DEVELOPMENT OF EYE GAZE ESTIMATION SYSTEM USING TWO CAMERAS

NEOH YU ZUN

UNIVERSITI SAINS MALAYSIA

2017

DEVELOPMENT OF EYE GAZE ESTIMATION SYSTEM USING TWO CAMERAS

By

NEOH YU ZUN

A Dissertation submitted for partial fulfilment of the requirement for the degree of Master of Science (Electronic Systems Design Engineering)

August 2017

DECLARATION

I, Neoh Yu Zun declare that all the concepts, designs and experiments in this research are not from others but is entirely from my own job except for the references used in this research.

ACKNOWLEDGEMENT

First of all, I would like to take this opportunity to thank my supervisor Associate Professor Dr. Haidi Bin Ibrahim for the guidance and helping me with this research project. Dr. Haidi has given me a lot of guidance and useful ideas to make this research project a success. The enthusiasm and patience presented by him while guiding me on this research project is very appreciated.

I would also like to thank all my friends for helping me and gives me ideas when I face problems while doing this project. Finally, I would like to thank my parents and other family members for their support and encouragement.

The guidance, help, support and encouragement from all the above-mentioned people have provided me the strength to embark on this thesis. Thank you once again.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

3D	3 dimensional
cm	Centimetre
CCD	Charge Coupled Device
fps	frames per second
GAET	Genetic Algorithm based Eye Tracking
GB	Gigabyte
GUI	General User Interface
HCI	Human Computer Interface
HDR	High Dynamical Range
IR	Infrared
LED	Light Emitting Diode
NTSC	National Television System Committee
PSOET	Particle Swarm Optimization based Eye Tracking
RAM	Random Access Memory
USD	United States Dollar

PEMBANGUNAN SISTEM PENGANGGARAN PANDANGAN MATA MENGGUNAKAN DUA KAMERA

ABSTRAK

Pandangan mata adalah arah mana seseorang itu melihat. Ia adalah sesuai untuk digunakan sebagai sejenis antara muka manusia dengan computer semulajadi. Penyelidikan semasa menggunakan inframerah atau LED untuk mencari iris pengguna untuk mendapatkan ketepatan penganggaran pandangan yang lebih baik berbanding dengan penyelidikan yang tidak melakukannya. Inframerah dan LED adalah mengganggu mata manusia dan mungkin menyebabkan kerosakan kepada kornea dan retina mata. Kajian ini mencadangkan pendekatan tidak menggangu untuk mencari iris pengguna. Dengan menggunakan dua kamera jauh untuk menangkap imej -imej pengguna, ketepatan penganggaran pandangan yang lebih baik dapat dicapai. Sistem ini menggunakan algoritma-algorithma lata Haar bagi mengesan kawasan - kawasan muka dan mata. Pengesanan iris menggunakan algoritma Jelmaan Hough Bulatan untuk menentukan kedudukan iris yang kritikal bagi pengiraan anggaran pandangan. Untuk membolehkan sistem untuk mengesan lokasi mata dan iris pengguna dalam masa sebenar, sistem ini menggunakan CAMshift (anjakan-min suai berterusan) untuk menjejak mata dan iris pengguna. Parameter-parameter daripada mata dan iris kemudian dikumpulkan dan digunakan untuk mengira arah pandangan pengguna. Kamera kiri dan kanan mencapai ketepatan 70.00% dan 74.67%. Apabila dua kamera digunakan untuk menganggar arah pandangan, ketepatan 88.67% dicapai. Ini menunjukkan bahawa dengan menggunakan dua kamera, ketepatan penganggaran pandangan bertambah baik.

DEVELOPMENT OF EYE GAZE ESTIMATION SYSTEM USING TWO CAMERAS

ABSTRACT

Eye Gaze is the direction where a person is looking at. It is suitable to be used as a type of natural Human Computer Interface (HCI). Current researches uses infrared or LED to locate the iris of the user to have better gaze estimation accuracy compared to researches that does not. Infrared and LED are intrusive to human eyes and might cause damage to the cornea and the retina of the eye. This research suggests a non-intrusive approach to locate the iris of the user. By using two remote cameras to capture the images of the user, a better accuracy gaze estimation system can be achieved. The system uses Haar cascade algorithms to detect the face and eye regions. The iris detection uses Hough Circle Transform algorithm to locate the position of the iris, which is critical for the gaze estimation calculation. To enable the system to track the eye and the iris location of the user in real time, the system uses CAMshift (Continuously Adaptive Meanshift) to track the eye and iris of the user. The parameters of the eye and iris are then collected and are used to calculate the gaze direction of the user. The left and right camera achieves 70.00% and 74.67% accuracy respectively. When two cameras are used to estimate the gaze direction, 88.67% accuracy is achieved. This shows that by using two cameras, the accuracy of gaze estimation is improved.

