Eye Tracking

Imaging and Image Processing Project

Hugh Baxter

Jason Lunn

The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

A report on a short research project investigating the implementation and accuracy of custom built software for eye tracking using a basic webcam and lighting setup

School of Physics and Astronomy
University of Nottingham

November 2019

Abstract

This project aimed to create and test a real time eye tracking system using only Python code, a standard webcam and a custom lighting rig. The software was designed to be used in conjunction with only a simple camera and lighting setup which could be replicated outside of a laboratory. The system was then tested in its ability to track and map the gaze of a person's eye across a computer screen while they look in the same direction as the camera is pointed. After a calibration process, the measured accuracy to which the gaze could be mapped along the screen was found. This data was then used to determine that potentially over 50 equally sized areas could be partitioned and defined on the screen. These individual areas could then be used to map out buttons to control input to a computer. The outcome of this project is an eye tracking system which demonstrates surprisingly accurate screen mapping capabilities using only basic tools at hand.

Contents

1 Introduction

1.1 Motivation

Using the movement and relative position of our eyes as a controlled input to a digital device creates a fluid and potentially very useful form of interface with a computer. The aim of this project was to create a real-time eye-tracking software capable of finding the position of a user's gaze on-screen for use with just a basic webcam and a mildly sophisticated lighting setup. To this end, the idea of the final product is an open-source piece of software which can be run on any PC with access to a webcam, i.e. most modern-day laptops and desktops. Assuming the face, and importantly the eyes, of the subject are lit sufficiently well (the definition of this will be detailed later on). This report will discuss the workings of the software developed in this project, its current capabilities and the problems overcome to achieve these, the aspects that continue to limit these and how they may be overcome with further development.

Specialist eye-tracking devices and software currently exists in many forms for either professional or clinical use [1]. A useful implementation of this kind of software would be to allow a screen to be used as an input interface. An example of this would be for a person with a disability limiting their functional use of a keyboard where instead each button could be mapped onto the screen and selected via gaze location. The biggest limiting factors of this implementation are the accuracy to which the gaze of the subject can be found and the workable field of view, i.e. the field of view within which gaze tracking is reliable. This field of view will be what limits the screen size that can be used while the accuracy of the system directly impacts how many buttons can be displayed on screen without incorrect inputs being registered. The size of the buttons must cover the maximum deviation of the measured gaze location to guarantee correct functionality. If the deviation of the measured gaze location for such a system is relatively large compared to the size of the screen being used, then such a system's usefulness within a user interface is limited since the number of on-screen buttons at any one time may be small.

These two main factors were investigated in order to quantify them in terms of the potential usefulness of the developed system of this project. The findings also inform what needs to be overcome to take the project to the next level.

1.2 Brief Overview of the Eye Tracking Method

Tracking a person's gaze, with the method used in this project, requires measuring eye positions before mapping them to the location of the gaze on a screen. A reference point in the eye needs to be located and tracked between video frames to track the eyes themselves. The most logical point to choose is the centre of the pupil as this is where the centre of our line of sight passes through [2]. In order to locate and track this reference point, the pupil needs to be consistently picked out from the frame of a video or live feed showing a persons face or eyes. Generally, the pupil is the darkest part of the eye region of a person's face since it is almost purely black for most people. This simplifies the processing of an image to extract the location of the pupil, assuming a well-lit face providing contrast between the pupil and the rest of the eye region. The application of a simple thresholding function to select the darkest areas of the eyes is enough to begin isolating the pupils for further processing. Once isolated, the centres of pupils can be estimated and mapped to gaze positions on a computer screen using geometry.

A more advanced method of picking out the pupil (outside the current scope of this project) involves the use of infrared lighting and cameras. Directly shining IR light at the pupil results in it becoming bright in the infrared due to reflection from the retina [3]. Unlike the case in this project, the pupil becomes one of the brightest parts of an image resulting in a very high contrast between the pupil and iris. Although IR was not involved in this project, due to lack of equipment, it is important to be aware of its use as it may make the issues with pupil detection in the project much easier to solve. Figure 1 below compares the two contrast methods.

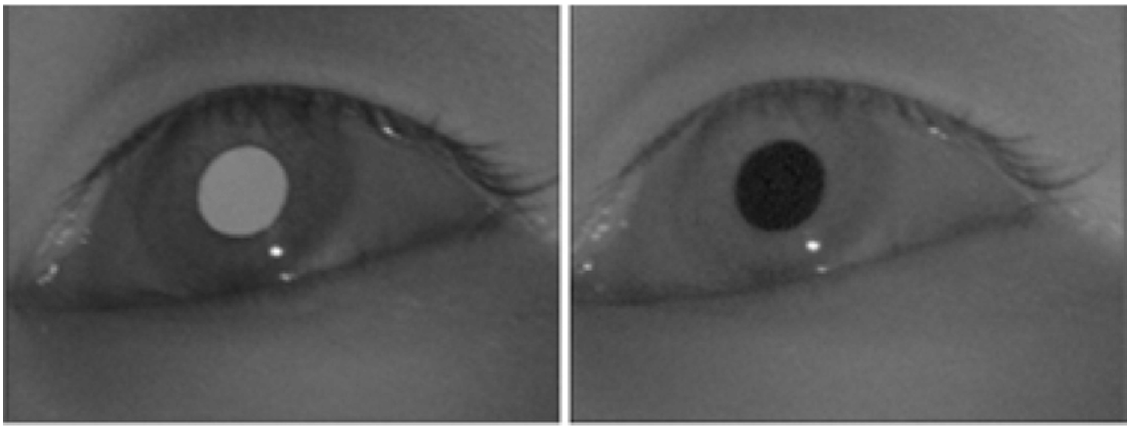


Figure 1: This figure compares two infrared images showing the contrast difference of a bright and dark pupil. The image on the right showing the dark pupil is the method of contrast that will be used in this project. The image on the left shows the effectiveness of a more advanced IR lighting and camera setup, giving a very good contrast between the pupil and background. [4]

2 Method

The workings of this project consist of two main components: the software and the experimental set-up. This section will discuss the problems faced in achieving the eye-tracking capabilities of this project, and their solutions, in both of these areas.

The most important aspects of the experimental side of the project are the computer/ camera setup and the lighting of the subject's eyes. For the software/ image-processing side, the main objectives are to isolate the pupil of each eye within an image, find its position and match this to the position of the subject's gaze on a computer screen.

2.1 Experimental Set-up

The key elements for this gaze-tracking system to function are a computer, a webcam and a controlled lighting environment.

2.1.1 Computer and Camera

As the project set out to find what was possible with readily available, cheap, equipment, the camera used to capture a person's face was a modest webcam with a resolution of 640x480 pixels. The caveat with this was that for best results the camera should be situated directly ahead of the subject's eyes (the reason behind this is discussed later in section 2.2.4), and so when running the full gaze-tracking software, the webcam was hung with string so that it sat over the centre of the computer screen, rather than sitting above the monitor in the more convenient, standard position. There is a limited area surrounding the camera within which gaze tracking is the most reliable, having the camera at the screen's centre makes more of the screen useful when gaze tracking.

When taking the results discussed later, the person having their gaze tracked was sat with their face about 45 cm away from the computer screen and webcam, with their eyes positioned directly ahead of the camera, all the while trying to remain as still as possible (even holding their breath to improve eye tracking accuracy, although this is not all that necessary for reasonable accuracy). Of course in a usable final product, the user should not have to remain still and ways of overcoming this will be discussed later.

