UNIVERSITY OF COIMBRA

FACULTY OF SCIENCES AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

DEVELOPMENT OF AN EYE-TRACKER FOR A HMD

JOSÉ PEDRO NUNES BARATA FRANCISCO

JURÍ: PAULO JOSÉ MONTEIRO PEIXOTO (PRESIDENTE JÚRI), JORGE NUNO DE ALMEIDA E SOUSA ALMADA LOBO, PAULO JORGE CARVALHO MENEZES

24 FEBRUARY 2017

UNIVERSITY OF COIMBRA

FACULTY OF SCIENCES AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

DEVELOPMENT OF AN EYE-TRACKER FOR A HMD

Thesis submitted to the Electrical and Computer Engineering Department of the Faculty of Science and Technology of the University of Coimbra (DEEC-UC) in partial fulfilment of the requirements for the Degree of Master of Science.

Under supervision:

Paulo Menezes, PhD, Professor at DEEC-UC

Bruno Patrão, MSc

JOSÉ PEDRO NUNES BARATA FRANCISCO

jfrancisco@isr.uc.pt

COIMBRA

2017

HÁ MINHA FAMÍLIA E AMIGOS.

Abstract

Head mounted displays (HMD), which are becoming available at accessible prices, are attracting the attention of researchers for their possible application beyond the traditional game use, namely for the development of systems capable of aiding the therapist in therapeutic procedures. Psychology is one of the areas that could benefit from this, especially in exposure-based therapies. Where patients are exposed to phobia inducing elements intended to activate a emotional response, this type of therapy is typically used for the treatment of phobias or post traumatic stress disorders. As such, monitoring of the emotional state of the patient is very important for the therapist. And these immersive systems can provide the automatic recognition of emotions from body posture, bio-signals and face expressions. However, the use of a head mounted display, that is a key ingredient for immersive virtual reality, partially precludes the observation of facial expressions. And as such is difficult to access the facial expression of the patient, and more importantly to determine where the user is looking at in the virtual world. Has the patient closed it's eyes when faced with a phobia inducing elements? Has the patient looked way from those elements? In order to answer this questions an eye-tracker can be used inside the head mounted display to provide real-time information to the therapist. On the market there are some commercial solutions that provide eye-tracking capabilities for the most popular head mounted displays, but do so at a prohibitively high cost for most researchers, and require extensive modification of the units. On this thesis I present a low cost solution to track the users gaze inside the popular Oculus Rift development kit 2, that requires minimal technical skills to assemble and comes with a software interface aimed at providing a simple to use platform for gaze analysis. This solution comprises of two spacers with two endoscopy cameras, placed behind the lenses of the helmet, that allow monitoring the eyes of the user wearing the helmet. On this thesis I discuss the reasons that led me to this design, and I also talk about the development of the software that accompanies it, specially the calibration of the users gaze. One of the applications of this eye-tracking platform is to complement bio-signal data to detect emotional activations, and also to enable the understanding of where the user is looking at in the scene and if he/she flees away from phobic element.

Keywords: Eye-Tracking, Head Mounted Display, Virtual Reality, Computer Vision.