CHAPTER 1

INTRODUCTION

1.1 Background

Human beings use their eyes to visualise things around them and to communicate with surrounding peoples. More than half of the human sensory information comes from the eye, so the eye gives many information to the user. By estimating the eye gaze, a person's presence, attention, focus, drowsiness, consciousness or other mental states can be identified (Panev & Manolova, 2015). Besides, eye gaze can also be used as a type of Human Computer Interaction (HCI) that helps human to communicate with machine or devices. HCI that is based on eye gaze enables user to interact with devices in a more natural way as compared to interactions through a mouse or a keyboard (Liu *et al*., 2006).

Figure 1.1 shows the anatomy of the eye. The sclera and the cornea are the outside layer of the eye. The pupil is the opening that allows light enters the eye, and size of the pupil is controlled by the iris. To obtain the eye gaze of the user, most researchers take gaze information from the iris. This is because the image property of the iris will change accordingly, depending on the focus point of the user. When the person changes the viewing direction, the iris will also change in the direction where the person is looking at.

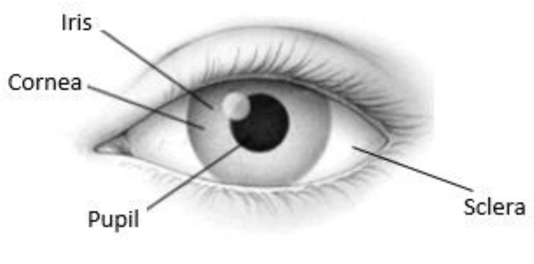

Figure 1.1 Anatomy of eye (Allen, 2015)

Recently, ergonomic HCIs have been developed. Some of the examples are body gestures, touch screens, facial and eye gaze and speech recognition (Kim *et al.*, 2016). All these HCIs share the same purpose, which is to let the user to have a natural way to interact with the computer. In this research, eye gaze interface is chosen because out of all these interfaces, eye gaze interface is the most natural way for a user to interact with a computer. This is because to use a computer, the user will first have to be able to see what is displayed on the computer.

Two areas which are the industrial field and the medical field will benefit by using eye gaze as an interface for HCI. For example, in a process plant, accidents often occur. This leads to large losses which are worth up to millions in United States Dollar (USD). Analysis shows that human errors are the primary cause for more than 70.00% of accident from the past (Kodappully *et al.*, 2016). By using eye gaze interface, this will allow the operator to interact with the computer without even touching the computer. This will help in cases where operators' hands are full with other stuffs but at the same time they still need to interact with the computer. Eye gaze can also serve as a preventive measure for the operators who are controlling the machine. This is because, by monitoring the eye gaze of the operators, the performance of operators is also monitored and this can reduce the errors caused by operators (Kodappully *et al.*, 2016).

Besides, the industry is now moving into Industry 4.0 era. Industry 4.0 paradigm states that all objects of the factory world will be equipped with integrated processing and communication capabilities (Gorecky *et al*., 2014). With this, it means that the machines will become more automated and self-organizing. Here the eye gaze interface can be applied on the machine as HCI. This HCI will definitely serve many benefits in the industry field.

The second area is in the medical field. Eye gaze based HCI will bring huge benefit to people who had been injured or handicapped, and had lost control in their hand. Using eye gaze interface, they can control the computers or machine without the need of using hands (Hiley *et al*., 2006). Besides, eye gaze is also able to revolutionize the mental health screening. This is because the information that are related to the symptoms of some mental health issues can be observed from the eye gaze (Itti, 2015).

1.2 Problem Statements

Current eye gaze researches have several weaknesses. The first weakness is that, to get an accurate location of the iris, intrusive methods such as locating infrared or LED lights near to the iris are normally being employed. Infrared and LED are considered intrusive to the human eye because when the eye is exposed to these light sources for a long period, they might cause damage to the cornea and the retina of the eye (Earman, 2016).

Intrusive methods normally have a better performance compared to non-intrusive methods (Santos *et al*., 2014). This is because by using intrusive methods, infrared is exposed to the eye and the system will then track where the iris is looking at (Hiley *et al.*, 2006). Non-intrusive methods used to obtain the iris location do not rely on infrared or LED, so the quality of image captured is lower as compared with by using intrusive methods. Lower quality of images captured leads to lower accuracy in gaze estimation.

Remote gaze estimation system usually has the camera attached to the screen of the system and is tracking the user's gaze remotely. While for wearable gaze estimation system, the system usually has the camera attached to the user and the camera is placed very near to the user's eye. The second weakness is that the performance of the remote gaze estimation system might not be as good as the wearable gaze estimation system. This is because for wearable gaze estimation system, with the camera put so closely to the eye, it is able to capture less noisy images (Lee *et al*., 2013). Besides, head mounted camera is able to capture images that contains much information about the eye (Kocejko *et al.*, 2009). On the other hand, by using a remote camera, the eye region on the image is relatively smaller, which means there is less information being captured as compared to the head mounted camera. The term remote camera refers to camera which are not attached to the user.

This research aims to have a non-intrusive method to locate the iris location using remote camera. This will let the user to have a comfortable experience in having their iris located. To overcome the weakness that existing researches are facing, this research proposed to use two cameras to overcome the weaknesses discussed. By using two cameras, more data of the eye can be collected. With more data for the system to analyse, the accuracy of the system can be improved. Besides, the secondary camera will also help to act as a verification for the first camera.