2.1.2 Lighting

In terms of the lighting, what is desired is diffuse lighting that will give a high contrast between the pupil and iris for all positions the eyeball could be in, without having reflections coming from the cornea which obscure the pupil. Unlike the computer and webcam, this sort of lighting isn't that easy to come by and some improvisation was needed for good results.

Initially, the face and eyes were lit from either side by ordinary desk lamps, with light coming in at an angle of around 60° to the normal of the face. This angle was required to situate the reflection of the lamp to the side of the eye rather than on or near the pupil. This lighting worked well for imaging the eye while looking straight ahead, but looking towards the nose left the pupil and iris in shadow, while looking away from the nose, towards the lamp, left them obscured by reflection. Figure 2 illustrates this issue.

The problem here is that the eyeball needs to be illuminated from all sides while keeping

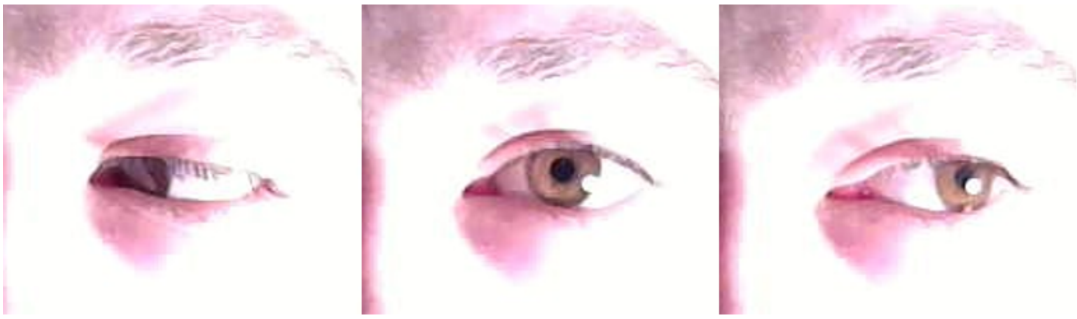


Figure 2: Example webcam stills showing the eye under lamp illumination with lamps positioned at the side. The left most image shows the eye looking towards the nose, the right most: the eye is looking away from it.

lighting from the front of the eye to a minimum. This way, the eye can be deflected by large angles without the pupil being obscured by reflections, and the pupil and iris should be well-lit when looking in all directions. Our closest solution to achieving this in the allowed time-frame was to wind some LED fairy lights into a ring to be placed around the eye. Although quite crude, this was a better way to light the eye for the desired conditions than with lamps. These rings were held in place around each eye when using the system, either by hand or with a mixture of tape and blue-tack. When taking the data discussed later in the report, only one eye was used as positioning one LED ring was easier than two (the program has modes for both single and double eye-tracking). The main benefit of having the LED ring around the eye is being able to light up the nose side of the eyeball from the direction of the nose, while also not obscuring too much of the pupil beneath reflections while looking up and down and to the sides. Figure 3 shows these benefits in webcam photo of the eye looking towards the nose when using this method.

This prototype lighting method has obvious impracticalities that will be discussed later but it does lay a good foundation for understanding how the eye should be illuminated in a final product. One prominent problem with this method faced in the project is discussed in the following section.

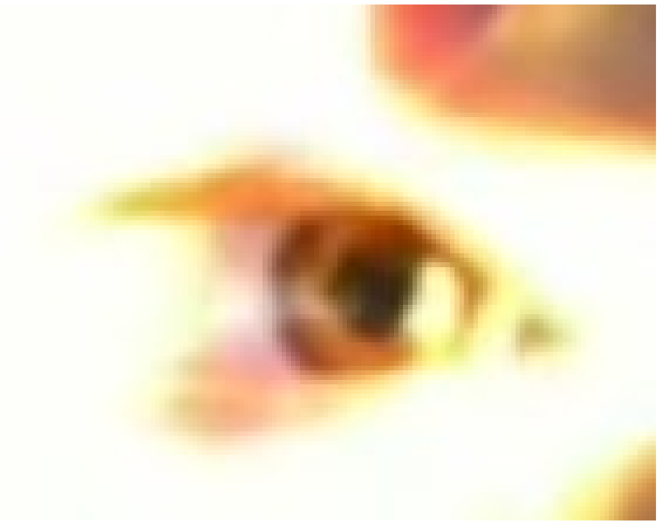


Figure 3: The eye looking towards the nose when the LED ring is used to illuminate the eyes. The low-resolution is due to this being a small region selected from a larger image.

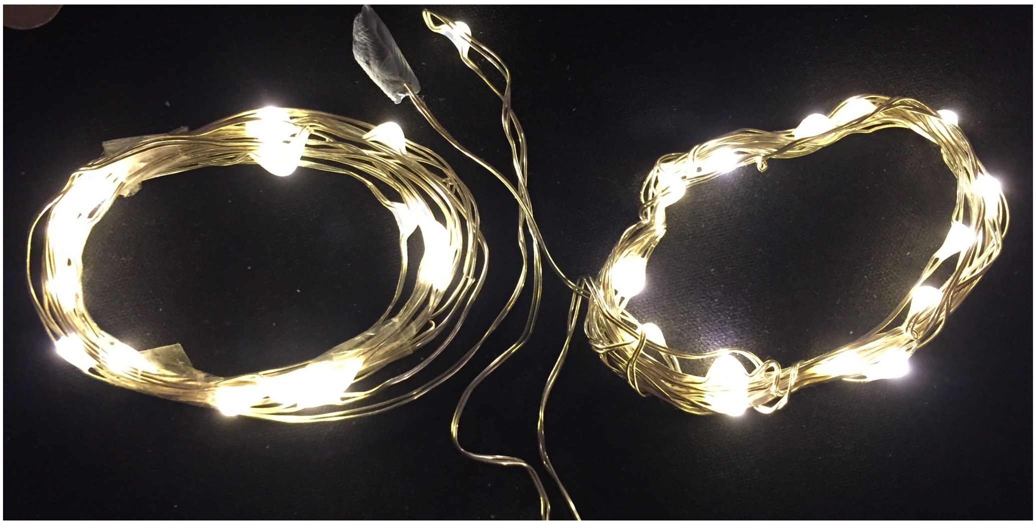


Figure 4: Improvised solution of LED rings for eye lighting.

2.2 The Software

2.2.1 Face and Eye Detection

As outlined in the introduction, the pupil is selected by thresholding. The pupil, however, is not the only dark part of the image as shadows exist both on the face and in the background. These regions can be dark enough to be selected when the entire image is thresholded, interfering with the pupil detection method. Since the region of interest (ROI) in this case is only the area containing the eyes, a simple solution to this is to only capture and threshold the region containing the eyes. One way to achieve this is by cutting out a small area of the image or frame, defined as the ROI, that is constant across all frames. However, this method makes the system inflexible as the head must always be in a specific position and scale within the video frames, with little room for movement.

A better and more automated solution to this problem is to add a Haar Cascade algorithm to the software. The Haar Cascade algorithm is a feature detection machine learning algorithm capable of very efficient and reliable detection of facial features [5]. The benefit of this method of highlighting the ROI is flexibility in where the face is located, allowing variations of head and eye position relative to the camera while still finding the correct ROI for the eyes. The software tracks the face and eyes of the subject in real-time, thresholding only the areas the Haar algorithm detects as eye regions which provides fluid pupil detection even on a moving target. An added bonus of this feature detection was its ability to detect eyes through glasses, owing to the overall versatility of the algorithm.

