate the pupil. Figure (a) shows the original image. Figure (b) shows the contour points that are found using the Starburst algorithm. The red pixels are inliers and are used to compute the threshold whereas black pixels are outliers. Figure (c) shows the thresholded image.

nor too small clusters are accepted. These limiting factors depend upon the camera position.

6.4.3 Ellipse Fitting

The pupil can be modelled as a circle. However, the camera does not point directly towards the eye making the pupil appear as an ellipse in the camera image. In addition, the ellipse formulation of a pupil is useful for estimating the actual centre of the pupil (see Section 4.2.2). Ellipse fitting is performed using the method of least squares [26, 39].

If the pupil is clearly visible, ellipse fitting will provide an accurate estimate of the pupil contour. However, usually the eyelids cover part of the pupil making the ellipse fit poorly. Therefore, fitting the ellipse only to the points on the sides of the cluster has been shown to provide more accurate results [40].

Ellipse fitting is applied for the clusters in two stages. The first ellipse fitting utilises contour points from all directions. The fitted ellipse is used for selecting proper contour points of the cluster for a second ellipse fitting. The contour points on both sides of the candidate should reside near the contour of the fitted ellipse. If a sufficient amount of pupil points are available on the sides, the points are used for the second ellipse fitting. Otherwise the candidate is discarded. [40]

6.4.4 Selecting the Pupil from the Clusters

Thresholding, clustering and double ellipse fitting removes the worst pupil candidates, after which several good candidates might remain. In order to select the correct candidate, two properties of the data is considered. First, the surroundings of the pupil is bright. Second, the size of the pupil remains approximately constant between successive frames.

6.4.4.1 Computing Errors for Pupil Candidates

It is possible to construct a template of the pupil, where the shape of the pupil is taken from the ellipse fit. In the template image the pupil appears black and the surrounding of the pupil appears white.

The template image is used for creating a difference image where the intensity values of the original image are subtracted from the template image. If the current pupil candidate is correct, the intensity values of the difference image are small. If the surrounding is not white or the pupil candidate is not black, the intensity values of the difference image increase. The error per pixel is computed as:

$$E = \frac{1}{X \cdot Y} \sum_{y=0}^Y \sum_{x=0}^X |\text{img}_i(x, y) - \text{img}_o(x, y)|, \quad (6.1)$$

where X denotes the width of the image, Y the height of the image, img_i the template image and img_o the original image.

6.4.4.2 Verifying the Consistency of the Pupil Candidates

The goodness of a pupil candidate can be examined by comparing the ellipse of the candidate with the five previous pupil ellipses. If three of the previous pupil ellipses are inconsistent with the current candidate, the candidate is rejected. The current pupil ellipse and the i th previous pupil ellipse are considered consistent if the following equations hold [41]

$$l - l_i < l_{diff} \quad (6.2)$$

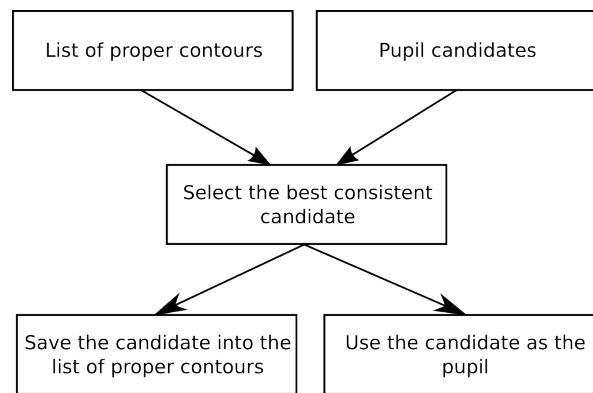
$$r - r_i < r_{diff}, \quad (6.3)$$

where l denotes the major axis of the current pupil candidate, l_i denotes the major axis of the i th previous ellipse, r represents the ratio between the major and the minor axes and r_i denotes the ratio for the i th previous pupil ellipse. r_{diff} and l_{diff} denote the maximal differences between the ellipses. [41]

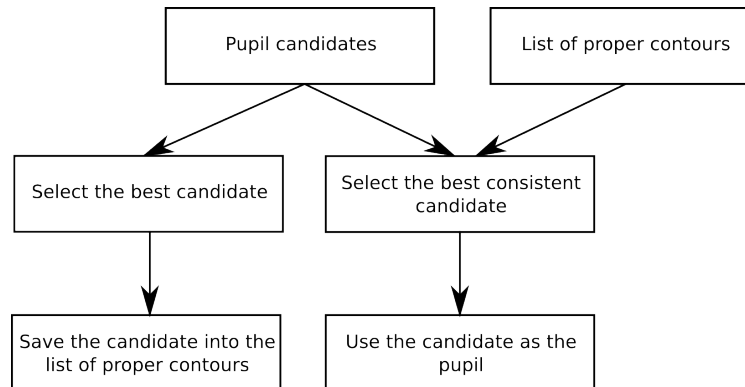
Tests have shown that good values for r_{diff} and l_{diff} are 0.05 and $L/120$, respectively. The parameter L denotes the width of the eye image. [41]

6.4.4.3 An Improved Method for Verifying the Consistency of the Pupil Candidates

Although the method for determining the consistency provides an adequate and reasonable basis, it is not directly applicable for gaze tracking purposes. If the camera is positioned near the eye the minor axis of the real pupil varies depending on the gaze target. Therefore, the minor axis is not a good feature for selecting the correct candidate and only the condition $l - l_i > l_{diff}$ is utilised in this work.



(a) The original method



(b) The improved method

Figure 6.6: Comparison between different methods used for verifying the consistency of the pupil cluster. Figure (a) shows the original method [41] and Figure (b) shows the improved method.

The original method [41] always stores the selected pupil candidate, which makes the algorithm vulnerable against errors. If a wrong candidate has been selected and that candidate is always preferred, the method will never select the correct candidate. In order to increase the robustness of the algorithm, this work presents a novel approach to the problem. The difference between the original and the novel methods is illustrated in Figure 6.6.

It is reasonable to assume that the correct pupil candidate has the smallest error most of the time. Therefore, the best candidate is always stored and the candidate which is consistent with the stored samples is selected. This modification improves the ability to recover. Finally, if no consistent candidates are available, the best cluster is selected in order to improve adaptiveness.

If no clusters are available, the frame is dropped. In addition, the information related to the last verification frame is removed.

6.4.5 Using the Previous Results for Determining the Region of Interest

Pupil tracking is the final result from the image analysis pipeline. This information is utilised in the next iteration as a starting point for ROI-determination.

The algorithms presented here are based on heuristics and knowledge about the analysis device and the structure of the eye. The method consists of multiple algorithms and issues will likely occur without sanity checks. If the analysis results are faulty in a single step, the subsequent parts of the analysis will fail. Thus, errors should be detected and recovery should be performed without any supervision.