1.3 Research Objectives

The objectives of this research are:

i) To develop an algorithm that can detect the iris correctly for non-intrusive, remote camera based eye gaze system

ii) To formulate a method that can accurately estimate eye gaze direction by using two cameras.

1.4 Project Scopes

For this research project, an eye gaze estimation system using two cameras will be developed. The system is designed by using two remote cameras to capture the image of the user. The system will first focus on how to control two cameras to capture the image of the user and how to process the data received from two cameras. Two identical cameras will be considered as the first option for this project because it will be easier to process data with the same format. The system designed is targeted to be used as HCI.

The user of the system designed is assumed to face the camera directly at a distance of around 40cm to 50cm. The system is designed for one user only. After the images are captured, the system should be able to detect the user's face region from the images. Next, the system will then detect the eye region of the user from the facial region. After the eye region is detected, the system will then track the eye of the user.

A formula is needed to estimate the eye gaze of the user by calculating the parameter obtained from the tracked eye region. Next the system will need to compare the estimation collected from two different cameras, processed it and come out with a better accuracy gaze estimation.

There are several limitations in the algorithm developed. First, the algorithm is not able to track the user's eye gaze when there is no light or in very low light environment. This is because the algorithm is developed to process the image captured from the camera. When in a dark room or a low light environments, the camera is not able to capture a usable image for this algorithm to extract the image needed to track the eye gaze. It is recommended to use this algorithm where there is sufficient light and the features of the user's face are captured clearly by the camera. The quality of image captured will affects the accuracy of this algorithm.

Second, this algorithm does not support user with a cross eye problem. This is because to support cross eye users, it requires this algorithm to add in pre-training steps before the user is able to use the algorithm. This project aims to let users have a user-friendly experience, by adding pre-training steps it will complicate this system and make it not user friendly to the user.

1.5 Thesis Outline

This report consists of five chapters. The first chapter is introduction. In the introduction chapter the background of eye gaze interface is discussed. The problem statements in the current approach to locate the iris of the user is also discussed. Next the objectives and project scope of this research is defined.

The second chapter is the literature review chapter. In this chapter, the strength and weakness of each algorithm used by past researchers are discussed. This is important to have knowledge on how past researchers design their system to use eye gaze interface as HCI.

The third chapter is the methodology chapter. In this chapter, the overall process flow of the system is designed. This chapter explains on how the system works to locate the iris of the user by separating into few stages such as the face detection stage, eye detection, iris detection and eye and iris tracking. Lastly, the gaze estimation of the user is then calculated.

The fourth chapter is the results and discussions chapter. Experiments are designed and carried out to evaluate the method designed in chapter 3. The results obtained from the experiment are then discussed.

The last chapter is the conclusion chapter. This chapter will give a conclusion of the whole research. Next the limitation of this research is discussed here. Finally, recommendations for future improvements on this project is discussed.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview on Related Researches

Eye gaze estimation aims to estimate the direction where the user is focusing on. By locating the iris position, the region of interest around the focusing point can be identified and this can be used as an important information for HCI (Kim *et al*., 2016). For example, the computer can interact with the user by just knowing where the user is looking at, and this will make the interaction between the user and the computer become more natural, without the need of a contact based input devices such as a mouse or a keyboard. With the advancement of technologies, currently there are many different approaches used to detect eye gaze. Therefore, in this section these approaches will be discussed.

Generally, methods used to locate the iris can be categorized into two categories, which are the model based methods and the appearance based methods (Lu & Chen, 2016). Model based methods work by modelling the eyeball and then eye gaze can be computed geometrically. The small features on the eyeball images are extracted to obtain key parameters of the eye model (Lu & Chen, 2016). On the other hand, the appearance based methods use a different type of approach where they represent all pixel values from the entire eye image as a vector and a training is required to learn the mapping between eye features and real gaze positions (Lu & Chen, 2016). However, both methods have their own advantages and disadvantages in tracking the eye gaze.

In order to locate the iris, first, the image of the eye need to be captured. Then, the image is processed with suitable algorithms to determine where the eye is looking at. There are two different approaches when it comes to the image acquiring process. Some of the researchers use head mounted camera while other researchers use remote camera to track the eye gaze (Kim *et al*., 2016). The image captured from a head mounted camera is easier to process because the camera is placed at a fixed distance between the eye and camera, so that the eye images are easier to be detected. Furthermore, the image captured is very clear and contains more information as compare to the image from a remote camera because the camera is placed very close to the eye. Yet, to place the camera very near to the eye, this requires the user to wear some equipment to hold the camera. As a consequence, this will cause the user to feel uncomfortable (Amudha *et al*., 2016). On the other hand, the camera in remote camera approach is placed further from the user's eye. Although the image captured by using a remote camera does not contain as much information as compared to the image from a head mounted camera, the user feels more comfortable during the iris location process. The image captured by a remote camera contains sufficient information for eye gaze estimation.

