REAL TIME GAZE TRACKING USING A LOW-COST WEB CAMERA

Darragh Cunningham
B.A.(Mod.) Business and Computing
Final Year Project April 2013
Supervisor: Rozenn Dahyot

School of Computer Science and Statistics
O'Reilly Institute, Trinity College, Dublin 2, Ireland
DECLARATION

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university

______________________________________________        ________________________
Name            Date
Abstract

As one of the most salient features of the human face, eyes play an important role in interpreting and understanding a person’s desires, needs, cognitive processes and emotional states. The importance of eye gaze is implicitly acknowledged as it is used for the inspection of our surroundings and is important in social interaction. This paper presents a review of eye gaze tracking technology and focuses on recent techniques which enable gaze tracking systems to be built using low-cost and standard hardware. This project aims to prove that it is possible to create a gaze tracking system using only a standard web camera and the free, open source Computer Vision library OpenCV.
Acknowledgements

I would like to thank my supervisor, Dr. Rozenn Dahyot, for her support, advice and guidance, without which this project would not have been completed.

I would also like to acknowledge and thank my classmates, friends and family for their support during this project.
Contents

1 Introduction
   1.1 Motivation .................................................. 1
   1.2 Aims and Research Goals ................................... 2
   1.3 Structure of Project ......................................... 2

2 Background
   2.1 Introduction .................................................. 4
   2.2 The Human Eye ................................................ 4
   2.3 Visual Selective Attention ................................... 6
   2.4 Eye Movements ................................................ 6
      2.4.1 Convergence ................................................. 7
      2.4.2 Saccades .................................................... 7
      2.4.3 Vergence .................................................... 7
      2.4.4 Smooth purist movement ................................ 7
      2.4.5 Vestibular-ocular movement .............................. 8
      2.4.6 Nystagmus .................................................. 8
         2.4.6.1 Pathological Nystagmus .............................. 8
         2.4.6.2 Physiological Nystagmus ............................. 8
## CONTENTS

2.4.6.3 Optokinetic Nystagmus ........................................ 8

2.5 Gaze Tracking Techniques ........................................... 8
  2.5.1 Infrared Reflection ............................................. 9
  2.5.2 Limbus Tracking ................................................ 9
  2.5.3 Pupil Tracking ................................................ 10
  2.5.4 Corneal and Pupil Reflection Relationship ..................... 11
  2.5.5 Corneal Reflection and Eye Image using an Artificial Neural Network ................................................. 11
  2.5.6 Purkinje Image Tracking ....................................... 12

2.6 Face Detection - Haar Classifiers .................................. 12

2.7 Integral Image .................................................... 13

2.8 Summary .......................................................... 14

3 Design ............................................................... 16
  3.1 System, Languages and Libraries .................................. 16
    3.1.1 Application Software Components ............................. 16
    3.1.2 Application Hardware Components ............................ 17
  3.2 Design Decisions ................................................. 18
    3.2.1 Face Pose .................................................. 18
    3.2.2 Eye Gaze .................................................. 18
    3.2.3 Calibration ................................................ 19
  3.3 Design Process ................................................... 20
  3.4 Summary ........................................................ 22

4 Implementation ....................................................... 23
  4.1 Introduction ..................................................... 23
  4.2 Feature Detection ............................................... 23
  4.3 Facial Detection ................................................ 24
  4.4 Eye Detection ................................................... 25
  4.5 Pupil Detection ................................................ 26
    4.5.1 Eye Centre Detection ....................................... 27
  4.6 Corner Detection ................................................. 28
    4.6.1 Alternative methods of Corner Detection ..................... 30
## List of Figures

2.1 Structure of the human eye [12] ................................. 5
2.2 Corneal reflection and bright pupil as seen in an infrared camera image [6] ................................................. 10
2.3 Corneal reflection and bright pupil technique and implementation [12] .................................................... 11
2.4 Purkinje Image Tracking [12] .......................................... 12
2.5 These Haar-wavelet-like features are computed by adding the light regions and subtracting the dark regions [1]. The image is originally from [29]. ..................................................... 13
2.6 Summed area of rotated integral image summed area of rotated integral image [30] .............................................. 14

3.1 Main Components of the gaze tracking system ............... 20

4.1 Face Detection image ................................................. 24
4.2 Eye Detection Image .................................................. 26
4.3 Pupil Detection Image (with grey scale removed) .......... 27
4.4 Pupil Image taken from an existing system [20] .......... 28
4.5 Corner Detection without greyscale ............................. 29
4.6 Corner Detection using smoothing techniques and greyscale implementation ............................................... 29

5.1 Gaze Detection image with and without greyscale ........ 33
5.2 System test with glasses ............................................. 33
5.3 System test while wearing a hat ................................. 34
5.4 Blink test and system accuracy test using reduced light . . . . . . 35
Chapter 1

Introduction

Eye gaze tracking systems have long been suggested as input devices for computer interfaces. Current systems require invasive, and often expensive, hardware. Contact lenses, electrodes, and head mounted devices are some of the invasive techniques in current eye gaze tracking systems. This project is an attempt to realise the possibility of a non-invasive eye gaze tracking system built using a regular web camera and computer vision software.