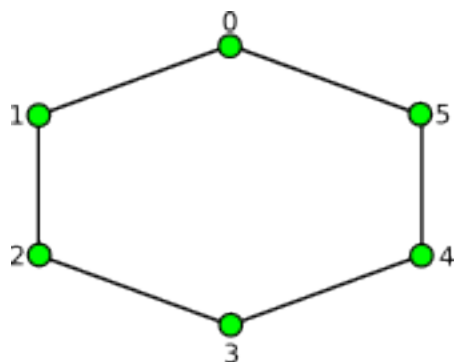
The first step of the method is finding the ROI. If this step fails to provide the correct ROI, or the Starburst algorithm does not provide enough valid points from the pupil contour, the pupil cannot be extracted from the image. The issue is solved by resetting the estimation of the ROI if the pupil is unavailable. However, the solution is not ideal, because the pupil is not always present in the image due to eyeblinks. Therefore, the algorithm is reset if no proper pupil has been detected in 10 frames. The estimated ROI is used as a starting point for the next iteration.

6.5 Assigning the Observed Corneal Reflections to Their Corresponding Light Sources

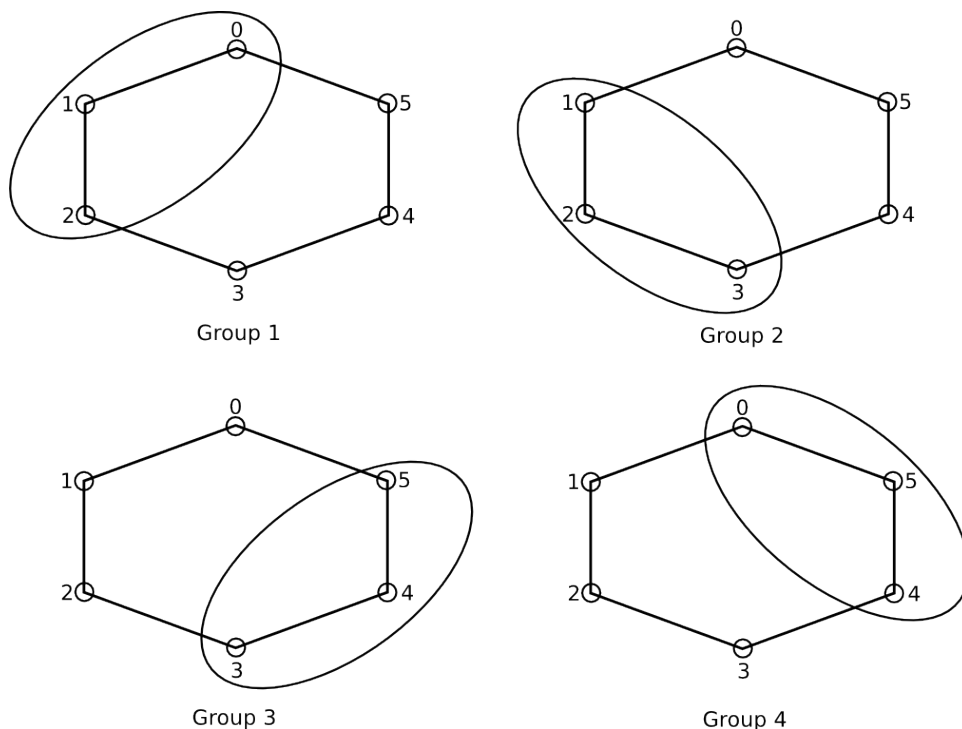
Section 4.2.1 describes a method for determining the coordinate of the centre of the cornea using several light sources. In order to estimate the centre of the cornea at least two CRs must be extracted from the image. If only two light sources are used, the relative order of the light sources from left to right is the same. Therefore, it is trivial to identify which light source causes a particular CR in the image.

The location of the pupil varies depending on the direction of gaze and hence it is not possible to position two light sources whose CRs are always visible to the camera. Therefore, more light sources should be present. Adding light sources increases the difficulty of assigning CRs to their corresponding light sources. For example, if three light sources placed on the same line and only two CRs are seen, assigning the observed CRs to their corresponding light sources is nontrivial.

Currently, six light sources are utilised to create a pattern on the cornea. The pattern is illustrated in Figure 6.7a. Figure 6.9 illustrates the whole pipeline of assigning the observed CRs to light sources in the shape.



(a) The numbering of the CRs



(b) The grouping of CRs

Figure 6.7: Positions of the CRs. Figure (a) shows the numbering of the CRs. Figure (b) shows four different groups to which the CRs are divided.

The pattern can be divided into four groups (Figure 6.7b). If the analysis equipment is constructed such that the centre of the pupil is always inside the bounding rectangle of the pattern in the eye image, the position of the pupil can be used for assigning the observed CRs to the groups. The heuristics for assigning the CRs into groups is presented in Table 6.1.

Table 6.1: The initial grouping of the observed CRs. The table shows the group to which a CR is assigned given its relative the position to the pupil.

	Left to the Pupil	Right to the Pupil
Above the Pupil	1	4
Below the Pupil	2	3

The grouping gives an initial estimate for the positions of the observed CRs and further analysis must be performed. The observed CRs are sorted in counterclockwise order inside the groups. All possible configurations are formed using two rules: (1) each CR must remain in its assigned group, (2) the order of the CRs must remain the same.

In order to select the best configuration from all possible configurations, it is necessary to define a goodness measure. This work utilises the angles between the CRs (see Figure 6.8 for illustration). For each generated configuration the angles between the CRs have some expected values. Given that a tested configuration is correct, the angles between the observed CRs should match the expected ones.

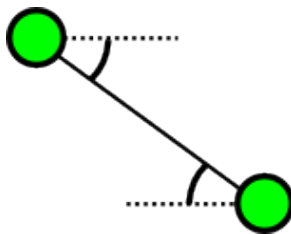
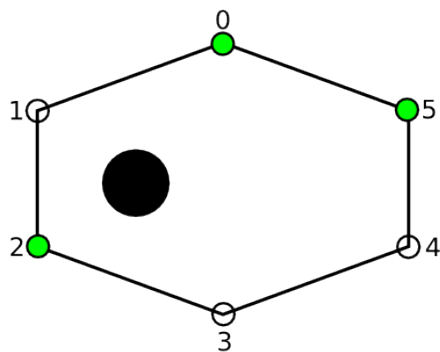


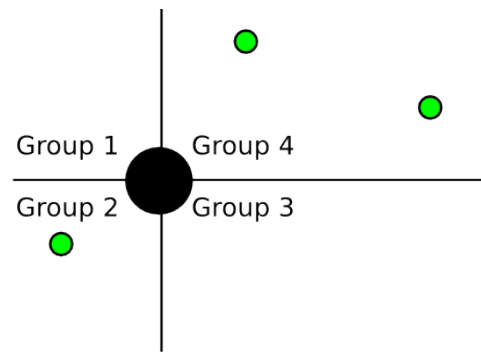
Figure 6.8: Determining the angle between two CRs.