The Haar cascade algorithm was obtained from the OpenCV python library. The library's Haar cascade function defines separate bounding boxes for the face and each eye for every frame. The bounding boxes for the eyes are then used as ROIs for pupil detection.

One problem that was discovered when using the Haar Cascade method was false positive detection of eye regions. Face detection was found to be quite reliable, however, areas of the face such as the mouth and nostrils would often be interpreted as eyes. A fix to this problem simply involved modifying the program so that the Haar cascade algorithm only looked for eyes in the upper half of the bounding box of the face detection. This proved effective at greatly reducing incorrect eye detection.

2.2.2 Alternative Eye Detection

Because the improved method for lighting the eye involves placing the LED ring onto the face, the Haar cascade algorithm is unsuccessful at face and eye detection due to the high contrast in the images within the face. To use the program when this lighting method is being used, an alternative eye-tracking algorithm had to be made. This took advantage of the fact that as long as no other light sources are present in the camera's view, the LED rings are two of the brightest parts of the image. To select the region containing the LED rings the program thresholds the frame of each image setting pixels with the highest pixel value in the image to one, while all others become zero. To remove bright pixels outside the LED rings (which may have been selected by the thresholding) the thresholded image is opened (using the function for morphological transforms from the OpenCV library) which removes small bright features that are present. (The opening kernel which worked well for this was a 4x4 array of ones.) The region within a bounding box containing the remaining non-zero pixels is then defined as the region containing both eyes. Within this, ROIs are defined for the right and left eye by taking a rectangular region within the left and right half of the main bounding box. The size of these boxes was chosen to be about a quarter of the size of the larger box surrounding both eyes as this was large enough to surround the eye without there being any dark regions from outside the LED ring in the corners of the box. Figure 3 is an example of what is found within the eye bounding box.

The ROIs containing the eyes, selected from either one of these detection methods are to be further processed to identify the location of the pupil within them.

2.2.3 Pupil Selection

Now that regions of the image corresponding to the subject's eyes have been found by the program, these regions can be processed separately to find the location of the centre of the pupil within each eye image.

As the pupil is the darkest part of the eye, the best way to begin is by thresholding the greyscale version of the image of the eye to select dark regions. Of course, the pupil will not be the only dark part of the image so it is important to get the threshold level as dark as possible to select only the pupil. Initially, a single threshold value was chosen with a slider for each eye. The problem with this is that for images of the eye when the eye is in different positions, the pupil in one image may be much brighter compared to that of another. I.e. the pupil in one image could be much darker than all other parts of the image and would be easily selected by thresholding with a value above the pupil-pixel values; in a different image, the pupil may lie in a shadow, only being slightly darker than surrounding regions. Applying the same threshold to the latter image as the first may also select surrounding pixels which do not lie in the pupil. In the context of processing video frames, this means a single threshold is not a good choice. Instead, the program of this project implements a dynamic threshold to select only the darkest pixels of each frame. The threshold value is set as the nth lowest pixel value of the image in question. The slider is used to control the value of n for setting this thresholding.

With this dynamic threshold, there will still be cases where other pixels outside the pupil are dark enough to lie beneath the thresholding value. These pixels are removed from the pixel selection using contours. Contours within the thresholded image (the boundaries between light and dark) are calculated and only pixels which belonged within the largest contour (that of the pupil) are selected.

Before thresholding an image, the program blurs it to remove any noise (using a Gaussian filter with a kernel of 7x7 pixels (this choice was made by judging blurred eye images by eye and is dependent on image resolution)). The potential noise being reflections in the eye/pupil. The problem with this is that blurring makes the pupil less well defined as seen in figure 5, so when it has a shadow nearby, the pupil and shadow may merge together, distorting the pupil as seen by the program in the thresholded image. To illustrate, figure 6 shows the contours from the binarised image plotted over the original image with and without blurring before thresholding. A branching shadow is connected to the pupil when the image has been blurred before thresholding.

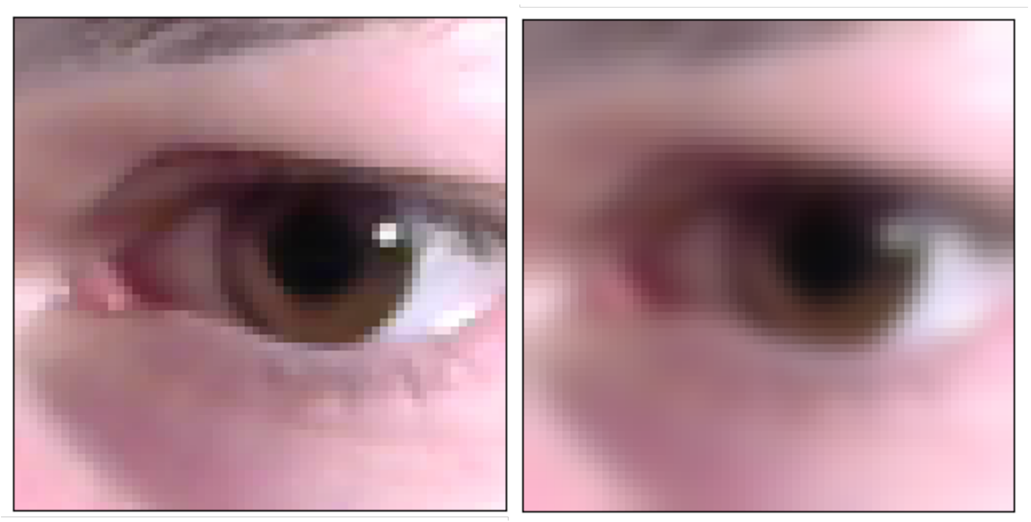


Figure 5: Image of the eye before and after blurring.

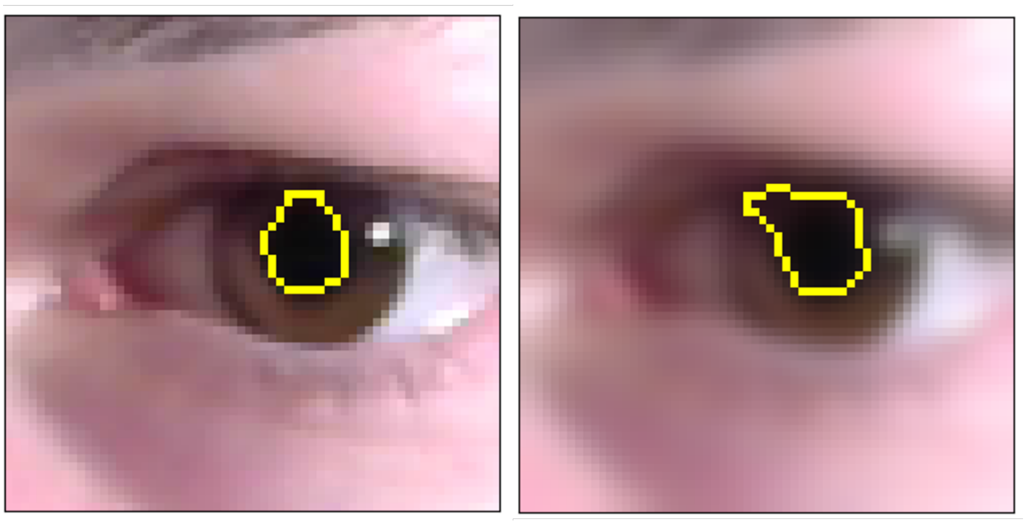


Figure 6: Image of the eye before and after blurring with contours of the pupil selection plotted over the top.