Resumo

Capacetes de realidade virtual, dada a sua recente disponibilidade a preços reduzidos, estão a atrair a atenção de investigadores pela possibilidade da sua utilização para além do uso tradicional nos videojogos, nomeadamente no desenvolvimento de sistemas capazes de ajudar o terapeuta em procedimentos médicos. A psicologia é uma dessas áreas que poderia beneficiar disto, especialmente em terapias de exposição. Onde os pacientes são expostos a elementos indutores de fobia com o propósito de provocar uma resposta emocional, este tipo de terapia é tipicamente usada para o tratamento de fobias e stress pós-traumático. Como tal monitorizar o estado emocional do paciente é muito importante para o terapeuta. Estes sistemas imersivos podem proporcionar o reconhecimento automático das emoções, desde a postura corporal, bio sinais e expressões faciais. Contudo, o uso de capacetes de realidade virtual, que são um ingrediente fundamental para realidade virtual imersiva, impedem parcialmente a observação das expressões faciais do paciente. Como tal é difícil visualizar a expressão facial do paciente, e mais importante, determinar onde é que o paciente está a olhar no mundo virtual. Será que o paciente fechou os olhos quando confrontado com um elemento indutor de fobia? Será que o paciente olhou para esses elementos? Uma maneira de responder a essas perguntas é usar um seguidor ocular dentro do capacete de realidade virtual, de modo a fornecer informação em tempo real ao terapeuta. No mercado actual existem algumas soluções comerciais que oferecem capacidades de seguimento ocular aos mais populares capacetes de realidade virtual, mas fazem-no a preços proibitivos para a maioria dos investigadores, requerendo modificações extensivas dos capacetes. Nesta tese apresento uma solução de baixo custo para fazer o seguimento ocular de um utilizador dentro do popular Oculus Rift Dk2, que requer habilidades técnicas mínimas para montar e que inclui uma interface de software que visa proporcionar uma plataforma simples para análise do seguimento ocular. Esta solução é composta por dois espaçadores, com duas cameras de endoscopia, colocados por detrás da lentes do capacete, que permitem a monitorização dos olhos do utilizador. Nesta tese falo sobre as razão que me levaram a esta solução, e falo também no desenvolvimento do software que a acompanha, especificamente o processo de calibração do seguidor ocular. Uma das aplicações desta plataforma de seguimento ocular é complementar a análise dos bio sinais na detecção de activações emocionais, e também para determinar onde é que o utilizador está a olhar na cena virtual e se ele(a) foge dos elementos

fóbicos.

Palavras-chave: Seguimento Ocular, Capacete Realidade Virtual, Realidade Virtual, Visão por Computator.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

ISR Instituto de Sistemas e Robótica

DEEC Departamento de Engenharia Electrotécnica e de Computadores

HMD head mounted display

VR Virtual Reality

SMI SensoMotoric Instruments

LED Light Emitting Diode

IR Infrared

USB Universal Serial Bus

PC Personal Computer

CMOS Complementary metal–oxide–semiconductor

PCB Printed Circuit Board

FPS Frames per second

GPIO General-purpose input/output

UDP User Datagram Protocol

HR Heart Rate

RR Respiratory Rate

EDA Electrodermal Activity

DK2 Development Kit 2

SMD Surface-mount Device

Chapter 1

Introduction

The first studies on eye-tracking date back to the late 19th century with the work of Louis Émile Javal, who with naked eye observations notice that readers eye's don't perform a fluid movement while reading text, but instead perform quick movements, called saccades, and small pauses, called fixations. The first eye-tracking device was created by Edmund Huey, in 1908, and used a sort of contact lens connected to a pointer that would move in response to the eye movements. Some time afterwords other less intrusive methods for eye-tracking were created that used cameras and beams of light to record, on film, the movements of the eye for later analysis (See fig. 1.1). These systems allowed for a better understanding of how the eyes move, particularly their dependence on the task being performed by the user (See fig. 1.2).

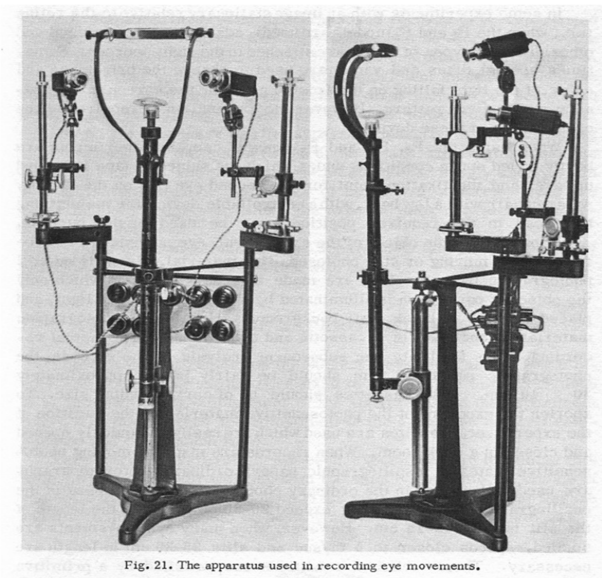

Figure 1.1: Eye tracker created by Alfred L. Yarbus to study eye movements [1]

On his book "Eye Tracking Techniques", Alfred L. Yarbus studied how different tasks influenced the gaze patterns. People were asked to look at a picture with the task of answering questions related to it. How old are the people? What are they wearing? [2]