6.6 Implementation

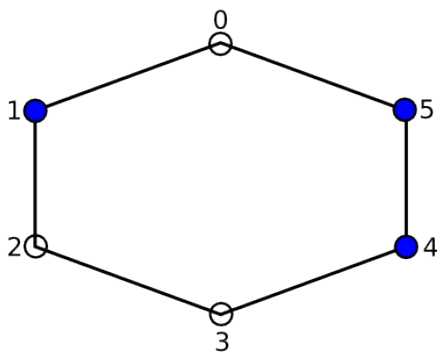
The algorithms are implemented in C++ for its efficiency. The implementation uses the OpenCV 2.3.1 library [26], which offers optimised implementations of commonly used computer vision algorithms (e.g. the TC89 contour detection algorithm, ellipse fitting, camera calibration, etc.).



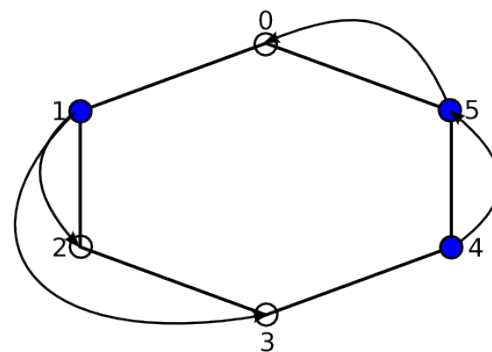
1) The real CRs. The coordinates of the CRs are known, however, there is no way to identify which CRs in the shape are observed.



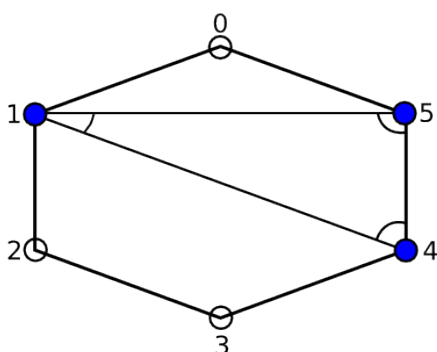
2) Grouping the CRs using the centre of the pupil.



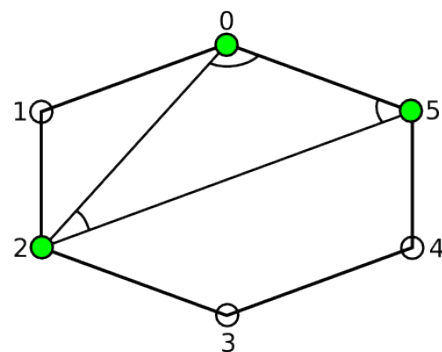
3) Initial configuration. The observed CRs are placed at the beginning of their groups.



4) Forming all possible configurations given that:
 - No CR can jump over another
 - Each CR must remain inside its group



5) Comparing the angles between the theoretic angles of each possible configuration with the angles observed in the image.



6) Select the shape whose theoretic angles matches best the real angles

Figure 6.9: Summary of assigning the observed reflections to the light sources.

Chapter 7

Validating the Developed Tracking Algorithms

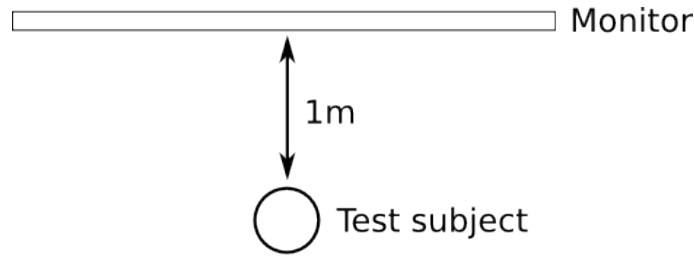
The developed method for recognising the necessary features is divided into two parts which are tested separately. The first part, feature detection, recognises the pupil contour and the CRs from the eye image. The second part, the assignment of the CRs, assigns the CRs into their corresponding light sources. The second part can be tested automatically as all possible configurations can be explored. However, creating a realistic simulation of an eye and an environment is not practical. Therefore, the first part is validated with measured data.

This chapter is divided into two sections. First, the measured data is used for validating the feature detection. The analysed data is visually inspected to verify the goodness of the developed method. Second, the assignment of the CRs is validated with automated tests.

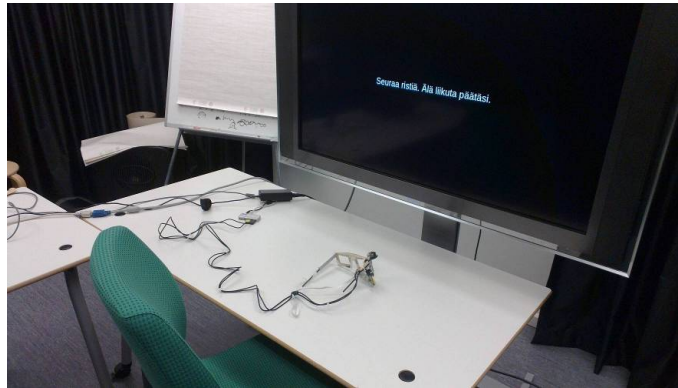
7.1 Experimental Setup for Validating the Feature Detection

The objective of the experiment is to explore the behaviour of the algorithm in different conditions likely to arise in mobile environments. The pupil size may vary depending on the lighting and the position of the pupil alters depending on the gaze target. Depending on the pupil positions, a different number of CRs is detected.

The experimental setup is illustrated in Figures 7.1a and 7.1b. The test subject is asked to look at different target points in the scene without moving the head. This task tests how the pupil position and the number of observed CRs affects the precision of the algorithm. The test is repeated using different lighting conditions to test the effect of varying pupil diameter and changing dynamics of the data. The effect of facial features is investigated by repeating the experiment for 6 healthy voluntary test subjects.



(a) The schematic setup



(b) A photograph of the experiment setup

Figure 7.1: The experimental setup. (a) The schematic setup. (b) A photograph of the experiment setup. The user is positioned approximately 1m away from a large monitor. The user looks at markers shown in different places on the monitor.

The experiment is performed using the analysis equipment described in Chapter 5. The recording resolution of the eye camera is 640x360. The analysis device is configured for a single person whose results are discarded from the analysis results. No further fine tuning or changes are made during the experiment.

7.2 Inspecting the Goodness of an Algorithm

The recorded data is analysed with the method developed in this thesis. The validation is accomplished by comparing the analysis results to the golden standard obtained by inspecting the data visually. The developed method consists of multiple algorithms and their performances are investigated separately.

Commonly, the performance of a gaze tracking system is studied with respect to the error in degrees of visual angle [11, 40]. However, the developed method is related strictly to computer vision in gaze tracking. Therefore, it is more convenient to study the performance of each algorithm with respect to *sensitivity* and *precision*.

In our context the sensitivity is defined as [42]:

$$\textit{Sensitivity} = P(\textit{feature detected correctly} \mid \textit{feature exists}). \quad (7.1)$$

In other words, the sensitivity describes the probability that the algorithm is able to find the correct feature from the image given that the feature exists. The precision describes the probability that the detected feature is caused by an actual feature in the image [42]:

$$\textit{Precision} = P(\textit{feature exists} \mid \textit{feature detected}). \quad (7.2)$$

The presented measures should give a good understanding of the goodness of the algorithm. If the sensitivity is poor, the algorithm is unable to detect the feature correctly. On the other hand, if the algorithm finds the feature correctly from all frames, the sensitivity is high. If the precision is poor, the algorithm produces false positives.