1.1 Motivation

The use of eye tracking has the potential to enhance human computer interfaces. Human-computer interfaces utilising eye-movement techniques can be categorised into two types - active and passive interfaces.

Active interfaces allows users to interact and control the interface through the use of eye movements [13]. Eye typing which enables users to type using a virtual on screen keyboard that can be controlled by eye movement is one such example [22].

In a non command-based dialogue, the user does not issue specific command; instead the computer interface observes the user and provide appropriate responses, for example the eye movement found in [15].
While eye tracking systems are prominent across research systems, deployment of consumer based eye tracking systems is relatively low. The primary obstacle preventing deployment of such systems is that current systems and techniques are too expensive and invasive for routine use.

1.2 Aims and Research Goals

The goal of this project is to ascertain whether it is possible to create a gaze tracking system that is low-cost, functional and efficient enough to be useful. The hardware used in this system will be a regular web camera.

Unfortunately, while there have been significant advances in eye-tracking technology, the cost of commercial systems remains prohibitive. Eye-tracking is often not used simply because of the cost factor; making it a technology that is used only for research purposes and in niche applications. The high cost of commercial systems ($1000 - $40,000) has led to numerous efforts to build a more commercially viable gaze tracking system [2; 9; 10; 11]. However, building an eye-tracker and researching application of eye-trackers are two very different tasks.

Coupled with aim of developing a low-cost system is the aim to keep the intrusiveness of the system to a minimum. Many of the current eye gaze tracking systems are quite invasive due to contract with the user or limitations imposed such as keeping head movement static. The aim is to develop a low-cost gaze tracking system which is also non-intrusive to use.

1.3 Structure of Project

The remaining chapters of this project are described below:

Chapter 2 describes the background information required to understand the proposed system as well as exploring existing techniques and systems.
Chapter 3 presents and overview of the system components, design and decision processes made.

Chapter 4 describes the implementation of the system and the steps involved in creating the system structure.

Chapter 5 provides the testing procedures and results obtained and discusses the projects successes, shortcomings and future opportunities.

Chapter 6 outlines the conclusions drawn from this research.
Chapter 2

Background

2.1 Introduction

The promise of non-invasive human-computer interactions using eye gaze tracking systems promises to ease communication difficulties between humans and machines by providing an interface that is both intuitive and easy to operate. Gaze tracking technology provides an alternative method of human-machine interaction which aims to be beneficial in assisting those with and without disabilities. Gaze tracking has a plethora of other uses stemming from biometrics, communication enhancements, information gathering and processing, and training to name but a few. The primary nature of gaze tracking requires a relevant understanding of both the behaviour of human vision as well as the biology and feature of the human eye. In this regard, this chapter presents the relevant information required for the creation of a gaze tracking system as well as exploring the different eye tracking techniques.

2.2 The Human Eye

Literature on the structure of the human eye [3] tells us that the retina consists of approximately 127 million light-sensitive cells. Of these 127 million cells, 120 million or approximately 94% are so called rods which have a high sensitivity, i.e. they can detect relatively small amounts of light. However the detected light is
without colour information. The 7 million cells that are not rods are classified as cones. Cones are less light-sensitive but can capture the colours of the light spectrum. The cones are concentrated in the centre of the retina, in a circular area called macula lutea. Within this area, there is a depression call the fovea which consists almost entirely of cones, and it is this area of high acuity, which extends over a visual angle of approximately $2^\circ$, that humans use to make their detailed observations of the surrounding world \[7\]. The remaining area of the retina provides peripheral vision, which has 15-50% of the acuity of the fovea \[14\]. This is partly due to the allocation of one efferent optic nerve for every 300 rod. However the structure of the peripheral area of the retina is more reactive to flashing objects and sudden movement \[8\].

Figure 2.1: Structure of the human eye \[12\]

The eyeball is controlled by three set of muscles; one set for horizontal eye movement, another for vertical movement, and a third for rotational movement i.e. movement around an axis in the direction of sight.
2.3 Visual Selective Attention

Yarbus [31] showed in his study of eye movements that the perception of a scene involves a complicated pattern of fixations, i.e. where the eye is in a (fairly) still position, and saccades, where moves to examine a new part of the scene. These patterns of fixations are important due to their relationship with the attention selection mechanism. Most research now agrees that the attention selection mechanism consists of two functionally independent, hierarchical stages: An early, pre-attentive stage that operates without capacity limitation and in parallel across the entire visual field, followed by a later, attentive limited-capacity stage that can deal with only one item (or at best a few items) at a time. When items pass from the first to the second stage of processing, these items are considered to be selected [27]. It is important to note that attention is shifted to a new location before the saccade that moves the gaze is initiated, and thus the movements of the eyes should not be considered as the selection process itself, but merely as the outcome of attentional selection processes preceding actual eye-shifts.