To solve this problem, while still being able to remove noise using blurring, a process was implemented whereby the image is thresholded both with and without blurring, each with a different user-defined threshold value for each eye (so that the boundary of the pupil is better defined without). Only pixels lying within the contour around the pupil are selected, as before. The bounding box of the pupil's contour in the non-blurred image (the smallest rectangle the contour can fit into) is then used to define the region (a bounding box) which contains only the pupil. The thresholded image obtained when the image was blurred then has a selection made corresponding to the region within the bounding box defined by the non-blurred image. This way, any extra appendages the pupil may be found to have in a blurred image are removed and noise within the pupil is too.

An alternative method which was considered to solve the above problem is to perform opening on the thresholded image, as any small features extending from the pupil would be removed. This process, however, was found to be too aggressive and removes a lot of information about the size and shape of the pupil.

Now with the program being able to select the pupil, the location of the centre of the pupil within the video frame is estimated as the centroid of the pixels (belonging to the pupil) in the thresholded image.

An alternative way to measure the centre of the pupil is to fit an ellipse to the contour of the pupil, found from the thresholded image, and use its centre as the estimate for the pupil's centre. This was found to work well for high-resolution images, but at lower resolution images, such as those produced from the 480p webcam used for the live tracking, (as seen in the figures of this section) the centroid estimate was found to be more accurate than the ellipse estimate. This is because it is difficult to fit an accurate ellipse to a pixelated eye image. This is discussed properly in section 3.1.2.

2.2.4 Mapping Pupil Positions to the Gaze Position On-screen

With the centre of the pupil located within the image, the next job is for the program to map this to the on-screen position of where the subject is looking: the gaze position.

Consider the case in figure 7 with the eyeball being located at the origin in spherical polar coordinates, with the z-axis aligning itself between the eyeball and the screen; the screen being normal to the z-axis. The vector of the centre of the pupil is given by r_p, θ, ϕ. θ and π are both zero when the eye looks directly ahead, down the z-axis.

The equation relating the distance of the pupil from the z-axis p to the distance of the gaze position s relative to the z-axis can be derived from the diagram in figure 7 as

$$s^2 = \frac{d^2}{(\frac{r_e}{p})^2 - 1} \tag{1}$$

where r_e is the radius of the eyeball. For small angular deflections of the eye in θ, this equation can be approximated to

$$s = \frac{dp}{r_e} \equiv \alpha p \tag{2}$$

which is linear. α is a parameter which is found by the program through calibration, as the parameters d and r_e cannot be measured with the experimental setup.

When experimenting, the distance from eye to the screen was roughly 50 cm and the maximum gaze distance s worked within was about 12cm from the z-axis. This equates to a maximum angle of eye deflection of around 0.2: close to the limit within which this linear approximation holds.

The equations above refer to the location of the pupil centre projected onto the x-y plane in real space p, i.e. the position of the pupil within an orthographically projected image of the 3D scene. The photographic image taken by the camera, however, will not be an orthographic projection of the scene. This is why it is important to locate the camera directly ahead of the eyes, in front of the screen (as in the diagram), as the image near the centre of the camera's field of view will be close to an orthogonal projection making the equations above viable for mapping the on-screen gaze location.

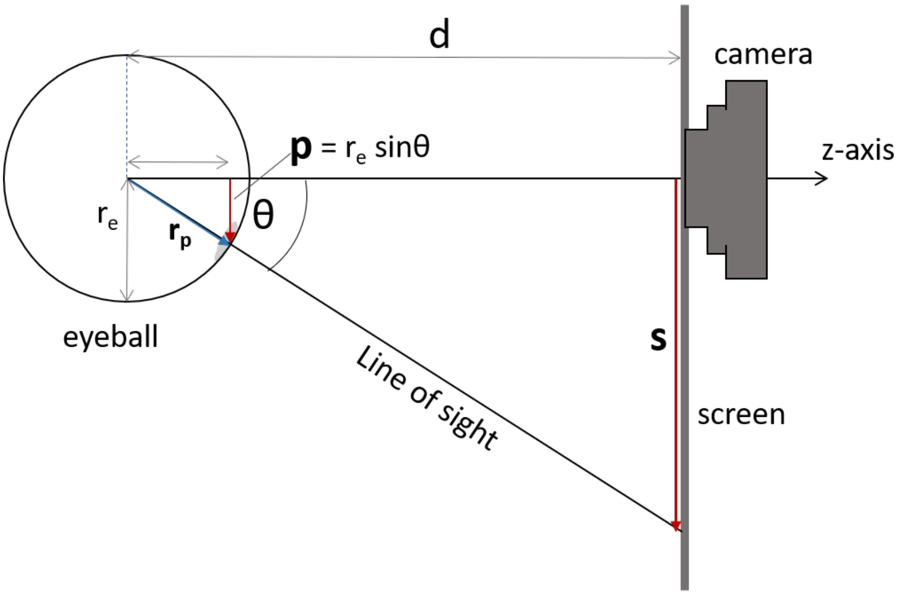


Figure 7: Diagram showing the parameters involved in mapping the pupil position relative to the z-axis p to the gaze position on screen s with the angular deflection of the eye relative to the z-axis being $theta$, and the distance from eye to screen being d.

2.2.5 Calibration

A calibration process was used to calculate α in equation 2. This entails the user to look at the four pips in the corners of a rectangle on the screen and press a key while looking at each pip so that the program knows where on the screen the user is looking. The rectangle of the calibration image is centred such that the webcam is sitting at its centre. All the while the program is determining the location of the centre of the pupil in each eye within the video frame. The measured pupil locations while looking at each pip of the on-screen rectangle also form the corners of a rectangle in what will be referred to as pupil-location-space, due to the linear relation of equation 2. A rectangle is fitted to the measured pupil locations and the scale and offset between this rectangle and the on-screen rectangle is calculated in both the vertical and horizontal directions to give α_x, α_y and O_x, O_y. The locations of pupils in the video frame are then offset and scaled in the x and y directions using

$$s_a = \alpha(p_a - O_a) \tag{3}$$

where a is x or y, to give the on-screen position of the user's gaze; this is used to displace a blue and a red dot at the gaze locations for the left and right eye in real-time, on the calibration screen. Note that p and s are measured in pixels of the video frame and of the screen respectively.

Only four pips in the corner of one rectangle were used for calibration, although, the program can be asked to use more than four calibration pips along the edges and corners of a rectangle. There was no noticeable improvement in the accuracy of the gaze location mapping when doing this, hence why only four pips were used (this was qualitatively judged by the subject rather than being quantified).

Before calibration, however, it is necessary to fine-tune the thresholding parameters for the pupil selection process explained in section 2.2.3. For this, the program displays a real-time image of the user's eye with a rectangle showing the bounding box for the primary pupil selection (the selection made from thresholding the non-blurred image), with an outline of the secondary pupil selection within that box (the selection from the blurred thresholded image). The thresholding parameters can then be adjusted using sliders for the primary and for the secondary pupil selection processes, with the eye image updating in real-time. The user should choose the values that best select the pupil for all eye positions. This can be checked by the user moving the eye image around the screen, looking at it to see how the pupil is selected when they look at those different on-screen positions. Figure 8 shows an example of the image the user sees.