7.3 Results of Feature Detection

Although the algorithms are designed to be adaptive with respect to the facial features of a user, the analysis equipment must be calibrated. The equipment was calibrated by selecting the best values for a single user. The selected parameter values are presented in Appendix A.

The calibration phase revealed two issues with the analysis equipment. First, the infrared light from the environment disturbed the measurements. Although the gaze tracking glasses are equipped with a hot mirror to separate the environmental lighting from the internal lighting, the mirror does not reflect all infrared light. Because some infrared light passes through the hot mirror, the pupil becomes less distinguishable. This issue is illustrated in Figure 7.2a.

The second issue is related to a low framerate and a relatively long exposure time of the camera. Each saccade distorts 1 – 3 image frames. This issue is illustrated in Figure 7.2b.

After measurements 15000 image frames (500 seconds of video data) were available. The first issue was compensated by selecting only frames where the environmental lighting did not distort to the image. The second issue was compensated by removing the distorted frames manually. After removing the distorted frames, 10400 image frames were available for the analysis (approximately 70% of the frames).

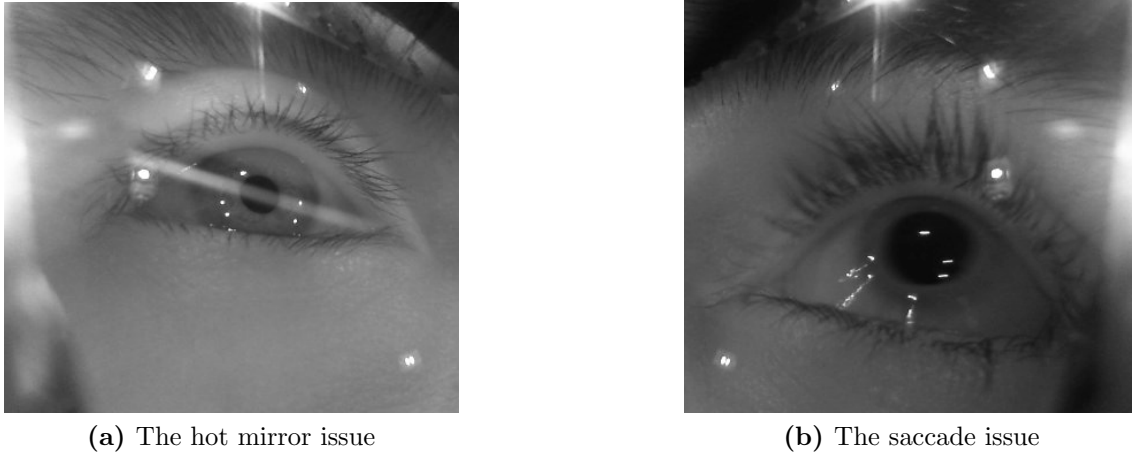


Figure 7.2: Issues while finding the pupil from an image. Figure (a) shows issues related to the analysis equipment. The hot mirror passes little infrared light, which can be observed in the image. Figure (b) shows distortions caused by saccades: The camera captures distorted images due to the low framerate and the long exposure time. The distortions are seen clearly in the CRs.

7.3.1 Estimating the Region of Interest

The estimation of the ROI performed excellently for all test subjects. The lighting conditions and the pupil position did not affect the performance of finding the ROI.

However, facial features seem to have a minor effect on the performance. For a single test subject, the ROI was estimated around the eyelashes in 30 adjacent frames out of 2814 frames. In other words, the algorithm failed for approximately 1.1% of the frames. The issue is illustrated in Figure 7.3. The ROI estimation was successful for other test subjects.

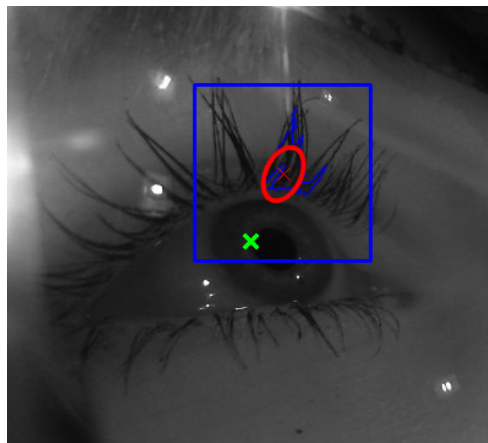


Figure 7.3: An issue while finding the ROI. The figure shows that the ROI may be selected incorrectly if the eyelashes are dark.

7.3.2 Finding the Pupil Contour

The sensitivity of finding the pupil contour was adequate on average: The pupil contour was estimated for 97% of the frames where the pupil was available. For the remaining 3% of the frames the pupil was not detected by the algorithm.

Table 7.1 shows the sensitivity of the algorithm for each test subject. The results show that individual differences affect the results. For test subjects 2 and 4 the increased lighting decreases the sensitivity of the algorithm significantly. For other test subjects the increased lighting has no similar strong effect.

The precision of the algorithm was also high. On average 98% of the contour estimates were correct. In most of the remaining cases the algorithm chose a wrong

Table 7.1: Sensitivity of pupil detection. Dashes indicate that the data is distorted by the environment lighting. The bolded result denote the average performance and the standard deviation of the results.

Subject	No Light	Minimal Light	Medium Light	Bright Light	Average	SD
Subject 1	1.00	1.00	1.00	1.00	1.00	0.00
Subject 2	0.97	0.94	0.92	0.89	0.93	0.03
Subject 3	0.96	0.95	0.96	0.99	0.97	0.02
Subject 4	1.00	0.99	0.93	–	0.97	0.04
Subject 5	1.00	1.00	–	–	1.00	0.00
Average	0.98	0.97	0.95	0.96	0.97	
SD	0.02	0.03	0.04	0.06	0.03	

Table 7.2: Precision of pupil detection. Dashes indicate that the data is distorted by the environment lighting. The bolded result denote the average performance and the standard deviation of the results.