2.4 Eye Movements

When the pre-attentive and attentive stages have determined the position of the target, the eye must move in such a way so that the target object can be inspected with more precise precision. The movement of the eyes to a new position is performed by executing a saccade. Jacob [14] and Green [3] state in their research findings that there are seven different eye movements and it is important to consider the different types of movement. These include voluntary conjugate horizontal gaze (looking side-to-side); voluntary conjugate vertical gaze (looking up and down); smoothly tracking objects; convergence; and eye movements resulting from head movements. These latter movements are part of the vestibular reflexes for eye stabilisation. All movements of the eyes are conjugate, i.e. both eyes moving in the same direction in order to keep the eyes focused on a target, with one exception. Convergence movements adducts the eyes to focus on near objects.
2.4.1 Convergence

Convergence is a motion of both eyes relative to each other that ensures that an object is still foveated by both eyes when its distance from the observer is changed; the closer the object is, the more the eyes point towards each other. This movement can be voluntarily controlled, but is normally the result of a moving stimulus [7].

2.4.2 Saccades

Saccades are the principal method for moving eye to a different area of a visual scene, and are described as sudden, rapid movements of the eyes. It takes about 100-300ms to initiate a saccade, i.e. from the time a stimulus is presented till the eye starts moving, and another 30-120ms to complete the saccade, depending on among other things-the visual angle traversed. During saccades the processing of the visual image is suppressed. Thus processing of the retinal image takes place mainly between saccades, during the so-called fixations, which last between 200-600 milliseconds.

2.4.3 Vergence

Vergence movements align the fovea of each eye with targets located at different distances from the observer. Vergence movements are disconjugate; they involve either a convergence or divergence of the lines of sight of each eye in order to determine if an object is near or farther away.

2.4.4 Smooth purist movement

Smooth purist movements are much slower tracking movements of the eyes which are designed to keep a moving stimulus on the fovea. Such movements are under voluntary control in the sense that the observer can chose whether or not to track a moving stimulus. Saccades can also be voluntary but are also made unconsciously.
2.4.5 Vestibular-ocular movement

Vestibular-ocular movement stabilise the eyes relative to the external world by compensating for head movements. The vestibular system detects transient changes in the head position and produces corrective eye movements thus keeping the image of the observed object at more or less the same place on the retina.

2.4.6 Nystagmus

Nystagmus is a pattern of eye movements that occur as a response of head movement. There are two main forms of Nystagmus: pathological and physiological.

2.4.6.1 Pathological Nystagmus

Pathological nystagmus is characterised by a biphasic ocular oscillation alternating a slow eye movement, or smooth pursuit, in one direction and a fast eye movement, or saccadic movement, in the other direction.

2.4.6.2 Physiological Nystagmus

Physiological Nystagmus is a form of involuntary eye movement which is part of the vestibule-ocular reflex. It is characterised by alternating smooth pursuit in one direction and saccadic movement in the other direction.

2.4.6.3 Optokinetic Nystagmus

Optokinetic Nystagmus is a reflexive response of the eyes in response to large-scale movements of the visual scene.

2.5 Gaze Tracking Techniques

The number of ways of tracking the direction of eye-gaze continues to grow. Today, several tracking techniques exist. Arne John Glenstrup and Theo Engell-sen [7] categorise the existing techniques into three main types of tracking:
1. Tracking of the eye by measuring the reflection of light that is shone onto the eye. This is typically carried out using infrared lights in order to minimise user distraction, and to avoid interference from other light sources.

2. Tracking of the eye by measuring the electric potential of the skin around the eye. This is generally carried out using electrodes which are attached to the area around the users eyes.

3. Tracking the eye using a special contract lens which facilitates the tracking of the eyes position.

All techniques require some form of calibration before usage, with some requiring ongoing recalibration during use. The techniques mentioned above are also quite intrusive to use. Tracking using the reflection of light is typically the least intrusive while the third technique, tracking using a special contact lens is regarded as the most intrusive as it is solely contact based.

2.5.1 Infrared Reflection

Of the three techniques discussed in Arne John Glenstrup and Theo Engell-Nielsens paper [7], gaze tracking using infrared reflection offers the least invasive method for tracking a users eye position and gaze. There are five tracking techniques which use light reflected onto the eye. Limbus tracking, pupil tracking, corneal and pupil reflection relationship, corneal reflection and eye image using an artificial neural network and Purkinje image tracking are the five tracking techniques.

2.5.2 Limbus Tracking

The limbus is the boundary between the white sclera and the iris of the eye. Due to contrast between the white of the sclera and the darkness of the iris the boundary can be easily detected and tracked. This technique is based on the position and shape of the limbus relative to the head. This relationship requires the head to be held in a still position. An apparatus which can be fixed to the users head is resolves this issue. Scott and Findlay [25] suggest that "it is
probably fair to regard limbus tracking as suitable for precise horizontal tracking only” due to the fact that much of the limbus is covered by the eyelids.