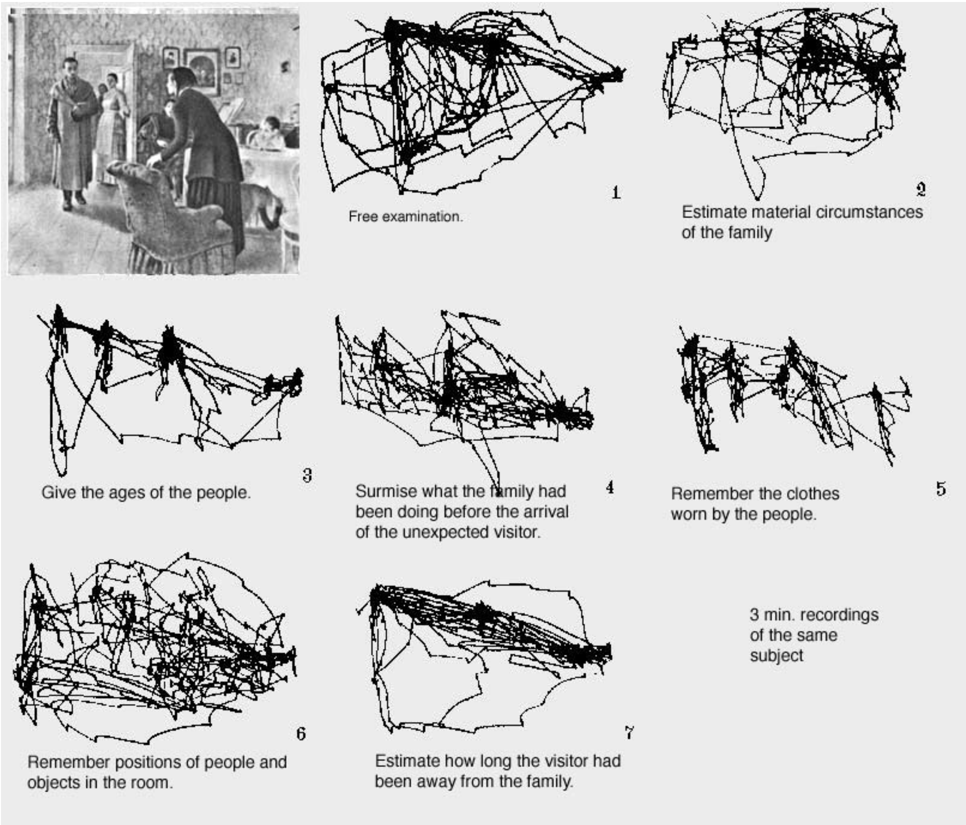

Figure 1.2: Gaze patterns according to the task being performed [3]

In the 1970 much work was done, specially by the military, on the improvement of precision and accuracy of eye-tracking systems, and on the their portability. With the appearance of small computer systems, eye-tracking data could be processed in real-time for the first time. In the 1980s with the boom in popularity of personal computers, eye tracking started being studied as a means of human-computer interaction. With researchers analysing how users navigated through menus and searched for commands [4].

Since a few years, eye tracking has been also explored for providing information about distraction and sleepiness. [5] [6]. Some companies, particularly those whose drivers need to operate heavy equipment, such as mining companies, use these systems as a way to prevent accidents. These systems allow site managers to monitor, in real time, the fatigue level of drivers and act before any accident happens [7]. Toyota and Lexus commercialised some car models which included eye-tracking systems [8]. These systems detect the attentiveness of the driver by analysing it's eyes, and will warn the driver with sounds and lights if a lack of attention to the road ahead is detected (See fig. 1.3), potentially applying the brakes if no action is done by the driver.