Subject	No Light	Minimal Light	Medium Light	Bright Light	Average	SD
Subject 1	1.00	1.00	0.99	0.99	1.00	0.01
Subject 2	0.98	0.99	0.95	0.95	0.97	0.02
Subject 3	0.98	0.98	0.97	1.00	0.98	0.01
Subject 4	1.00	1.00	0.97	–	0.99	0.02
Subject 5	1.00	1.00	–	–	1.00	0.00
Average	0.99	0.99	0.97	0.98	0.99	
SD	0.01	0.01	0.02	0.03	0.02	

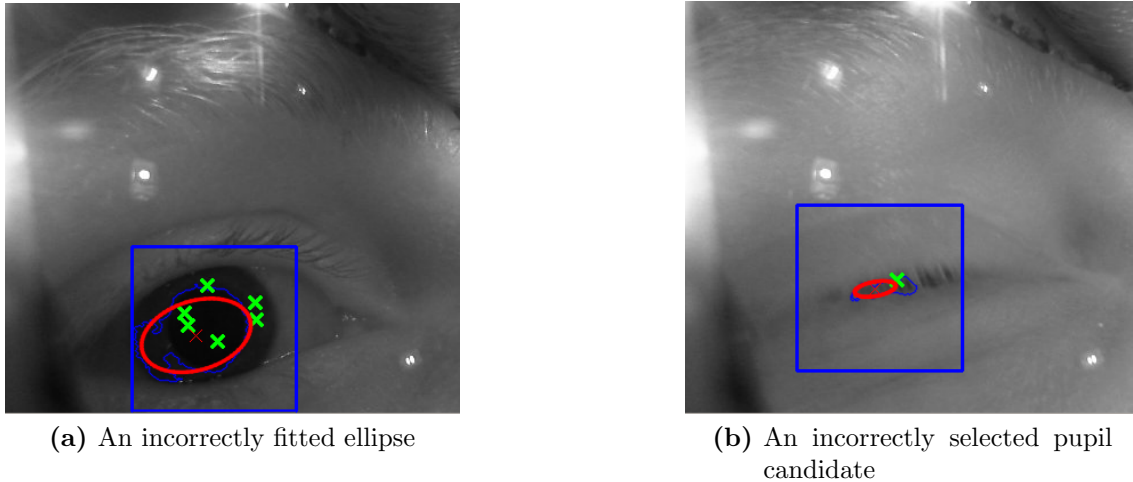


Figure 7.4: Issues with pupil detection. Figure (a) shows an incorrectly estimated pupil ellipse. Figure (b) shows that eye blinks cause an incorrect contour to be regarded as a pupil contour.

cluster or estimated the pupil contour incorrectly (see Figure 7.4a for illustration). The pupil was detected falsely in 18 frames where the pupil was not visually observable (see Figure 7.4b). The total number of frames where the pupil was not visually observable was 20.

The effect of facial features and lighting on the precision is presented in Table 7.2. The results indicate that increased lighting and facial features only has minor effect on the precision.

The position of the pupil in the image had no effect on the accuracy or the performance of the algorithm. Changes in the pupil diameter due to changed lighting conditions caused no changes in the behaviour of the algorithm.

7.3.3 Locating Corneal Reflections

The gaze tracking glasses were equipped with six LEDs. Throughout the analysis the maximum number of searched CRs was six. However, the number of CRs in the image was dependent on the direction of gaze and the algorithm was assumed to detect only the true CRs.

The sensitivity of finding a CR is shown in Table 7.3. The table summarises the results with respect to the lighting conditions and test subjects. The results show that the algorithm finds the CRs adequately in all lighting conditions. In the worst case the sensitivity was 85% and in the best case the sensitivity was 99%. The average sensitivity was 94%. The individual differences appear to have a minor effect on the sensitivity.

Table 7.4 shows the precision of the algorithm. The precision of the algorithm

is highly sensitive to individual differences. For some test subjects the CRs were incorrectly detected at the border between the sclera and the cornea. False positives occurred also in the border between the sclera and the skin. The precision of the algorithm is very high for test subject 3 (the average precision is 98%) whereas the average precision is only 76% for test subject 5. The increased lighting seems to affect the results for some test subjects.

Table 7.3: Sensitivity of detecting CRs. Dashes indicate that the data is distorted by the environment lighting. The bolded result denote the average performance and the standard deviation of the results.

Subject	No Light	Minimal Light	Medium Light	Bright Light	Average	SD
Subject 1	0.98	0.93	0.94	0.99	0.96	0.03
Subject 2	0.85	0.91	0.97	0.89	0.91	0.05
Subject 3	0.96	0.91	0.99	0.99	0.96	0.04
Subject 4	0.98	0.99	0.97	–	0.98	0.01
Subject 5	0.94	0.99	–	–	0.96	0.04
Average	0.94	0.95	0.97	0.96	0.95	
SD	0.05	0.04	0.02	0.06	0.04	

Table 7.4: Precision of detecting CRs. Dashes indicate that the data is distorted by the environment lighting. The bolded result denote the average performance and the standard deviation of the results.

Subject	No Light	Minimal Light	Medium Light	Bright Light	Average	SD
Subject 1	0.93	0.97	0.86	0.82	0.89	0.07
Subject 2	0.70	0.85	0.81	0.79	0.79	0.06
Subject 3	0.95	0.99	1.00	0.99	0.98	0.02
Subject 4	0.91	0.87	0.80	–	0.86	0.06
Subject 5	0.79	0.72	–	–	0.76	0.05
Average	0.86	0.88	0.87	0.87	0.87	
SD	0.11	0.11	0.09	0.11	0.09	

7.4 Testing Assignment of the Corneal Reflection with Unit Tests

The assignment of the CRs is validated using unit tests. The idea of unit testing is to automatically test the implemented algorithm by feeding certain inputs for the method and comparing the result of the algorithm to the desired outcome. A good set of unit tests explores all different code paths. Therefore, an understanding of the implementation is crucial.

Let us consider the necessary tests for the assignment of the CRs. Obviously, all different configurations of the observed CRs must be tested. However, it is neither practical nor necessary to test all possible pupil positions. The algorithm consists of two steps: (1) The initial grouping and (2) finding the best possible configuration given the initial groups. Only the first step is dependent on the position of the pupil. Hence, it is possible to define six pupil positions in which the end result can be different. These positions are illustrated in Figure 7.5.

The test results are promising, however, not all unit tests pass. The algorithm performs correctly for all cases when the number of observed CRs is at least three. When only two CRs are observed, the algorithm fails sometimes to correctly assign the CRs. The incorrectly assigned configurations are shown in Figure 7.6.

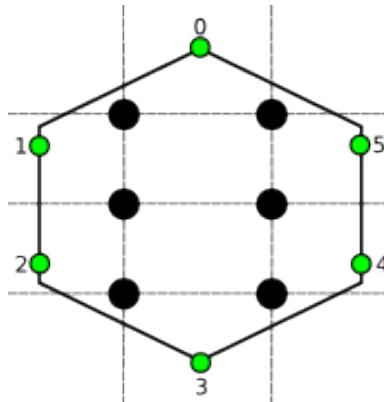


Figure 7.5: Designing the unit tests. The figure shows six pupil positions that define all possible outcomes of the first step of the algorithm.