2.5.3 Pupil Tracking

Tracking the direction of the gaze by the pupil tracking technique shares many similarities to limbus tracking. The main difference relates to the use of the small boundary between the pupil and the iris. The apparatus used to track the pupil must also be held still in relation to the head. Tracking the direction of the gaze using pupil tracking has significant advantages over limbus tracking. The main advantages being that the pupil is far less uninstructed by the eyelids, which enables vertical tracking. The border of the pupil is also sharper than that of the limbus which results in a higher resolution.

![Figure 2.2: Corneal reflection and bright pupil](image)

Figure 2.2: Corneal reflection and bright pupil as seen in an infrared camera image [6]

The only disadvantage is that the difference in contrast is lower between the pupil and the iris compared with the contrast between the iris and the sclera. This makes border detection more difficult.
2.5.4 Corneal and Pupil Reflection Relationship

The first Purkinje image is often called the glint, and combined with the reflection of light off the retina the so-call bright-eye can be recorded using an infrared sensitive camera. The camera detects the bright spot found on the eye and a less bright disc. The positioning of the glint and the centre of the bright-eye change accordingly when the eye is positioned horizontally or vertically. The direction of the gaze can be calculated from these relative positions. The main problems associated with this technique are primarily related to finding a good view of the eye. Lateral head movement can also cause the video image to fall out of focus.

2.5.5 Corneal Reflection and Eye Image using an Artificial Neural Network

An Artificial Neural Network (ANN) is another technique which can detect eye gaze positions by using computations. One of the primary benefits of an artificial neural network based gaze tracker is that it is non-intrusive and allows for free movement of the head. In order to account for the shifts in the position of the eye, the system must locate the eye in each frame. Similar to the other methods mentioned, the system requires a light to be shown on the eye which the system then searches for the reflection of the light glint. It then extracts a small, rectangular part of the image and feeds this to an ANN. The output of the ANN is typically a set of coordinates.

The ANN requires more complex calibration than other techniques. It must
also be trained by gathering images of the user's head and eyes while the user
tracks a calibration program on the display.

2.5.6 Purkinje Image Tracking

Of the four Purkinje images available, only the first and fourth Purkinje images
can be used for tracking the direction of the gaze.

![Purkinje Image Tracking](image)

Figure 2.4: Purkinje Image Tracking [12]

The Dual-Purkinje Image technique [23] uses the relative positions of the
reflections on the eye to calculate the direction. Purkinje Image tracking is gen-
erally more accurate than other techniques and its sampling frequency can be
quite high, up to 4000Hz. The main disadvantage with this technique, however,
is that the fourth Purkinje image is quite weak. This means that the surrounding
lighting condition must be heavily controlled [5].

2.6 Face Detection - Haar Classifiers

Object Detection using Haar feature-based cascade classifiers is an effective object
detection method proposed by Paul Viola and Michael Jones in their paper, Rapid
Object Detection using a Boosted Cascade of Simple Features in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Figure 2.5: These Haar-wavelet-like features are computed by adding the light regions and subtracting the dark regions \([1]\). The image is originally from \([29]\).

The Haar-like features used by the classifier are shown in Figure 2.5. These features use the change in contrast values between adjacent rectangular groups of pixel.

### 2.7 Integral Image

The simple rectangular feature of an image are calculated using an intermediate representation of an image, called the integral image \([28]\). The integral images is an array containing the sums of the pixels intensity values located directly to the left of a pixel and directly above the pixel at the location \((x, y)\). So if \(A[x,y]\) is the original image and \(AI[x, y]\) is the integral image then the integral image is computed as shown in equation 2.1 and illustrated in Figure 2.6.
\[ AI[x, y] = \sum_{x' \leq x, y' \leq y} A(x', y') \]  

(2.1)

The rotated integral image is calculated by finding the sum of the pixels intensity values that are located at a 45 angle to the left and above the x value and below for the y value. The rotated integral image is computed as shown in equation 2.2 and illustrated in Figure 2.6.

\[ AR[x, y] = \sum_{x' \leq x, x' \leq x - |y - y'|} A(x', y') \]  

(2.2)

Figure 2.6: Summed area of rotated integral image summed area of rotated integral image [30]

2.8 Summary

The detection and location of the pupil is paramount in tracking the user’s gaze. By tracking the pupil and other features it is possible to work out where the gaze is. The background research suggest that the fixations of the eye play a key role in finding the gaze.

There are a large number of algorithms available for use, many of which form
the bases of the current eye-gaze tracking technologies such as the Starburst Algorithm [21].

The fundamental requirements appear to be eye detection, pupil detection, detection of an addition feature, and the transformation of all information to focus coordinates which should detect the gaze. These requirements play an important role in deciding how the system will be designed.
Chapter 3
Design

The chapter begins by detailing the design components that are necessary to develop a gaze tracking system. Language and libraries as well as the application components are firstly outlined, before moving onto the specification requirements. Finally the actually design specifics and processes are explored.

3.1 System, Languages and Libraries

The aim of this project is to create a low cost Gaze tracking system using simple techniques, a regular web camera and a computer Vision library. The software and hardware components were chosen based on the aims and criteria as mentioned previously.