Aside from the difference in the methods used and how the camera is positioned to acquire images, the other variation is on the use of intrusive and non-intrusive methods to capture the image to track the eye gaze. Intrusive methods normally involve the usage of infrared light or LED light that are directly exposed to the eyes. Examples of some researchers that used intrusive methods are Park (2004), Liu *et al*. (2006) and Ebisawa and Fukumoto (2013). With distrust on the approach of using intrusive ways to track the eye, this motivates researchers to use another approach, which is non-intrusive methods (Santos *et al*., 2014). Non-intrusive methods do not expose the eye under infrared, and they deem to be a safer and more comfortable way to track the eye gaze. Intrusive methods are generally more accurate when comparing to non-intrusive method, but the accuracy of non-intrusive methods are also sufficiently accurate to be applied on practical uses (Santos *et al*., 2014).

Eye gaze estimation using remote camera for both intrusive and non-intrusive methods to locate the iris normally follows a general image processing sequence. This is shown in Figure 2.1. As shown in this figure, the process can be broken down into three main stages.

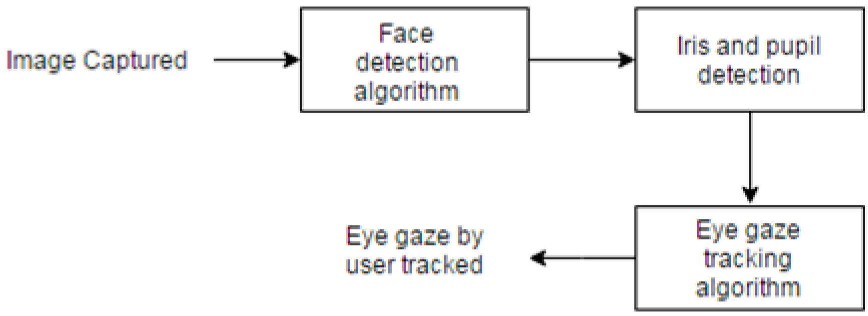

Figure 2.1 General block diagram in image processing to estimate the eye gaze

To be able to estimate the eye gaze of the user, first, an image of the user is captured. Then face detection algorithm is applied to detect the face. In this phase, the algorithm applied will find out the location of the face and it will then make suitable changes to accommodate user's head movement. The face detection algorithm will be discussed in detail in Section 2.1.1. Next is the iris and pupil detection phase. Some researchers such as Florea *et al*. (2016) uses the centre of the iris to determine eye gaze while some other research such as Santos *et al.* (2014) detect the pupil and used it to estimate the eye gaze. There are also researchers that use both pupil and iris to estimate the eye gaze. Here, the algorithms to detect the iris and pupil are discussed in Section 2.1.2.

The last stage is the stage where many different algorithms are used to track the eye and iris. After the eye and iris are tracked, the gaze of the user is estimated.

Different types of eye and iris tracking algorithms and gaze estimation are discussed in Section 2.1.3. After going through all three processes, the eye gaze of the user is estimated.

2.1.1 Face Detection Algorithms Use to Determine the Head Position

Face detection algorithms are used to extract the face out from the image. This is mainly used to minimise the area to be processed because smaller images use shorter time to process (Corcoran *et al*., 2012). Besides, the face detection algorithm is also used to determine the head position to locate the iris of the user. Remote camera approach to acquire an image normally needs a face detection algorithm before eye and iris detection. This is because the image captured normally includes the user's face and this can be used to detect the eye of the user easily. Head mounted camera approach does not require face detection algorithm because the image captured only contains images of the eye.

Table 2.1 shows examples of face detection algorithms used on some past researchers, from year 2009 to 2016. From this table, it is shown that the most used algorithms by past researchers to detect the face is Viola-Jones algorithm. It is worth noting that Haar Cascade and Haar-like features are actually quite similar to the Viola-Jones algorithm because Viola Jones algorithm is a reminiscent of Haar Basis functions (Viola & Jones, 2001). Haar classifiers works by features extraction and it will find variation in a group of pixels and distinguish into darker and lighter shades (Santos *et al.*, 2014). Haar classifiers are trained by using two groups of images with one group that contains a good image while the other group contains bad images.

Table 2.1 List of face detection algorithms used by past researches

Researchers	Face detection algorithm used
Orozco *et al*. (2009)	Appearance based trackers
Wang *et al.* (2010)	Viola-Jones algorithm
Corcoran *et al.* (2012)	Modified algorithm based on Viola-Jones
Elahi *et al.* (2013)	Viola-Jones algorithm
Santos *et al.* (2014)	Haar Cascade classifier
Kraichan and Pumrin (2014)	Viola-Jones algorithm
Florea *et al*. (2016)	Viola-Jones algorithm
Amudha *et al*. (2016)	Haar-like features
H.-I. Kim *et al.* (2016)	Constrained local model
B. C. Kim *et al.* (2016)	HD face model by Kinect

2.1.1 (a) Viola-Jones Algorithm

Viola Jones algorithm uses three kinds of features for object detection. The three features are:

i) The value of a ***two-rectangle feature*** is the difference between the sum of the pixels within two rectangular regions (shown in Figure 2.2 (a) and (b)).

ii) A ***three-rectangle feature*** computes the sum within two outside rectangles subtracted from the sum in a center rectangle (shown in Figure 2.2 (c)).

iii) A ***four-rectangle feature*** computes the difference between diagonal pairs of rectangles (shown in Figure 2.2 (d)).