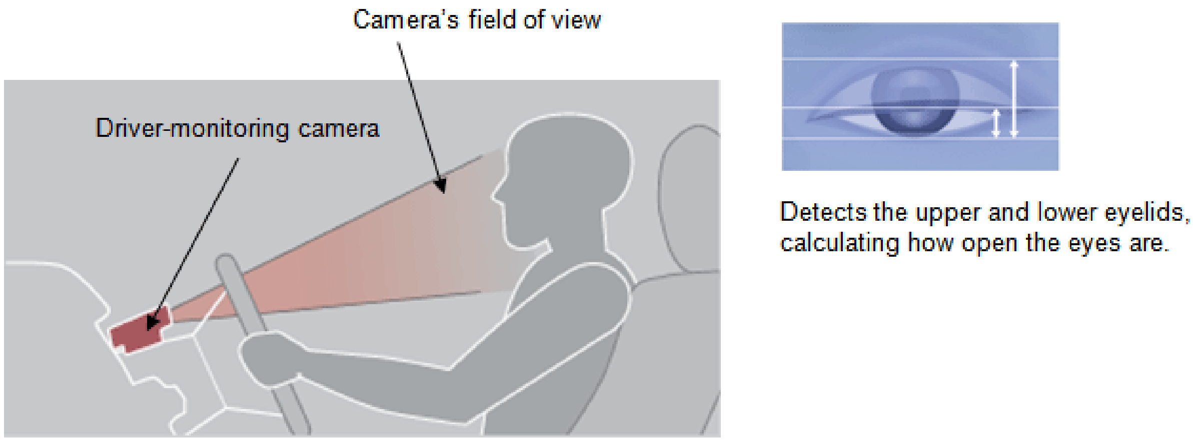

Figure 1.3: Outline of Pre-crash Safety System Eye-monitoring Feature by Toyota [9]

Marketing also uses eye-tracking to great extent [10] [11]. Marketeers analyse designs of advertisement posters and product labels, in terms of how successful they are in attracting costumers attention (See fig. 1.4). This process is also popular for website design, where web page layouts are designed in such a away as to maximize the amount of information that is given to the user in the shortest amount of time.

Figure 1.4: Eye tracking heat map used for marketing research [12]

The use of eye-tracking is also very popular as an assistive technology to help patients who are unable to communicate in standard ways (speaking through the mouth or with gestures) [13] [14]. For some handicapped people the eyes are the only means of interaction, relying on blinking patterns to communicate with others. A typical example for this type of communication is the care taker speaking or pointing at an alphabet board, waiting for the patient to blink in order to construct sentences, letter by letter. Such processes are laborious and require the constant presence of the caregiver. However with the use of eye-tracking patients can speed up these procedures, and use their eyes to gain some level of autonomy. These systems not only allow them to speak but can also be used for standard computer interaction, replacing the use of a mouse and keyboard.

1.1 Existing Eye-tracking solutions

The eye-tracking systems used nowadays fall into different categories: Electro-OculoGraphy (EOG), scleral contact lens/search coil and Photo/Video-OculoGraphy (POG/VOG) [15] [16] Electro-OculoGraphy detects eye movement by measuring the electric potential difference of electrodes surrounding the eyes (See fig. 1.5).

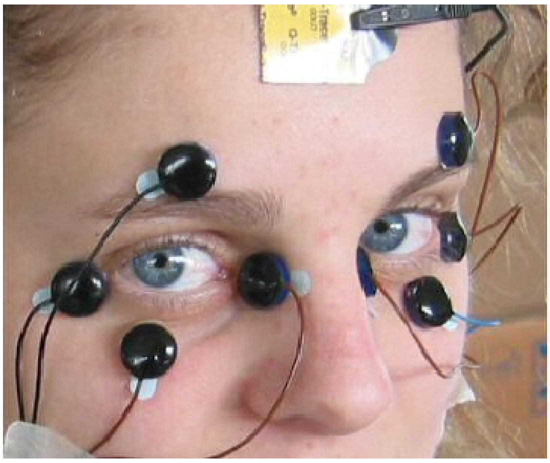

Figure 1.5: Electro-OculoGraphy system [17]

Scleral contact lens based systems use a contact lens connected with mechanical linkages to some recording/pointing device and search coil ones use a contact lens with a embedded coil to detect movement by analysing the deformation of a known magnetic field (See fig. 1.6).

Figure 1.6: Contact lens with embedded coil used for eye tracking [18]

POG/VOG systems use a wide variety of recording techniques, ranging from the position of the limbus (the iris-sclera boundary), the apparent shape of the pupil, the corneal reflection of a direct light source, etc, to determine eye-movement (See fig. 1.7).