3.1.1 Application Software Components

The programming language on which the system should be implemented was determined by several factors. Time, cost and programming experience were all taken into account in this decision process.

Open source language libraries were considered above commercial based libraries in order to keep in line with the expected low-cost nature of the system.
The language and libraries chosen also needed to have strong computer vision characteristics. The choice to develop the Gaze-tracking using OpenCV (Open Source Computer Vision Library) was reached as it addressed all the concerns and criteria that was required.

OpenCV is an open source computer vision and machine learning software library.¹ The library has more than 2500 optimised algorithms, which include a comprehensive set of both classic and state of the art computer vision and machine learning algorithms. These algorithms can be used to detect and recognise faces, identify objects, track camera movements, track moving objects etc.

OpenCV also supports several programming languages including C++, C, Python, Java and MATHLAB and is supported on Windows, Linux, Android and Mac OS. As OpenCV is written natively in C++ it was decided that the programming language to use should also be C++ in order to minimise potential risk factors.

The final software component required was an integrated development environment (IDE). Microsoft’s Visual Studio was chosen due to its support of C++ and OpenCV.

3.1.2 Application Hardware Components

The two main hardware components necessary to develop a gaze tracking system are a machine interface i.e. a laptop or desktop, and a camera to find and track the eye. Keeping with the aim of developing a low-cost system, the web camera chosen for this project is a Microsoft LifeCam HD-3000. The decision to choose this web-camera was due to its relatively cheap cost, the LifeCam HD-3000 currently retails at $39.95², and due to its ability to shoot in 720 pixels, which has a resolution of 1280x720. It also films at 30 frames per second.

¹http://opencv.org
The computer interface and environment used to develop the project on is a Dell Latitude E6500.

3.2 Design Decisions

Eye tracking has traditionally been achieved using several approaches as discussed in Chapter 2. The direction of a person’s gaze is determined by two factors: The orientation of the face (face pose) and the orientation of the eye (eye gaze).

3.2.1 Face Pose

Face pose determines the global direction of the face, while eye gaze determines the local direction of the gaze [18]. These approaches can be categorised into head-based approaches, ocular based approaches, and hybrid approaches using a combination of head and eye approaches. The head based approach determines a user’s gaze based on the head orientation. Rae and Ritler [19] developed a gaze tracking system using a set of Gabor filters which were applied to the image region of the face. This resulted in a feature vector which was used to train a neural network that observed neck angles as well as pan and tilt movement. Head orientation could then be determined from the captured information. Further effort by Ji [16] and Ji and Yang [17] led to the development of several techniques used for face pose estimation. Gaze estimation by head orientation only provides a global gaze as one’s gaze can change considerable given only head orientation. In this regard the accuracy of the user’s gaze is traded for flexibility and ease of use.

3.2.2 Eye Gaze

The ocular based approach estimates gaze by examining the geometric properties of the eye. The special characteristics of the iris structure, namely the transition from white-to-dark and dark-to-white, makes it possible to identify the iris from the eye region. In a conventional ocular based approach [24], the eye is illuminated by infrared light, usually emitted by a light emitting diode. The amount of light reflected back from the eyes is measured using a photodiode and
the information received details the position of the eyes. As the cornea and iris reflect much less light than the sclera, the reflected light allows for ocular-based gaze estimation based on the relative position of the eye and the glint (cornea reflection) on the cornea of the eye.

While non-intrusive these methods of gaze detection only work well when the head is held in a static position. A chin rest is often used in order to maintain a static position of the head, which restricts natural head movements. Such restrictions pose a significant hurdle to natural human-computer interaction. It was decided to forgo the use of a chin rest in the initial design in order to keep with the aim of developing a non-intrusive system.

3.2.3 Calibration

Another issue with existing gaze tracking systems is the need for calibration. Moirmoto et al [4] (2000) used a nine point calibration process based on a 3 x 3 on screen grid. This required the user to fixate their gaze on a certain point or area of the grid when instructed. For each fixation, the pupil glint vector and corresponding screen coordinate were obtained using polynomial transformation to map the relationship between the pupil glint vector and the screen coordinates. Such calibration can be complex and tedious and thus in this regard calibration is seen as a hindrance in human-computer interaction. With the different approaches and their limitations discussed, it was decided that the process of determining gaze estimation should account for computation of the local gaze using a combination of the head-pose approach and the ocular parameter approach. A general approach that combines head pose information with eye gaze information in order to perform eye gaze estimation is proposed. This approach allows for natural head movement while estimating gaze accurately. Another effort is to make the gaze estimation calibration free. This removes the need for calibration or re-calibration when the users moves their head or moves out of frame. By discounting calibration requirements and the need for static head movement, the design proposed needs to be both robust and accurate enough to account for these omissions.
3.3 Design Process

The main program incorporates all steps necessary to create a gaze tracking system. The design in its simplest form can be seen in Figure 3.1.