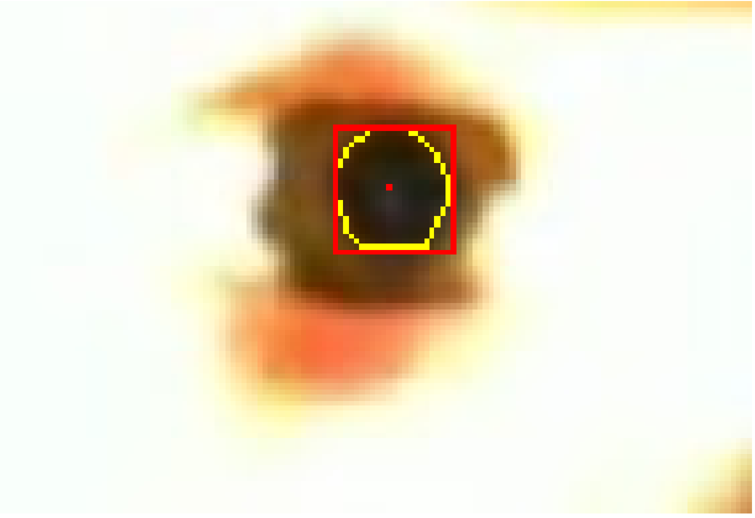


Figure 8: Example image displayed to the user of their eye in real time showing the pupil detection. The red box shows the pupil selection from the primary pupil selection, the yellow line outlines the final pupil selection and the red dot marks the centre of the pupil calculated from finding the centroid of the pupil selection.

3 Limitations and Results

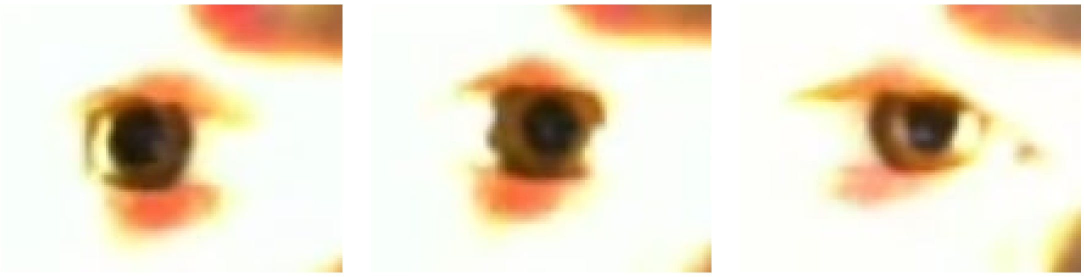
3.1 Limitations on Pupil Centre Detection

The basis of this eye tracking system revolves around locating the point at which the gaze of a person originates. We have defined this point to be the centre of the pupil. The technical challenge that then arises is to accurately locate the pupil centre in each image or frame of a video. The camera used the capture the data and the lighting used to illuminate the subject are the limiting factors in this case. The point at which the software and image processing techniques employed have measurable effect on the tracking accuracy is far beyond the physical limits for this project. Given below are details on how the camera quality impacts the centre locating accuracy, as well as how the angle of deflection of the eye away from the camera affects the error.

Initially it was thought a fitted ellipse would provide the most accurate way of determining the pupil centre since the pupil itself can be modelled as an ellipse. However, the sections below along with plots give a quantified confirmation that this is in fact not true. The centroid of the contour (the threshold centroid) is in fact consistently more accurate to the true centre. This is the reason for abandoning ellipse fitting in favour of other techniques.

3.1.1 Effect of Eye Deflection

The shape of the pupil in two dimensions can be modelled as an ellipse when viewed and imaged at a head-on angle. As the eye moves, the shape of this ellipse is changed as more or less of the pupil can be seen by the camera. This section outlines a test done using images of a single eye looking at horizontal intervals measured across a computer screen. The camera was placed at the centre of the screen and the images were taken at measured horizontal distances (in centimetres) either side of the camera. Figure 9 shows a sample of these images, all taken at the same resolution with a webcam. The LED light ring lighting was used to provide the best contrast between the pupil and the rest of the eye.



(a) Looking at a point 20 cm away from the origin on screen moving away from the nose.

(b) Looking directly at the camera at the origin point on screen.

(c) Looking at a point 20 cm away from the origin on screen moving towards the nose.

Figure 9: The images above were taken using the LED light ring setup, with a webcam moved to the very centre of a computer screen. Measured points horizontally moving outwards from the camera were then looked at with images taken. This was done to measure the effect on a single eye when looking towards or away from the nose, as well as see the general trend as the gaze moves further away from the origin.

Using a manual variable thresholding technique on the images, a single threshold value best for all images was found as outlined in Section 2.2.5. The centre of a fitted ellipse and a contour centroid was then found for each image. A manually chosen centre of the pupil was also plotted on every image, this will be used to define the "true" centre of the pupil as we are assuming the accuracy a human can deduce the centre is a higher standard which we can then compare the algorithms to.

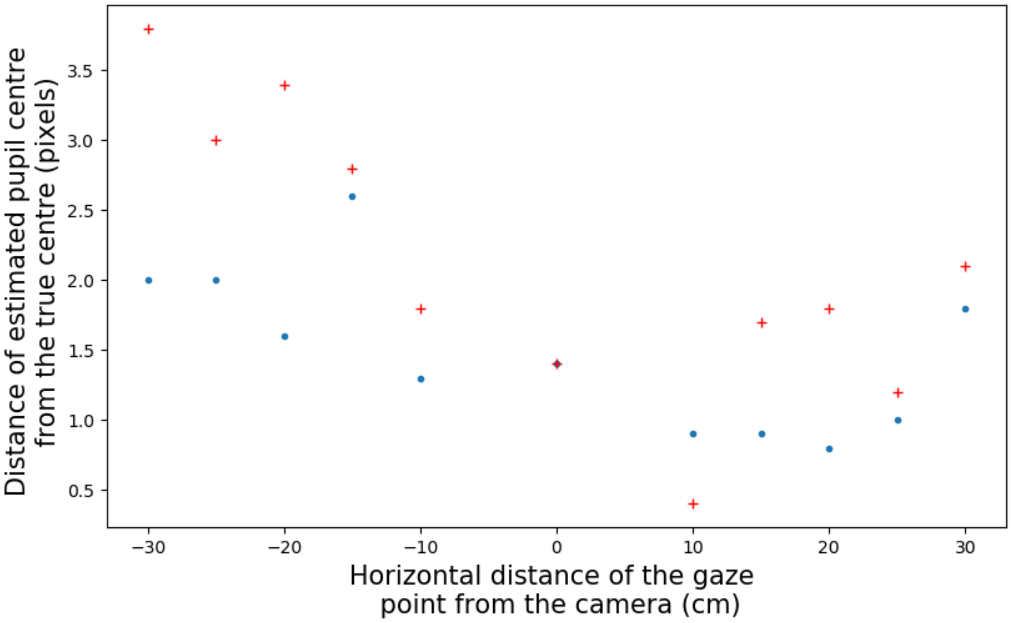


Figure 10: This scatter plot shows how the accuracy of the pupil estimation by different methods is affected by the deflection of the eye relative to the camera. The red crosses represent an ellipse fitted centre, whereas the blue dots represent the centre of the contour around the threshold. The negative values on the x-axis represent the eye being deflected towards the nose, whereas the positive is deflection away. The zero pixel value on the y-axis is a representation of the manually chosen true centre.

As Figure 10 shows, when the eye is deflected towards the nose (negative values) the accuracy of pupil centre detection is worse relative to looking away from the nose (positive values). Generally the plot shows the ellipse fitted centres (red crosses) are less accurate to the true centre when compared to the threshold centroids (blue dots), which is as we would expect from Section 2.2.3. From a comparison of the original data and the plot, it can be deduced that the accuracy improves when looking away from the nose due to the lessening of the reflection of light coming from the bridge of the nose and obscuring part of the iris as shown in Figure 9c.