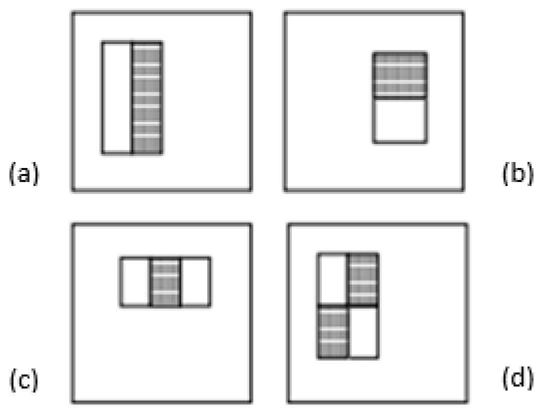

Figure 2.2 Rectangle features used in Viola Jones algorithm (Viola & Jones, 2001)

The rectangle features are then computed using an intermediate representation to obtain the image which is known as integral image. After that, the Viola Jones algorithm requires some learning to be able to detect the face. In this case the AdaBoost learning algorithm is used to boost the classification performance. AdaBoost works by combining a collection of weak classification to form a stronger classifier. Each weak learner classifier is designed to select the single rectangle feature which can best separate the positive and negative examples. The equation for a weak classifier is shown in Equation (2.1).

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

Each weak classifier ($h_j(x)$) consists of a feature (f_j), a threshold (θ_j) and a parity (p_j). In this formula, the weak learner determines the optimal threshold classification such that the minimum number of examples is misclassified. After the learning section, a cascade of classifiers is constructed to achieve higher detection performance while reducing the computation time. This cascade classifier is used to reject negative sub windows while detecting almost all positive instances. It functions like a decision tree where subsequent classifiers are trained using examples which pass through all previous stages as shown in Figure 2.3 (Viola and Jones, 2001).

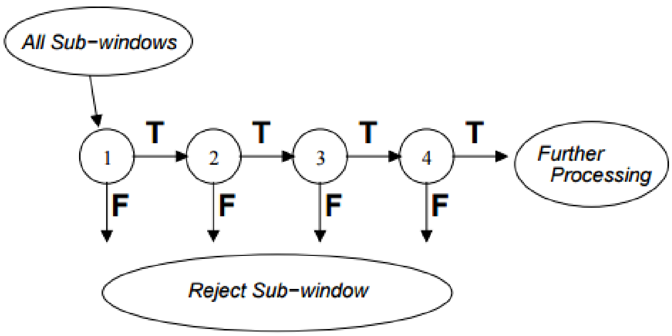

Figure 2.3 Schematic depiction of a detection cascade (Viola and Jones, 2001)

2.1.1 (b) Other Face Detection Algorithms

Aside from Viola Jones algorithm the other algorithms use to detect the face are constrained local model, appearance base trackers and HD face model by Kinect. The constrained local model consists of two steps which is a face tracking and facial feature analysis step. It also uses some of the Haar basis functions (Kim *et al.*, 2016). Appearance base methods identify the face by marker-less approaches. However, these techniques are very hard to be implemented (Orozco *et al*., 2009). On the other hand, the HD face model is an algorithm specifically designed for the Microsoft Kinect computer which involves the use of cameras and depth sensor.

2.1.2 Iris and Pupil Detection Algorithms

Some researchers such as Santos *et al.* (2014) uses Haar cascade to detect the eye after face detection stage while other researches such as Kim *et al.* (2016) skip the eye detection stage and directly proceed to detect the centre of the iris or pupil. Most of the head mounted camera approach, researchers also directly detect the iris or the pupil. Table 2.2 shows the list of algorithms used by selected past researchers from year 2006 until the year 2016.

Table 2.2 List of iris and pupil detection algorithms used by past researches

Researchers	**Eye and Iris Detection Algorithm Used**
Hiley *et al*. (2006)	Bright-eye effect
Zhang *et al.* (2009)	Subtraction between two consecutive frame images which are the “bright pupil” and the “dark pupil
Wang *et al.* (2010)	Bright pupil and dark pupil difference scheme, Hough Transform
Corcoran *et al.* (2012)	Haar Cascade Classifier
Ebisawa and Fukumoto (2013)	Pupil-corneal reflection
Kraichan and Pumrin (2014)	Curvature
Santos *et al.* (2014)	Hough Circle Transform
Amudha *et al.* (2016)	Haar Cascade Classifier
Kim *et al.* (2016)	Iterative iris circle detection method
Florea *et al.* (2016)	Template matching

Table 2.2 shows that there are many different algorithms used by past researches to detect the iris and pupil. From the algorithms used, they can be categorized into two categories. The first category, which is by using intrusive methods, normally have infra-red (IR) or Light Emitting Diode (LED) which is exposed directly to the eye. The first category uses pupil reflection or bright pupil and dark pupil difference to detect the iris or pupil (Wang *et al*., 2010). That is why it can only be achieved by using intrusive methods to capture the eye image. The second category, which is by using non-intrusive methods, have to rely on template matching algorithm to detect the iris or pupil (Florea *et al*., 2016).