Figure 3.1: Main Components of the gaze tracking system

The first step in the design process involves the processing of the video image input from the camera. Once the video feed is retrieved the next step requires the program to be able to distinguish the facial features of the user. Face detection
is firstly carried out in order to reduce the search area or region of interest (ROI) of the captured image input. The face detection should be almost instant once the video-input is captured in order to keep with the goal of a fast responsive system. The proposed approach is based on the Viola Jones technique. This face detection method is a variant of the AdaBoost Algorithm [Ada] which achieves rapid and robust facial detection.

Before the gaze can be detected the region of the eye of the user needs to be found. Taking the area of where the face is found, the program then searches for each eye individually using Haar-classifiers and records the position of the eyes if found.

After the face and eye regions are found, the program can now focus on detecting the users gaze. This process requires the pupil and corners of the eye to be detected. There are many different methods for pupil detection, some which are further explained in Chapter 2, which can be roughly categorised into three distinct groups:

1. Feature based methods.
2. Model-based methods.
3. Hybrid methods.

A feature based approach is proposed as it best fits the scenario of gaze tracking in low resolution images and videos. However the feature based approach requires several necessary techniques such as noise reduction and filtering in order to accurately detect the centre of the eye. Finding the eye corners was another technique that was explored. This feature is required in order to track any changes in the eye when it moves within the socket. Once the multi-stage approach is complete the gaze estimation of each eye can be found and displayed.
3.4 Summary

Due to the limitations of both the Face Pose and Calibration processes, the logical decision was to use the ocular based approach as the main process. The main steps in developing the system include face detection, eye detection, pupil detection and finally corner detection. The eye gaze can then be determined once all the features are detected.
Chapter 4

Implementation

4.1 Introduction

This section of the report outlines the steps undertaken to implement the gaze tracking system from the design. The setup of OpenCV in a Windows operating environment and the linking the OpenCV library in visual Studio 2013 are covered in the notes found on the accompanying DVD. The implementation of the system was carried out on a 64 bit Windows 8.1 Operating System using a Dell Latitude E6500 laptop with 4GB RAM and an Intel Core 2 Duo T9600 (2.8 ghz) processor. This chapter will discuss the implementation of the different design stages.

4.2 Feature Detection

The detection of the many features involved in build the gaze tracking system rely heavily upon the classifiers provided in the OpenCV library. Object detection using Haar feature based cascade classifiers is an effective object detection method, which was first proposed by Paul Viola and Michael Jones in their paper [28].

It is a machine learning based approach where a cascade function is trained using a high number of positive and negative images. It is then used to detect objects in other images by comparing what it knows to be true.
Two primary classifiers, supplied from the OpenCV library, were used for the facial and eye detection processes. The first step in the development of the system required the video based capture and display of the user using the web camera.

4.3 Facial Detection

Face detection is the first major step in implementing the system. Using pre-defined Haar classifiers which can be found in the OpenCV library was necessary in determining whether facial features were present in the frame captured by the camera. The exact classifier used for the face detection was the classifier: Haarcascade_frontal_facealt2.xml

![Face Detection image](image)

Figure 4.1: Face Detection image

Once the cascade is loaded the Haar Detection method called CVHAARFIND-BIGGESTOBJECT can be used to find the ROI of the face. Before this was implemented however, the image was subjected to minimal pre-processing tech-
niques. The image is grey scaled which makes the Haar detection more robust. For the sake of testing, a rectangle is drawn around the face. The grey scale function was also temporarily removed in order to show the face detection more clearly.

The Haar detection method first takes in the image. The Haar classifier cascade and the storage space that the answer should be stored in along with several other optional parameters are also included in this process. The full use of the function can be seen in appendix C.1. The function then returns a sequence of the detected face, from which the x and y coordinates can be extracted. These coordinates are needed to determine the region of interest (ROI) to search when using the Eye Detection method.

In order to keep the program efficient and fast it is important to define specific regions of interest to search for the eyes.

### 4.4 Eye Detection

The method used for the eye detection is similar to the method used for the face detection in that it uses the same processes but with a different classifier. The classifier used for the detection of both eyes is the Haar cascade eyetreeeyeglasses.xml. The steps carried out for the eye detection follow the same steps as used in the facial detection: Use the Haar detection method, set the ROI, copy the ROI to a new image and return this image. Because there are two eyes presented, the eye detection is done separately for each eye, though the function is still carried out in one iteration of the run process. Testing of the program found that the Haar detection for the eyes was not as accurate as the face. The addition of filtering to smooth the face image was applied along with other noise reducing techniques in order to increase the accuracy of the program.
The pupil detection is by far the most important and difficult part of the process in determining the eyes gaze. The process of detecting the pupil was carried out in a loop step process.

Pupil Detection Process

1. Load the image source
2. Invert the image
3. Convert to binary image by thresholding it
4. Find all the blobs
5. Remove noise by filling holes in each blob
6. Get blob which is big enough and has a round shape

After implementing the first two steps, it is necessary to convert the image by thresholding it.
After thresholding the image the process then looks for round shape objects. The round object may contain holes due to light reflections so it is necessary to remove these holes in order to find the correct circle area. After the blemishes are removed the program runs a loop of each blob in the image until it finds the blob with a round shape. The resulting blob is the pupil of the eye.