It could be argued this data shows a limitation with the lighting rather than the angle of eye deflection. The shape of the eye and eyelid however are not the same when looking left or right. The images (a) and (c) in Figure 9 demonstrate this as the reflection interrupts the pupil more when looking towards the nose. This is caused by the bridge of the nose forcing the light ring to sit further forward on the face compared to the other side alters the reflection of light on the eye. Clearly the bigger factor at play here is the non-symmetrical geometry when considering only one eye. This impacts reflections on the eye and in turn alters the accuracy at which the software can find the pupil centre.

This test was only carried out looking at horizontal eye displacements. A prediction is difficult to make from this of how vertical displacements would change accuracy. This again comes down to the geometry of facial features. Our eyes are not perfect circles viewed head on in two dimensions. The eyelids block the pupil at extreme vertical deflections which would potentially cause a larger deviation in accuracy than that of extreme horizontal deflection. This test would need to be carried out with a similar setup and the data compared to draw any solid conclusions. Unfortunately, the scope of this project prevented this comparison, although it would be an important test for a more advanced eye tracking system to undergo.

3.1.2 Effect of Image Resolution

By taking a relatively high-resolution image of an eye and then incrementally lowering the resolution, the effect of pixel density at the eye on the measured pupil centre location accuracy can be found. Using a relatively high-resolution modern laptop camera, an image of a face with one eye illuminated to provide best pupil contrast was captured. The region of interest (the eye) was then extracted and tested on. Figure 11 below shows the original resolution region of interest image that was used to collect this section's data.

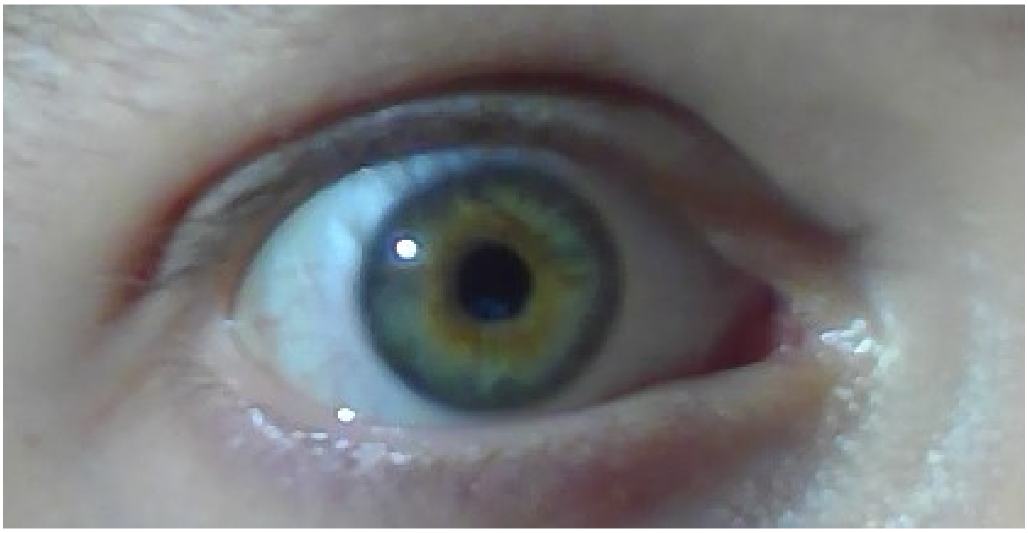


Figure 11: This image was captured using a single close diffuse source of light to provide a good pupil to background contrast. An important feature of this image is the reflection from the light source in the eye is well outside the pupil region to ensure good pupil detection results.

Since this image is a cropped version of the original containing the full face, the resolution is not at a maximum that the camera could theoretically capture of the eye. One limit preventing this is the cameras focal length, i.e. keeping the image in focus to retain information, preventing a close eye image being taken. This limit is beneficial to the results collected however, since our setup does not rely on an up-close imaging of the eye. The subject is meant to be at a reasonable distance from the camera and screen. The resolution of this region of interest better reflects a scenario this system is built to handle. The test below yields data closely applicable to the real time webcam eye tracking which is exactly what we want. The variable thresholding method and manual "true" centre location technique discussed previously was used on the image, which was then resized to incrementally lower resolutions. Both the ellipse fitted centre and threshold centroid were determined to again check if the theory the ellipse is a worse fit holds true.

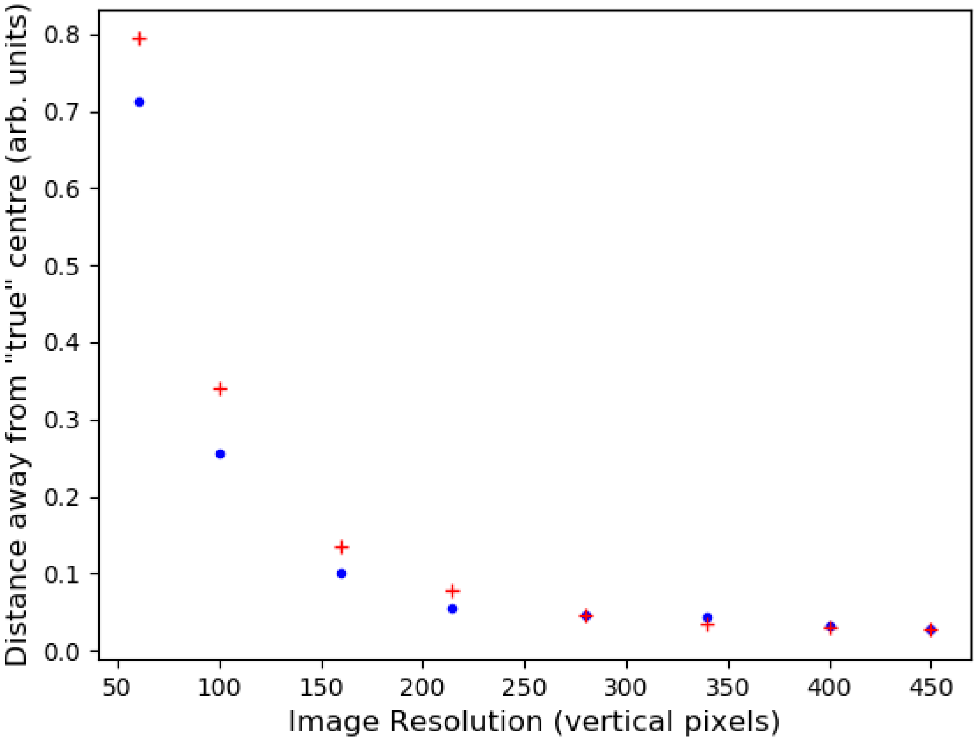


Figure 12: This plot shows the relative change in accuracy for finding the pupil centre in the same image at varying resolutions. As before, the red crosses represent an ellipse fitted centre, whereas the blue dots represent the centre of the contour around the threshold. The y-axis is scaled in arbitrary units since the data is only meaningful when the resolutions are compared. The x-axis plots the resolution of the eye image in units of vertical pixels.

Figure 12 above shows a clear strong correlation between the region of interest image resolution and the accuracy of pupil centre detection. The unit of vertical pixels has been used on the x-axis since this is the most intuitive way to quantify resolution, a standard video format of 720p or 1080p for example refers to the maximum amount of vertical pixels in the video. These results are very good for the method we have employed here since the accuracy only becomes noticeably affected for relatively low-resolutions compared to a basic webcam setup. This in turn means a higher resolution camera only provides diminishing returns in the way of pupil centre tracking accuracy at least for small deflections of the eye. Our system then works well within the confides of basic low-resolution webcams.