2.1.2 (a) Iris and Pupil Detection for Intrusive Methods

The works by both Zhang *et al.* (2009) and Wang *et al.* (2010) detect the pupil by highlighting the pupil feature using the difference between a bright pupil and a dark pupil. The bright pupil is captured when the LED is turned on, whereas the dark pupil is captured when the LED is turned off. To get the location of the pupil, subtraction is made between the bright picture and the dark picture. The facial contour is then removed using connectivity analysis. The pupil is then located by searching the whole image for two dark regions (Zhang *et al*., 2009). This type of approach is not sensitive to external interference as it is based on comparison of the dark and bright pupil. The pupil corneal reflection algorithm proposed by Ebisawa and Fukumoto (2013) also uses a similar concept to the bright pupil and dark pupil way of detecting the pupil.

2.1.2 (b) Iris and Pupil Detection for Non-Intrusive Methods

The non-intrusive approach has a different way of detecting the eye and iris when compare to the intrusive approach. Non-intrusive approach uses Haar Cascade classifier or Hough circle transform algorithms. Haar Cascade classifier is also used to detect the face and has been discussed in Section 2.1.1. One research that uses Haar Cascade classifier is by Corcoran *et al.* (2012). In this research Haar Cascade classifier is used to detect the centre of the pupil from the convolution of the input image with Haar-classifiers. Hough circle transform algorithm is used in Santos *et al.* (2014) and Wang *et al.* (2010).

2.1.2 (c) Hough Circle Transform Algorithm

Hough circle transform is used for circle detection. The algorithm is able to determine the parameters of a circle when number of points that fall on the perimeter are known (Rhody, 2005). The Hough transform will identify the iris, which resemble a circle shape (Wang *et al*., 2010). Hough transform is also able to detect curves in both noise-free and noisy images with better performance as compared with other algorithms (Wang *et al*., 2010)

Parametric equations of a circle with radius R and centre (a, b) is given by Equation (2.2). The angle θ in the equation sweeps through full a 360-degree range of points (x, y) to trace the perimeter of a circle (Rhody, 2005).

$$x = a + Rcos(\theta) \qquad y = b + Rsin(\theta) \tag{2.2}$$

If the image contains many points, which many of the points fall in the perimeter of circles, the search program will find parameter triplets (a, b, R) to describe each circle. The parameter space is in 3D which makes the implementation heavy on computer memory and time. If the circle of the image contains known radius R, then the search program will become 2D. The search program will then find the (a, b) coordinates of the centres of circles.

2.1.3 Eye Gaze Tracking Algorithm

Eye gaze tracking is a very important part in this research because the whole research is to estimate the eye gaze of the user, to know the point of interest of the user on the screen and use it for HCI. A list of algorithms used to track the eye gazes by past researches is shown in Table 2.3. It is found out that there are many different approaches used by past researchers to track the eye gaze. Every researcher has their own way of tracking the eye gaze.

Table 2.3 List of eye gaze tracking algorithms used by past researches

Researchers	**Eye gaze tracking algorithm**
Kawato and Tetsutani (2004**)**	Between the eye template
Liu *et al.* (2006)	Centre of Pupil
Zhang *et al.* (2009)	Kalman filtering
Wang *et al.* (2010)	Centre of Pupil
Ebisawa and Fukumoto (2013)	Simple eyeball model
Majaranta and Bulling (2014)	Infrared Pupil-Corneal Reflection tracking
Santos *et al.* (2014)	Mean Shift procedure, Kalman filtering
Kim *et al.* (2016)	Facial and eye-gaze vectors are combined using a weighted sum
Amudha *et al*. (2016)	Genetic Algorithm based Eye Tracking (GAET) and Particle Swarm Optimization based Eye Tracking (PSOET)
Florea *et al.* (2016)	Anchoring the eyebrow

Kawato and Tetsutani (2004) uses the between the eye template to track the eye gaze. The between the eye template is to get the area between both eyes as shown in Figure 2.4. The reason why the point between the eyes is used in this algorithm is because when the face is changing the template will keep on updating. The patterns of the eye will change drastically when the user blinks their eye and this might cause the tracking point to migrate to another position (Kawato & Tetsutani, 2004). This algorithm makes use of the bright part of the nose bridge and the relatively dark part around the eyes to determine the point between the eyes. After this algorithm is applied, it provides a very good feature for obtaining an accurate location by template matching.

Figure 2.4 Between-the-eye region used by Kawato and Tetsutani (2004)

Florea *et al.* (2016) track the eye gaze by anchoring the eyebrow. The eyebrow is chosen in this case as it is shown by anchoring the eyebrow, better precision can be achieved. This is because unlike the eye, which will blink during detection, the eyebrow of the user stays the same throughout the detection process and this makes the eyebrow easier to be detected compared to the eye. This algorithm focuses on having the eyebrow to help in localizing the eye of the user and thus improves the precision in eye gaze tracking. In this research, the blink of the user is not considered so they recommend that future research to add in a blink detector. By adding a blink detector, there will not be any wrong indication provided when the user's eye is closed.