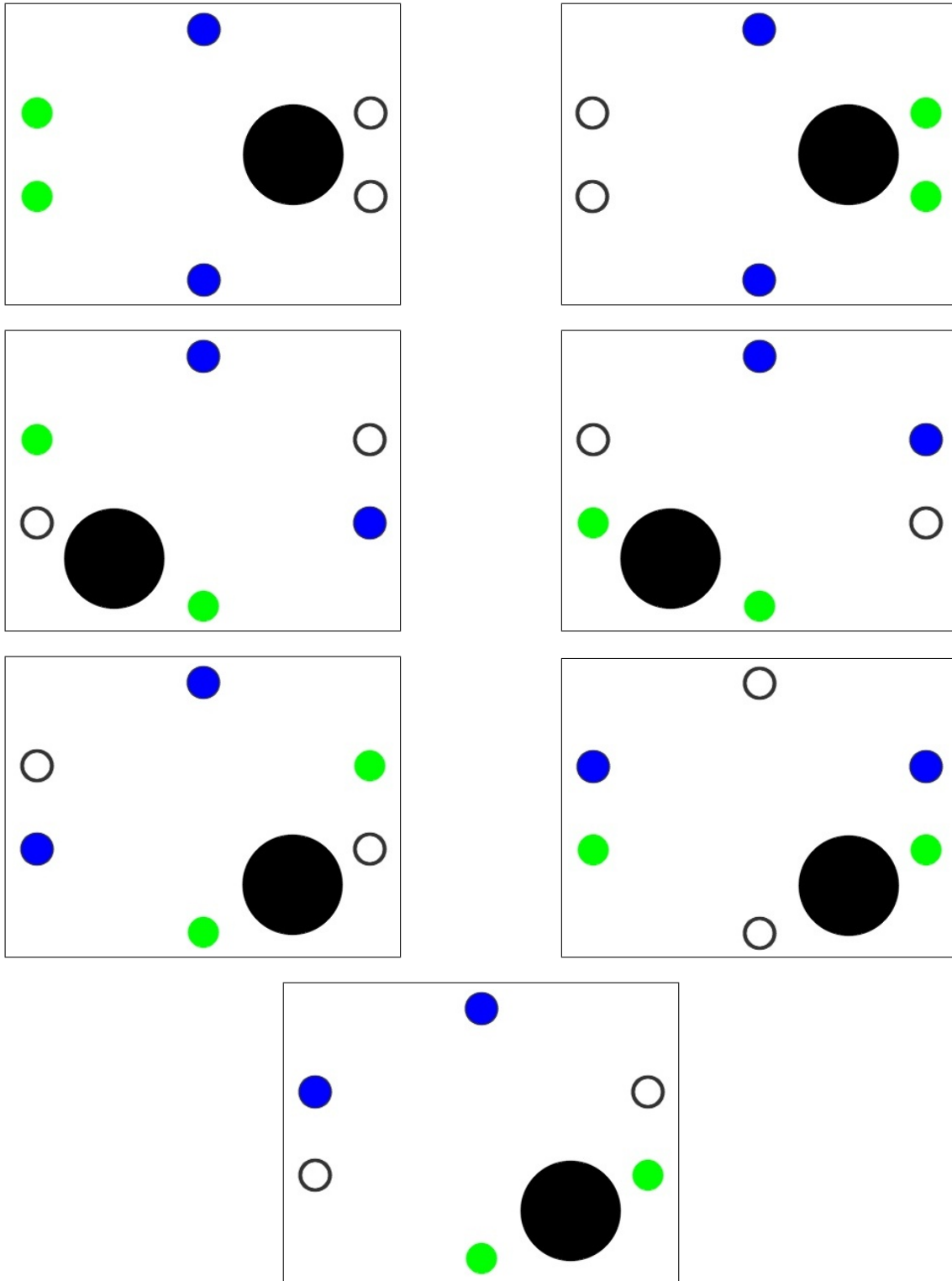


Figure 7.6: Erroneous assignments of CRs. The black sphere denotes the pupil. The green coloured spheres denote the real CRs and the blue coloured spheres denote the assignments of the algorithm.

Chapter 8

Discussion

This master's thesis describes the development of a mobile gaze tracking platform. The thesis overviews the necessary background of the human eye and computer vision. Chapter 4 utilised the GEM in model-based gaze tracking. Chapter 5 described the development of the gaze tracking glasses by utilising the information about the optical features of the pupil and the iris. A method for detecting features from the eye image was developed. Finally, the developed method was studied with the data from different test subjects in Chapter 7.

This chapter overviews the main results of the developed method. The chapter explores reasons for the performance and describes possible directions for future development. In addition, the chapter discusses the overall design of the system, limitations and potential applications of the system.

8.1 Discussion about the Developed Method

The developed method consists of four main parts: (1) The identification of the ROI, (2) the detection of the pupil, (3) finding the CRs and (4) identifying the CRs. The developed method was tested with experimental data from six subjects. The data from one of the test subjects was used for determining the optimal parameter values for the test equipment. The data from the remaining five participants were analysed with the developed method.

8.1.1 Finding the Region of Interest

The ROI was recognised correctly in almost all studied cases: For a single test subject the ROI was estimated around the eyelashes in 30 adjacent frames (one second) out of 2814 (approximately 94 seconds). However, the issue was temporary and the method was able to converge to the correct pupil immediately. This indicates, that dark eyelashes rarely cause issues and the algorithm can overcome the issues effectively.

The determination of the ROI is based on the robust Starburst algorithm [16]. However, the original algorithm was sensitive for the edge threshold value. This thesis presents a new method for estimating the optimal edge threshold value. According to the results, the method is precise and robust. The method makes the Starburst algorithm adaptive to different lighting conditions, facial differences and differences in image quality.

8.1.2 Finding the Pupil

The developed method can identify and estimate the pupil contour excellently. The sensitivity of pupil detection was 97%. In practise, this result means that if the pupil is visible in the image, the algorithm will correctly find the pupil with 97% probability. The identified pupil contour was correct with 98% probability.

The pupil contour was estimated incorrectly for some test subjects due to too small contrast between the iris and the pupil. This result can be described by individual differences in the pigmentation of the iris.

This thesis presented an improved method for validating the consistency of the pupil with respect to the previous frames. Compared to the original method [41] the presented approach improves the robustness and adaptiveness of the method.

8.1.3 Finding the Corneal Reflections

CRs are troublesome for the developed method. The sensitivity of finding CRs was high (94% on average), however, the precision was low (88% on average, 70% in the worst case). The individual differences appeared to affect the results greatly. For some test subjects the CRs were incorrectly located at the border between the sclera and the cornea. Also the border between the sclera and the skin caused false positives. Increasing lighting seemed to weaken the results, however, the effect of increased lighting is ambiguous.

The precision of detecting the CRs can be improved by reducing the number of searched CRs. Identifying two CRs is enough for estimating the centre of the cornea. CRs are searched by finding bright areas from the eye image and computing an error for these areas. Currently six areas with smallest error are regarded as CRs. The results indicate that the error of false positives are commonly higher than the error of the real CRs. Therefore, reducing the number of searched CRs reduce the number of false positives.

Despite this issue can be minimised by reducing the number of utilised CRs, the algorithm itself does not improve. Therefore, the correct solution would be to create a better algorithm. The current algorithm tests the potential CRs against a simple mask and selects the CRs with the smallest errors. One possible approach would be to utilise shape factors (e.g. circularity) to verify the goodness of the potential CRs.

8.1.4 Unit Test Results

Unit tests were used for validating the algorithm that assigns the detected CRs to the light sources. The algorithm performed adequately when the number of detected CRs was at least three. However, some unit tests failed when the number of CRs was two.