By scaling up the region of interest image to find an approximate size of the original image, a lower limit can be found for the required webcam resolution before accuracy is significantly impacted. The resulting resolution comes out close to 720p. A more generous lower resolution of 480p is still within an okay accuracy which is ideal as many basic webcams stream video feeds at this resolution, like the main one used throughout this project. Assuming the use of a webcam with better video resolution of 720p or 1080p which are much more common these days, built into laptops for example, these resolutions will not at all be detrimental to the accuracy.

3.2 Gaze Tracking Accuracy

Having discussed in the accuracy of the measurements of the centre pupil centre, the accuracy of the measurements of the on-screen gaze position will be discussed. Once the system is calibrated by the user focusing their gaze on the pips of a calibration image, the estimated position of their gaze for each eye is displayed as circles in real-time on this image. To quantify the accuracy of the gaze-tracking, an experiment was done where the screen presented an image similar to the calibration image, with pips in multiple locations. The subject focused their gaze on a number of pips for a certain amount of time for each pip while the program tracked their gaze. The mean estimated gaze location for the time that the subject was focused on a pip was compared to the location of the pip in question (the location of the true gaze) to obtain a measure for the accuracy. Only the right eye of the subject was tracked to make the test simpler (as only one LED ring could be held to the face), although it would be interesting to repeat the test with both eyes to see if there is a difference in accuracy between the two eyes.

Figure 13 shows the on-screen image the subject was presented with, with labels for each pip. Figure 14 shows this image with the estimated gaze locations (measured over time) for each pip plotted in a different colour. The gaze positions are associated with the nearest pip to them. From this image, it is already quite clear how accurate the gaze-tracking system is simply by looking at the positions of the estimated gaze points relative to the pips. Figure 15 displays a plot of the distance between the mean estimated gaze location and true gaze location, along with the standard deviation, for each pip. The mean of the mean distance of the estimated gaze from the true gaze across all pips is 8mm and the mean of the standard deviations of the distances across all pips is 5mm.

When taking this data, the subject tried to remain as still as possible, including holding their breath in order to increase the gaze tracking accuracy. It also should be noted that the time spent looking at each pip was not exactly the same so some pip locations have more data than others. This needs to be considered when drawing any conclusions from comparing the data across the pip locations.

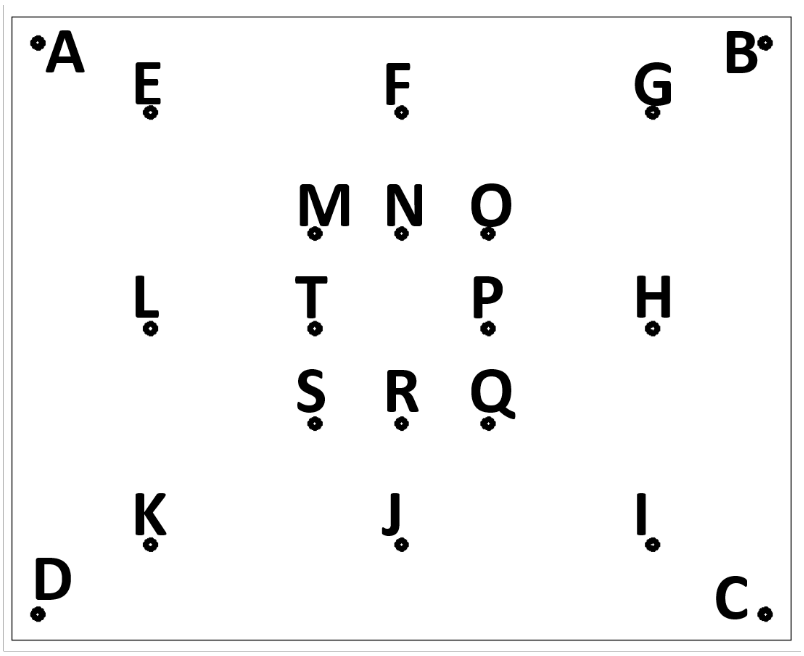


Figure 13: On-screen image the subject was presented with. Labels have been added to each pip.

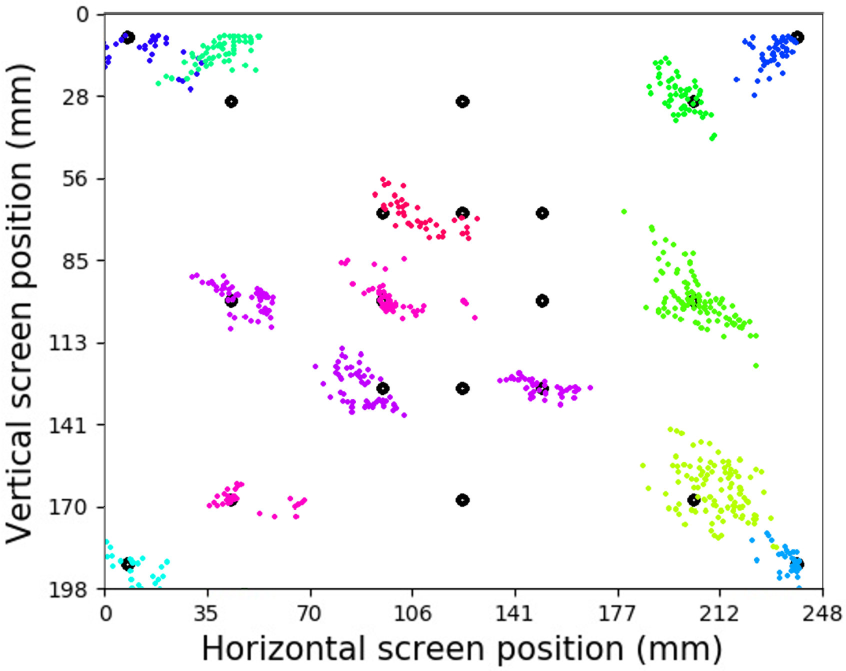


Figure 14: On-screen image the subject was presented with with the estimated gaze locations (measured over time) for each pip plotted in a different colour

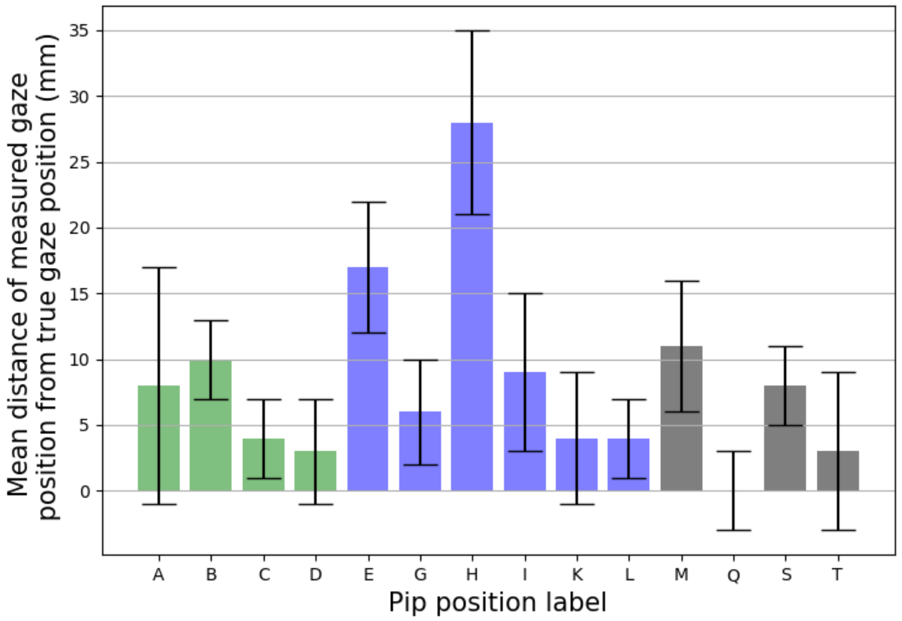


Figure 15: Distance between the mean estimated gaze location and true gaze location for each pip. The error bars show the standard deviation of the gaze position. Pips in the outer region of the image are given green bars, pips in the mid-region: blue, and pips near the centre of the image (nearest the camera): black.