One of the difficulties in detecting the pupil stems from the clarity of the captured eye image seen in Fig 4.3.

![Figure 4.3: Pupil Detection Image (with grey scale removed)](image)

By comparison to an existing system it is clear that the captured image of the pupil is likely to fail in detecting a user's pupil and thus gaze.

Due to the poor clarity of the capture eye region, the techniques implemented to capture the pupil presented a high failure rate during testing. To fix this issue it was decided that the centre of the eye should be used in lieu of the problems of detecting the pupil itself.

### 4.5.1 Eye Centre Detection

Finding the centre of the eye also makes use of previous techniques, that were proven to work, from the face and eye detection methods. The region of interest
was determined using the eye ROI, and the main algorithm to test for all possible centres was used. The process of detecting the centre of the eye starts with determining the region of the eye. The gradient then needs to be found and normalised. The main algorithm is then run to calculate returning the $x$ and $y$ coordinates of the centre of the eye.

### 4.6 Corner Detection

The Corner detection is the only step not to work directly off the prior step. The corner detection uses the same ROI used for the eye detection. Corner detection like the Pupil Detection has a number steps it must implement before the corner is found.

**Corner Detection Process**

1. Create an eye corner Kernel
2. Create corner Kernels for left and right eye and flip horizontally
3. Create an eye corner map
4. Find the eye corner
5. Find sub-pixel eye corner

A conventional kernel is a single-channel floating point matrix. The anchor of the eye corner kernel is relative to the ROI of the eye. After the kernel is applied to the ROI it is necessary to create a map of the eye corner region. OpenCV provides a function called filter2d which allows for the specification of the input and output image along with the kernel to use.

![Figure 4.5: Corner Detection without greyscale](image1.png)

**Figure 4.5: Corner Detection without greyscale**

![Figure 4.6: Corner Detection using smoothing techniques and greyscale implementation](image2.png)

**Figure 4.6: Corner Detection using smoothing techniques and greyscale implementation**

After creating the eye corner map the eye corner can be found by adjusting the sub-matrix size and its position within the parent matrix.
4.6.1 Alternative methods of Corner Detection

The function cvGoodFeaturesToTrack finds corners with big eigenvalues in the image. The function first calculates the minimal eigenvalue for every source image pixel using cvCornerMinEigenVal function and stores them in eig-image. Then it performs non-maxima suppression (only local maxima in 3x3 neighbourhood remain). The next step is rejecting the corners with the minimal eigenvalue less than quality-level*max (eig-image(x,y)). Finally, the function ensures that all the corners found are distanced enough one from another by considering the corners (the strongest corners are considered first) and checking that the distance between the newly considered feature and the features considered earlier is larger than min-distance. As the image region is quite dark and texture less it was decided against using cvGoodFeaturesToTrack as a method for finding Corner Detection [26].

4.7 Conclusion

The Gaze detection is determined by combining the Pupil and Eye Corner Detection processes with the captured video stream. The gaze is then detected and displayed in a new output window. Many problems were found in the implementation process but were largely resolved through error reduction using redundancy and through further research of alternative or complementary techniques.
Chapter 5

Evaluation

The purpose of this chapter is to review the project as a whole. First, the results that were achieved are presented and discussed, followed by a review of the projects success and the difficulties encountered. Finally, the chapter will conclude with a discussion relating to future work opportunities.

5.1 Results

The implementation of the system extended methods to infer facial detection, pupil detection and eye corner detection.

5.1.1 Facial Detection

The system was able to successfully detect the users face from the input video stream. However since it is colour-based, a skin-colour tone similar to the background of the captured input will impair the face finder slightly. The facial detection method has only been tested on Caucasian skin and it may not work reliably with other skin types.

5.1.2 Eye Detection

The detection of the eye was also successfully implemented and worked surprisingly well given the technological limitations of the hardware compared to more
advanced eye tracking systems.

5.1.3 Pupil and Corner Detection

The system failed to present an accurate representation of the pupil, as mentioned in the Chapter 4, due to limitations posed by the hardware used. Although region of interest of the pupil could be determined, due to the poor clarity of the capture eye region, the techniques implemented to capture the pupil presented a high failure rate during testing.

As mentioned in Chapter 4, the failure of detecting the pupil meant that the gaze detection could not be detected initially. To fix this issue it was decided that detecting the centre of the eye could be used in lieu of detecting the pupil.

The exactness of the centre of the eye was difficult to establish. Due to errors relating to finding the eye centre, detecting the corner of the eye was needed. The corner of the eye was determined with little difficulty and improved the accuracy of determining the users gaze.

5.1.4 Images Results

Figure 5.1 shows the successful implementation of the eye gaze system. The eye corner and the centre of the eye was successfully detected when run under normal conditions.