The reason for misassignment is caused by the error criterion that is utilised for selecting the correct configuration. The error criterion is based on the angles between the horizontal axis and the other CRs. Due to the geometry of the shape, some angles are identical. The incorrect assignments shown in Figure 7.6 are geometrically identical to the correct assignments.

The algorithm can be improved in many ways. It is possible to create a non-regular shape that has unique configurations for all cases. Another solution is to modify the error measure to utilise the distance between the observed CRs. However, the distance between the CRs is dependent on the position of the eye with respect to the gaze tracking glasses making this approach troublesome.

8.1.5 Limitations of the Developed Method

The presented method is able to find the ROI and the pupil from the eye image. However, the precision of finding the CRs is inadequate. The goodness of detecting the CRs is heavily dependent on the individual differences of the eye. In order to make the system work adequately, the number of searched CRs must be decreased.

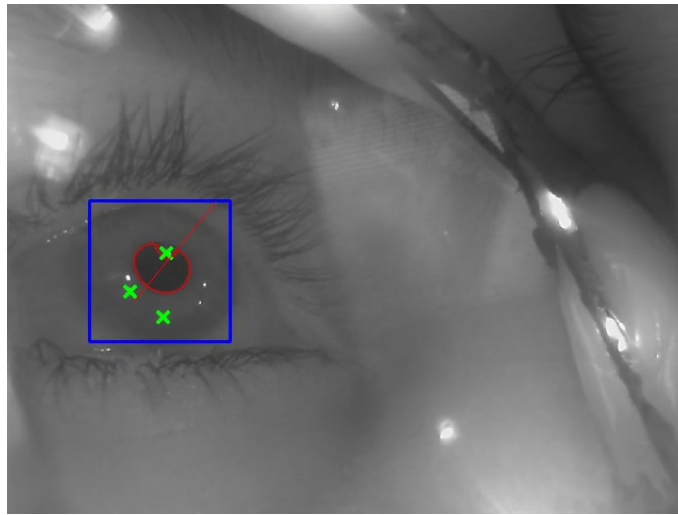
The algorithm for identifying the CRs require at least three CRs to work correctly. This issue must be considered while designing the gaze tracking glasses.

8.2 The Mobile Gaze Tracking System

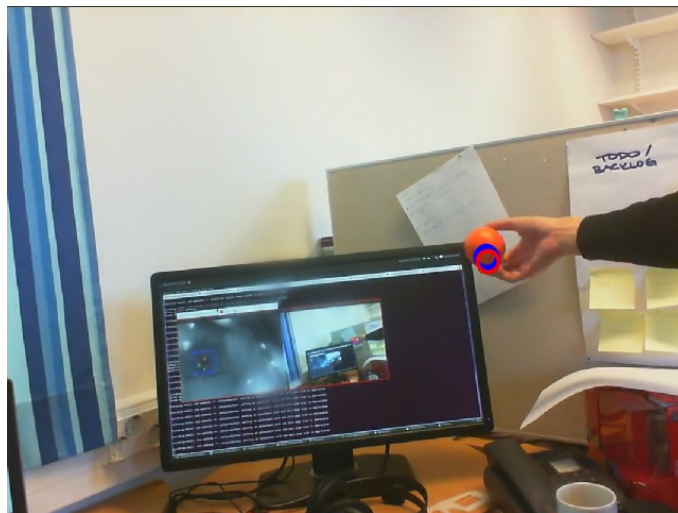
Despite the limitations of the developed method, the tests indicate that the method performs adequately if the number of searched CRs is decreased and the gaze tracking glasses are correctly configured. Therefore, the developed algorithm is sufficient for testing other parts of the gaze tracker.

For demonstration purposes, the developed method was combined with model-based gaze tracking described in Section 4.2. The parameter values for the model are taken from the GEM presented in Section 2.2. The gaze vector is mapped to the scene by assuming a constant depth of 1m for the gaze vector (see Section 4.2.4.2 for details). Neither calibration nor kappa correction are utilised. The eye and the scene images were recorded with the equipment presented in Chapter 5.

Figure 8.1a shows the eye image with the gaze vector. Figure 8.1b shows the point at which the gaze vector is mapped. The test subject was asked to look at the orange ball shown in the image. The overall system gives reasonable results.



(a) Determining the gaze vector



(b) Determining the point of gaze

Figure 8.1: Utilising model-based gaze tracking for retrieving the point of gaze in the scene image. Figure (a) shows the results after applying model-based gaze tracking for the eye image. The red line indicates the gaze vector from the centre of the cornea to the centre of the pupil. CRs used in the analysis are marked with green crosses and the pupil contour is marked in red. Figure (b) shows mapping of the gaze vector to the scene image. The red circle shows the estimated point of gaze. The blue circles show the 15 previous points of gaze.

Bibliography

- [1] J. Fukushima, K. Fukushima, T. Chiba, S. Tanaka, I. Yamashita, and M. Kato. Disturbances of voluntary control of saccadic eye movements in schizophrenic patients. *Biological Psychiatry*, 23(7):670 – 677, 1988.
- [2] W. A. Fletcher and J. A. Sharpe. Saccadic eye movement dysfunction in alzheimer’s disease. *Annals of Neurology*, 20(4):464–471, 1986.
- [3] O. Rascol, M. Clanet, J. L. Montastruc, M. Simonetta, M. J. Soulier-Esteve, B. Doyon, and A. Rascol. Abnormal ocular movements in parkinson’s disease. *Brain*, 112(5):1193–1214, 1989.
- [4] R. S. Rimmel. An inexpensive eye movement monitor using the scleral search coil technique. *Biomedical Engineering, IEEE Transactions on*, BME-31(4):388–390, april 1984.
- [5] R. J. K. Jacob and K. S. Karn. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *Work*, 2(3):573–605, 2003.
- [6] C. Ehmke and S. Wilson. Identifying web usability problems from eye-tracking data. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it - Volume 1*, BCS-HCI ’07, pages 119–128, Swinton, UK, UK, 2007. British Computer Society.
- [7] L. A. Granka, T. Joachims, and G. Gay. Eye-tracking analysis of user behavior in www search. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR ’04, pages 478–479, New York, NY, USA, 2004. ACM.
- [8] R. J. K. Jacob. Virtual environments and advanced interface design. In Woodrow Barfield and Thomas A. Furness, III, editors, *Virtual environments and advanced interface design*, chapter Eye tracking in advanced interface design, pages 258–288. Oxford University Press, Inc., New York, NY, USA, 1995.
- [9] H. Yanco. Wheellesley: A robotic wheelchair system: Indoor navigation and user interface. In Vibhu Mittal, Holly Yanco, John Aronis, and Richard Simpson, editors, *Assistive Technology and Artificial Intelligence*, volume 1458 of *Lecture*