Liu *et al.* (2006) and Wang *et al.* (2010) used the centre of the eye to track the eye gaze. This is the most commonly used algorithms as most researchers will take the centre of the iris as the indicator to where the eye gaze direction is. This is because in normal condition, the centre of the eye is where the user is focusing. Kim *et al.* (2016) uses a different approach from all the past researches by considering the face into the eye gaze tracking algorithm. They detect the eye gaze through 3D approach. Their research uses two cameras with 3D depth sensor in order to be able to detect the eye gaze through 3D. First, the facial gaze is defined. Then, the eye gaze vector is defined. Next, both face and eye gaze are combined as a weighted sum. The intersection point between the gaze vector and the screen plane is calculated. Then it is transformed to screen pixel coordinates and the eye gaze tracking is performed. This is an interesting approach where the face gaze is taken into account for eye gaze tracking.

For Santos *et al.* (2014) and Zhang *et al.* (2009)apply the Kalman filter in their algorithms. The Kalman filter is used because there are small variations of pupils during the pupil detection stage. The formula for the Kalman position estimation equations is shown in Equation (2.3).

$$x_k = x_{k-1} + w_k \qquad z_k = x_k + v_k \tag{2.3}$$

In this equation, x_k is the true state, x_{k-1} is the previous state and w_k is the process noise. z_k is the observation or measure of the true state x_k and v_k is the observation noise.

By applying Kalman filter, a fixed motion model is assumed. The advantage of applying Kalman filter is that more stable results can be obtained after the filter is applied. Santos *et al.* (2014) also used the mean shift algorithm to detect the pupil. The mean shift algorithm is an image segmentation algorithm. This algorithm creates an image map through image projection based on object histogram. The algorithm will then choose the object to follow based on the map and the object position from previous image.

Genetic Algorithm based Eye Tracking (GAET) and Particle Swarm Optimization based Eye Tracking (PSOET) are two algorithms used by Amudha *et al*. (2016) for eye tracking. These techniques are based on computational intelligence. A genetic algorithm is a heuristic search method inspired by the evolution of genes.

Particle swarm is inspired by the social behaviour of bird flocking or fish schooling. However, the use of these computational intelligence increases the complexity of the algorithm.

2.2 Hardware Used in Past Research to Track Eye Gaze

Many different types of hardware are used in different researches. A low cost approach is to use a webcam as in the case of Kim *et al.* (2016) and Santos *et al.* (2014). Both researchers used webcams. Webcams are not intrusive and are more comfortable and safer to human eyes. Some researchers such as Kim *et al.* (2016) and Funes-Mora and Odobez (2016), use Kinect camera by Microsoft. Both research groups make use of the depth sensor of the Kinect camera to have a 3D model and applied eye tracking on it. Microsoft Kinect uses infrared setups on it to be able to sense the depth. There is also a special camera set up by Kocejko *et al*. (2009), which uses two cameras. One camera is a screen mounted camera while another camera is a head mounted camera as shown in Figure 2.5.

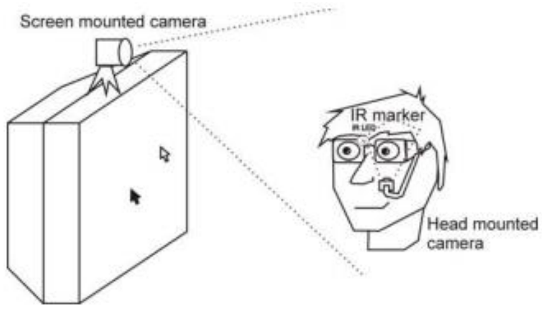

Figure 2.5 Dual camera setup by Kocejko *et al*. (2009).

Some other researchers use very high specification camera. Ebisawa and Fukumoto (2013) used a very high specification camera which is NTSC standard CCD camera, which has a near infrared sensitivity. Graovac and Markov (2016) used a commercial camera which is Nikon D3200 with a maximum recording frequency of

60 fps. With higher fps, it is expected that the more accurate the eye gaze tracking is. However, with larger fps, it will also lead to higher computer requirements as a lot of data need to be processed in a short time, in order to operate in real time mode.

2.3 Evaluation Methods for Eye Gaze Systems

There are many different types of evaluation used by past researchers to evaluate their system. In eye gaze estimation systems, the most important evaluation is the accuracy of the system. This is because if the system has low accuracy, then the information is not suitable to be used in human computer interaction.

To measure the accuracy of an eye gaze estimation system, most researchers rely on the coordinates on where the user is looking at and the gaze estimation coordinates of the system. The research by Santos *et al.* (2014) uses two testing environments to test their system, which is a pre-recorded video with 800×600 pixels resolution and a webcam real time stream with 640×480 pixels resolution. This is to compare the accuracy of the system when it is processing a pre-recorded video and the accuracy of the system when processing video which are captured in real time. In their research, it is found out that there are more errors on video captured in real time by webcam compared to the pre-recorded videos as the webcam lower resolution make their system to lose important information (Santos *et al*., 2014).