5.2 Testing

Although the accuracy of the system is bounded by the quality of the webcam used, the results indicate that it is possible to approximate gaze detection. It was found that the web camera can film up to 30 frames per second, but only in a resolution of 640 x 360 pixels. In 720p resolution the camera resorted to a frame rate of 10 fps, which isn’t really enough. Lighting conditions and image quality are the main limitations were the main conditions considered during testing.
Lighting conditions in the testing room can lead to poor recognition of the eye, a problem that could be solved if an infrared camera was used. The system was tested in both natural (daylight) and artificial lighting conditions.

The Facial detection was only tested on Caucasian skin and research shows that other skin tones may cause the system to fail.

It was found during testing that the optimum head position for both Face and Eye detection stages is approximately 30-40cm from position of the camera.
When the user is placed at a distance greater than 40cm from the camera the accuracy of the system was greatly reduced and prone to failure. Although the system can support facial detection up to 2metres from the camera the system was unable to detect the eye features.

It was found during testing that glasses and head wear did not hinder facial detection and only slightly increased detection failure rates. However, for detecting eye regions, glasses can introduce reflections and distortions which lead to increased difficulty when detecting specific eye features (see Fig 5.2). It is possible that the trained classifier used did not have enough glasses wearing examples but the reflections on the lens of the glasses is a more likely reason for the failure to detect the eye corner and eye centre regions.

Figure 5.3 highlights shows the successful detection of the face, eyes and eye centre. In the optimum position of 30cm from the camera, the Face and Eye detection stages have a failure rate of approximately 1%.

One hundred iterations of the program were run using a range of eye movements, with a recorded failure of 1 out of 100 which was caused by poor lighting conditions. The resulting failure was captured Figure 5.4.
The system ability to handle blinking is also shown in Figure 5.4. When the eye is closed the system records the last known position of the gaze and will re-run the detection methods if the reappears.

5.3 Project Successes

The aim of this project from the outset was to create a non-invasive eye gaze tracking system using a low cost web camera and computer vision software. The eye gaze tracking system required the detection of both facial and eye detection features in order to approximate the gaze of the eye. To this extent, the project goals and aims were met and can be deemed a success.

The project was implemented was developed using computer vision techniques provided by the computer vision library OpenCV. The design components were developed in stages starting with facial detection and finishing with the gaze detection of the user. In this regard the components can be changed or removed depending on requirements.

The development of this project was embarked upon without any previous
vision based software development experience. Since vision based software is primarily developed using C++, it was necessary to develop the ability to code using this language having never used this language before in a development sense.

5.4 Project Difficulties

As is the case with any software development, many difficulties were encountered.

5.4.1 Approach

As a novice to the vision side of computer science, I had no experience with computer vision programming having not studied the computer science module. From the outset the lack of computer based vision experience meant that there were minor difficulties in implementing OpenCV in Visual studio. These difficulties were overcome with the help on online support and further research.

5.4.2 Design Choices

Relatively few difficulties arose from the design choices chosen. Research papers and literature outlined the numerous design choices used in developing similar gaze tracking systems. The design choices were limited due to the hardware that was available.

5.4.3 Implementation of Choices

The system is implemented in Windows using Microsoft Visual Studio. The system itself however can work outside of Windows, though the code listings found in appendix B will only work in a C++ environment with OpenCV installed.

5.4.4 Application Difficulties

The major application difficulty that arose related to the pupil detection. Although the pupil was detected, due to the poor clarity of the captured eye region
the technique implemented had to be changed. Techniques used to find the centre of the eye were instead implemented and further techniques to improve the image clarity were required and successfully implemented.

5.5 Future Work

As with any software application, there is always room for future work. The following are some of such areas:

- Improvement on existing techniques through technique comparisons.
- The addition of complementary techniques such as calibration.
- Incorporating controls for left and right movements.
- Implementing a gaze tracker using a mobile phone or tablet.
- Enhancement of the image quality using pre-processing techniques.
Chapter 6

Conclusion

This report has outlined the successful implementation of an eye tracking system capable of autonomously detecting, locating and displaying facial and eye region features. Gaze tracking systems are evolving at a rapid pace and have the potential to become an important tool in human-computer interaction. An eye tracker as an input device is far from perfect, in the sense that a keyboard or mouse is. This is due to the nature of human eye movements and the limitations of current equipment. Gaze trackers, however, provide an important insight into a user’s interaction and intentions.

In undertaking this project, a more rounded knowledge of computer vision and tracking technologies was attained. While difficult and challenging, this project has been engaging, interesting and ultimately a rewarding experience. Due to the design and implementation of this system, the system is open for change and improvement be it by my hand or other. Therefore this system can potentially act as a good starting point for any future work in a similar area.

The finished system and code can be found on an accompanying DVD. Instructions on the installation and use of OpenCV are also provided on the disc.
References


[16] Q. Ji. 3d face pose estimation and tracking from a monocular camera l1. *Image and Vision Computing*, 20. 18


REFERENCES