- Notes in Computer Science*, pages 256–268. Springer Berlin / Heidelberg, 1998. 10.1007/BFb0055983.
- [10] Aalto University. aivoaalto. <http://www.aivoaalto.fi>, April 2012.
 - [11] D. Li, J. Babcock, and D. J. Parkhurst. openeyes : a low-cost head-mounted eye-tracking solution. *Computer*, page 7, 2002.
 - [12] V. Rantanen, T. Vanhala, O. Tuisku, P.-H. Niemenlehto, J. Verho, V. Surakka, M. Juhola, and J. Lekkala. A wearable, wireless gaze tracker with integrated selection command source for human-computer interaction. *IEEE transactions on information technology in biomedicine a publication of the IEEE Engineering in Medicine and Biology Society*, 15(5):795–801, 2011.
 - [13] Sensomotoric Instruments. Eye tracking glasses. <http://www.eyetracking-glasses.com>, March 2012.
 - [14] Tobii. Tobii glasses. <http://www.tobiiglasses.com/scientificresearch/>, March 2012.
 - [15] Sensomotoric Instruments. Demonstration of sensomotoric instruments eye tracking glasses. Presentation, November 2011.
 - [16] D. Winfield and D. J. Parkhurst. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR05 Workshops*, 3:79–79, 2005.
 - [17] W. J. Ryan, A. T. Duchowski, and S. T. Birchfield. Limbus / pupil switching for wearable eye tracking under variable lighting conditions. *Proceedings of the 2008 symposium on Eye tracking research applications ETRA 08*, pages 61–64, 2004.
 - [18] A. Meyer, M. Böhme, T. Martinetz, and E. Barth. A single-camera remote eye tracker. In Elisabeth André, Laila Dybkjær, Wolfgang Minker, Heiko Neumann, and Michael Weber, editors, *Perception and Interactive Technologies*, volume 4021 of *Lecture Notes in Computer Science*, pages 208–211. Springer Berlin / Heidelberg, 2006. 10.1007/11768029_25.
 - [19] D. Bäck. Neural network gaze tracking using web camera. Master’s thesis, Linköping University, 2006.
 - [20] C. Hennessey, B. Nouredin, and P. Lawrence. A single camera eye-gaze tracking system with free head motion. In *Proceedings of the 2006 symposium on Eye tracking research & applications, ETRA '06*, pages 87–94, New York, NY, USA, 2006. ACM.
 - [21] R. A. U, E. Y. Ng, and J.S. Suri. *Image modeling of the human eye*. Bioinformatics and biomedical imaging. Artech House, 2008.

- [22] M. Falhar. A theoretical model of the human eye based on ultrasound and corneal data. *Optica Applicata*, 39(1):195, 2009.
- [23] S. M. Hosseini, B. N. Araabi, and H. Soltanian-Zadeh. Pigment melanin: Pattern for iris recognition. *CoRR*, abs/0911.5462, 2009.
- [24] C. Morimoto. Pupil detection and tracking using multiple light sources. *Image and Vision Computing*, 18(4):331–335, March 2000.
- [25] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [26] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O’Reilly, Cambridge, MA, 2008.
- [27] D. C. Brown. Decentering distortion of lenses. *Photogrammetric Engineering*, 32(3):444–462, 1966.
- [28] D. M. Stampe. Heuristic filtering and reliable calibration methods for video-based pupil-tracking systems. *Time*, 25(2):137–142, 1993.
- [29] H.D. Young, R.A. Freedman, and L. Ford. *University physics*. Addison-Wesley series in physics. Addison-Wesley, 2006.
- [30] F. Schaeffel. Kappa and Hirschberg ratio measured with an automated video gaze tracker. *Optom Vis Sci*, 79:329–334, May 2002.
- [31] G. Daunys (SU), B. K. Ersbøll (DTU), M. Böhme (UzL), O. Stepankova (CTU), A. Villanueva (UPNA), E. Barth (UzL), M. Vester-Christensen (DTU), T. Delbruck (UNICH), D. Dervinis (SU), D. Droege (UNI KO-LD), M. Fejt (CTU), J. Fejtová (CTU), D. W. Hansen (ITU), L. K. Hansen (DTU), D. Leimberg (DTU), A. Meyer (UzL), T. Martinetz (UzL), N. Ramanauskas (SU), and V. Vysniauskas (SU). D5.2 report on new approaches to eye tracking. communication by gaze interaction (cogain), ist2003-511598: Deliverable 5.2, 2006.
- [32] G. Boening, K. Bartl, T. Dera, S. Bardins, E. Schneider, and T. Brandt. Mobile eye tracking as a basis for real-time control of a gaze driven head-mounted video camera. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, ETRA ’06, pages 56–56, New York, NY, USA, 2006. ACM.
- [33] A. Wilson. Cmos cameras find a niche. *Vision Systems Design*, pages 53–58, 2006.
- [34] European Parliament and the Council of the European Union. Directive 2006/25/EC of the European Parliament and of the Council of 5 April 2006 on the minimum health and safety requirements regarding the exposure of workers to risks arising from physical agents. *Official Journal of the European Union*, L 114:0038–0059, 2006.

- [35] D. Winfield and D. J. Parkhurst. Starburst: A robust algorithm for video-based eye tracking. Unpublished Manuscript, 2005.
- [36] S.-W. Shih and J. Liu. A novel approach to 3-d gaze tracking using stereo cameras. *IEEE Transactions on Syst. Man and Cybern., part B*, 34:234–245, 2004.
- [37] J. R. Wayne, L. W. Damon, T. D. Andrew, and T. B. Stan. Adapting starburst for elliptical iris segmentation.
- [38] C. H. Teh and R. T. Chin. On the detection of dominant points on digital curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11:859–872, August 1989.
- [39] R. Hal and J. Flusser. Numerically stable direct least squares fitting of ellipses. *The Sixth International Conference in Central Europe on Computer Graphics and Visualization*, 21(5):125–132, 1998.
- [40] O. Takehiko, M. Naoki, and Y. Atsushi. Freegaze: A gaze tracking system for everyday gaze interaction. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, ETRA '02, pages 125–132, New York, NY, USA, 2002. ACM.
- [41] M. Kiyama, H. Iyatomi, and K. Ogawa. Robust video-oculography for non-invasive autonomic nerve quantification. *Conf Proc IEEE Eng Med Biol Soc*, 2011:494–496, Aug 2011.
- [42] R.C. Eberhart and Y. Shi. *Computational Intelligence: Concepts to Implementations*. Elsevier Science, 2007.

Appendix A

Parameter Values Used to Validate the Method

Algorithm	Parameter name	Value
CR detection	CR_MAX_ERR_MULTIPLIER	0.14
	CR_THRESHOLD	135
	MAX_CR_WIDTH	5
	CR_MASK_LEN	15
ROI estimation	STARBURST_CIRCULAR_STEPS	50
	STARBURST_REQUIRED_ACCURACY	10
	STARBURST_MIN_FEATURES	45
	STARBURST_BLOCK_COUNT	1
Cluster validation	ROI_W_DEFAULT	145
	MIN_CLUSTER_SIZE	30
	MAX_CLUSTER_SIZE	800
	MIN_PUPIL_AREA	314
	MAX_PUPIL_RADIUS	400
Ellipse fitting	MIN_NOF_RADIUSES	30
	NOF_RAYS	170

Table A.1: Parameter values used to validate the method. The parameter values are obtained by selecting the values that give the best performance for a single user.