4 Discussion

4.1 Discussion of Gaze Accuracy Results

The results from the experiment of section 3.2 show the system to be very accurate at tracking someone's gaze considering the simplicity of the equipment being used to track the eyes. The 25 x 20 cm area of the screen worked within equates to a field of view with an angle in the horizontal and in the vertical of about 30° and 25° respectively. As the gaze-tracking was relatively accurate, this field of view acts as a lower limit on the size of the workable field of with the program and set-up used. The 30° and 25° field of view was chosen so that calibration was done in a region of the screen equating to small angular deflections of the eyes, so as to fit with the linear approximation used for mapping pupil positions to gaze positions. The subject's gaze was only tracked within the field of view defined by the calibration data, however, it would be good if the program was made to track the eyes outside of the calibration region, giving a larger field of view to take data in.

It would be interesting to repeat this experiment again, taking more, and equal amounts of data for more screen positions, both inside and outside the calibration field of view, using a proper system to support the head so that it is stationary. This information could then be used to create a map of the accuracy at different screen positions in both the x and y directions, i.e. a measure of accuracy with field of view. This could inform us about the true size of the workable field of view and about how good the linear function of equation 2 is for mapping pupil positions to screen position (assuming that a method for accurately measuring pupil positions at larger angular deflections of the eye, see section 3.1.1). It should be that at larger distances from the camera, the accuracy drops due to the linear approximation of the mapping function only holding for small angles of eye deflection as discussed in section 2.2.4.

As the experimentation in this project has given an idea of the accuracy of the gaze-tracking system developed, its potential usefulness can be inferred (given that the system is made more user-friendly, i.e. allowing the user to move). From the experiment of section 3.2 The mean inaccuracy (measured as the distance between the mean estimated gaze position and the true position) across all pips was 8mm and the mean of the standard deviations of the distances across all pips was 5mm. If the system was developed to be used to control on-screen buttons with someone's gaze, this means the buttons would have to be about 3x3cm (from adding the mean inaccuracy with the mean standard deviation). This ignores the fact that the accuracy in the x and y directions are probably different and it assumes that the program requires the user to stare at a button for long enough before being pressed so that the program is sure which button has been selected. The same eye to screen distance as used in the experiment, of about 45 cm is also assumed, for this button size. So with a more reasonable eye to screen distance of around a meter for a desktop setup, this button size would be more like 6x6 cm (this is as long as the resolution of eye images at this distance is still good). For the same field of view as the calibration field of view of the experiment, potentially, over 50 buttons could be displayed on a roughly 50x40 cm screen at once: a system using the same fundamental method as that of this project should be able to tell which button is the desired button.

4.2 Potential Improvements Using Infrared (IR) Lighting

It is generally true of everyone's eyes that the pupil is the darkest region relative to the entire eye as discussed previously. The iris surrounding the pupil, however, is not the same colour for everyone. A brown or hazel coloured iris has less contrast to the pupil when compared to a green or blue one. Both green and brown irises were used throughout the development of this project (a brown iris was used for the experiment which measured the accuracy of the gaze-tracking), and it was noted that with less than ideal lighting, the eye tracker was better at locating the pupil with the green iris. Although not quantified in this report, this behaviour assumed to be down to the contrast being larger between a lighter coloured iris and the dark pupil. Improving and removing the colour discrepancy could be achieved using an infrared lighting and camera setup.

As discussed in the introduction, using IR light to illuminate the pupil provides better contrast than that of a dark pupil. Equipment limitations prevented the use of this technique in this project, but a significant improvement on the pupil centre detection via thresholding would be made with the use of an IR camera and lighting setup. Not only would thresholding give more consistent pupil centre results with a bright pupil, it would also eliminate the issues involving reflections on the pupil from optical light. This issue has been a common problem throughout this project (requiring lighting set-ups were little light should come from in front of the eye) so it is clear that an IR eye-tracking method already indicates a superior methodology. Were this setup possible for this project, it would certainly be a priority over any optical lighting scenario.

4.3 Further Development for a User-Friendly System

The current limitations to how user-friendly the system is are quite clear. For accurate tracking, the user must hold a crude ring of LEDs to their eye, keep as still as possible, and look at a screen with a webcam hanging over the middle, all in a somewhat dimly lit room. These are all things which could be overcome given time for further development.

For example, instead of holding very still, the program could be developed to track a specific point on the face, perhaps a dot positioned on the forehead. This could be used as a reference point for locating pupils relative to the face rather than to the camera. The theory behind this being that movement of the eyes relative to the head will be negligible with small head movements while tracking a stationary object. Alternatively, a camera could be attached to the head with some headgear.

Rather than holding LEDs to the eye which is inconvenient in both having to hold them and having bright light shone at your eyes, infrared LEDs attached to a pair of goggles or glasses could be worn to work with an infrared camera.

A more complex function could be used to map pupil locations to gaze positions rather than the linear approximation. This way the camera could sit above the screen as is normal.

4.4 Other General Observations

The scope of this project has also limited the definition of the gaze of a person down to the movement of one pupil centre at a time. The reality is however, our gaze is the combination of both of our eyes working together. An advancement to the system created in this project would be moving from mapping the movement of one pupil centre on the screen, to finding a mathematical/trigonometrical method to combine both of the pupil centres movements into a true gaze location for both eyes at once.

5 Conclusions

In this project, simple equipment consisting of a webcam, computer and some LEDs have been used to create an eye-tracking system potentially capable of being developed into a computer interfacing system able to hold over 50 buttons on a screen at one time.

The main limitation to overcome before the result of this project can be developed into such a system are that the user is required to remain still while holding rings of LEDs held over their eyes in a dimly limit room with a webcam in an unusual position. Ways in which these limitations can be removed have been discussed, and the findings point towards a system which uses IR light rather than optical, for the benefit of both improved accuracy in eye tracking, and for the comfort of the user.

References

[1] *Glover PM. et al (2014) "A dynamic model of the eye nystagmus response to high magnetic fields" Physics in Medicine & Biology 59:631–645*

[2] *Krejtz K., Duchowski AT., Niedzielska A., Biele C., Krejtz I. (2018) "Eye tracking cognitive load using pupil diameter and microsaccades with fixed gaze." PLOS ONE, 13(9).*

[3] *Almeida S., Veloso A., Roque L., Mealha O. (2011) "The Eyes and Games: A Survey of Visual Attention and Eye Tracking Input in Video Games." 10.13140/RG.2.1.2341.3527.*

[4] *Harezlak K. & Kasprowski P. (2017) "Application of eye tracking in medicine: A survey, research issues and challenges." Computerized Medical Imaging and Graphics, 65.*

[5] *Viola P. & Jones M. (2001) "Rapid Object Detection Using a Boosted Cascade of Simple Features." IEEE Conference on Computer Vision and Pattern Recognition, 1:I–511–I–518*