Hiley *et al.* (2006) also uses coordinates to measure the accuracy of their system. In their research, they take into consideration different ambient light conditions. Their system uses Infrared (IR) tracking thus ambient light is a concern because different light gives different IR. For example, incandescent light gives off much more IR than fluorescent light. In this research the throughput was limited to 8 gaze points per second due to the maximum frame rate that their system has. This takes roughly 125ms to display the eye gaze results in the General User Interface (GUI) (Hiley *et al*., 2006). The final result shows that the system provides sufficient accuracy for human computer interface.

A research by Amudha *et al*. (2016) also uses the coordinates to determine the accuracy of the eye gaze estimation system. The average time for prediction in each frame is recorded in this experiment. Besides, this experiment also tests the tracking results on various face orientations. Some examples of the face orientation are when the face is looking up, when the eye is blinking, where there is eye movement and when the user looks down. This research experiment includes the head movement into the consideration in the system accuracy (Amudha *et al*., 2016).

Corcoran *et al.* (2012) has a very interesting experiment that is to test their system with different platforms. Different types of processor are used to test with their system and the average time needed to track the eye gaze is recorded. This is important as their research targets for gaming design so the performance of the system when used with different processors is important for this research. Kim *et al.* (2016) also use coordinates, but this research includes the distance between the user to the monitor screen into consideration. Different screen sizes are used for this research. The distance between the screen and the user is also different based on the screen size.

The experiment setup by Kim *et al*. (2016) is shown in Figure 2.6. In Figure 2.6 there are two monitors. The small monitor displays the video taken by the webcam. For the large monitor in Figure 2.6, it indicates the area where the user should gaze at. When both areas in the small monitor and the large monitors overlapped then it means the user is looking at the correct area and this is defined as correct frames. The accuracy of the gazing area r is measured by using equation shown in (2.4).

$$accuracy_r\,(\%) = \frac{c_r}{n_r} \times 100\% \quad (2.4)$$

In Equation (2.4), the subscript *r* means the region index, *c* represents the correct frames and *n* represent the number of frames.

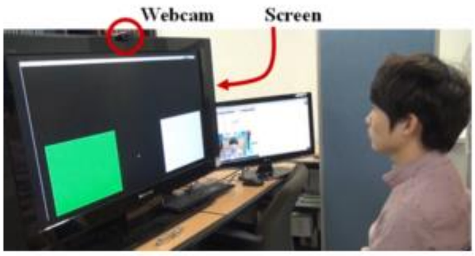

Figure 2.6 Experiment setup by Kim *et al*. (2016).

Kraichan and Pumrin (2014) evaluate their system by having three parts. The first part is the distance parameter between the user and the camera. The camera used is a built camera in a notebook. It is found that the best distance is around 60cm, where this system has the best accuracy. Next, the roll and pitch angles of the user's head pose is used to evaluate this system. The s*ensitivity* is defined using the equation in (2.5)

$$sensitivity = \frac{\#\, true\ positives}{\#true\ positves + \#\ false\ negatives} \quad (2.5)$$

In Equation (2.5), #*true positive* is defined as the correct frame of eye detection and #*false negative* is defined when the algorithm is able to detect the eye but it cannot find the center of the eye. The yaw angles are not considered in this experiment as it is not an appropriate normal head pose for human. The last part is to use the system to control the mouse.

Figure 2.7 Nine parts test area by Kocejko *et al*. (2009)

For Kocejko *et al*. (2009), hit test is used to verify their system. First, the user is asked to direct his gaze to one of the nine points as shown in Figure 2.7. Every area is lighted up for ten times. The user is asked to direct his eye gaze to the area which is lightened up and every successful attempt is recorded. A hit test result is then plotted and shown in Figure 2.8. The hit test is then performed by dividing the screen into 16 parts and then 256 parts.

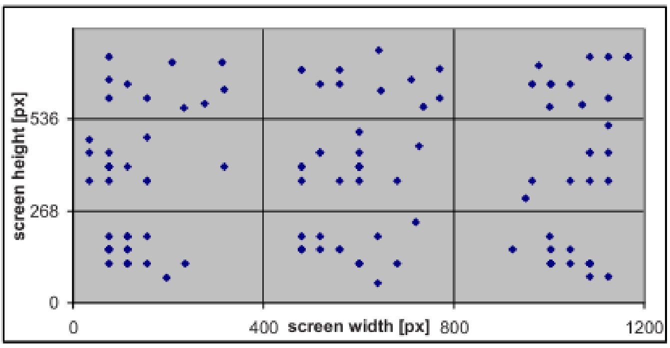

Figure 2.8 Hit test results by Kocejko *et al*. (2009)

Overall, there are many evaluations done by past researches. The most used evaluation is by using the coordinates to get the accuracy of the eye gaze estimation system. Interactive approach similar to the coordinates is by having the user to control mouse and move to the desired coordinates to calculate the accuracy. Few research also takes into considerations such as the distance between the camera and the user and the head movement of the user when determine the accuracy of their system.