Figure 1.7: VOG type Eye tracker with head rest [19]

Some systems don't allow head movement, such as the one above, and use a head rest for that purpose. While others allow for head movement by tracking the position of the users face in relation to the cameras with markers, either by using face tracking or by having a front facing camera mounted on the user that views the world from it's perspective (See fig. 1.8).

Figure 1.8: Eye tracking glasses with front facing camera [20]

These systems may support monocular or binocular eye-tracking, where naturally the ones that track both eyes may provide better information like depth estimation from vergence. Those who only track one eye normally do so on the dominant eye, which can be determined by a simple test: the user points it's finger at a distant object, and them by closing one eye at a time, the dominant eye is the one where the finger doesn't appear to move.
As for functionalities other than gaze tracking, systems that measure pupil size can return fixation information, systems that operate at rates of 30Hz can return information about blinking and those that operate at more than 60Hz can also track saccades. On these systems the error is typically measured by the angular error, which for high end devices is normally below 1^o. For a system with a angular error of 1^o this means that at a distance of 60 cm from the screen the users fixation point uncertainty may be approximated by as lying on a circumference with a radius of 1 cm.
The biggest players on the field are SensoMotoric Instruments (SMI) [21] and Tobii [22]. Both providing remote and non remote eye-tracking solutions. Particularly their eye-tracking glasses

have been used extensively by athletes, marketeers and researchers for several projects. These include analysing athletes gaze during sporting events, customers' behaviour to analyse their response to changes in product distributions in supermarkets, in web-page design to improve user interaction, or even in the development of interaction support for disabled people.

1.2 Eye tracking solutions for HMD

With the advent of the Oculus Rift, in 2012, head mounted display (HMD) gained a boost in popularity given their bigger field of view and lower cost, when compared to previous headsets. This refuelled the attraction of researchers to Virtual Reality (VR). Gaze tracking has also attracted researchers that develop or use VR for different purposes. Nevertheless, the inclusion of the eye-tracking devices in traditional HMDs is a hard task. Currently the more popular HMDs are the Oculus Rift [23] (See fig. 1.9), the HTC Vive [24] and the Samsung VR [25]. These three headsets have upgrade packages provided by SMI, which in the case of the Oculus Rift development kit 2, the oldest of the three, costs upwards of 14000€.

Figure 1.9: Oculus Rift upgraded with the SMI eye tracking package [26]

There are also announced headsets to include eye-tracking capabilities as is the case of the FOVE [27], funded on Kickstarter (popular crowdfunding website), and the StarVR [28] which

Table 1.1: Comparison of eye tracking solutions for HMD

HMD	Accuracy	Frequency of capture	Cost
SMI upgrade for Oculus Rift dev. 2	0.5-1º	60Hz	>14000€
SMI upgrade for HTC Vive	0.2º	250Hz	Very High
SMI upgrade for Samsung VR	0.5º	60Hz	Very High
FOVE VR headset	>1º	120Hz	$599
StarVR powered by Tobii	Unknown	Unknown	Unknown

is aimed at the high end market, particularly for IMAX experiences. A comparison of the claimed capabilities of each system is presented below.

1.3 Motivation and Objectives

Psychology is one of those areas with a particular interest in eye tracking for different purposes, and among them the exposure based therapies. In exposure based therapies the patient is exposed to elements intended to trigger an emotional response. For example, if a patient has a phobia for spiders, the therapist is going to expose the patient to scenarios with spiders, and through a desensitisation process expect that the fear decreases over time. In order for the therapy to work, the emotional and behavioural state of the patient must be analysed. But, for the sake of having a more reliable evaluation of the patient condition, the therapist needs to know how he/she reacts when looking at the triggering/traumatic element, and this can be done with the help of eye-trackers. However with the use of HMD the face of the user is partially obscured, and as such is difficult to access his facial expression, and more importantly to determine where the user is looking at. When the user is looking at a virtual scenario through the helmet the therapist doesn't know exactly where the he/she is looking at. Has the patient closed it's eyes when faced with a phobia inducing elements? Has the patient looked way from those elements? In order to answer this questions an eye-tracker can be used inside the HMD to provide real-time information to the therapist.

The use of an eye-tracker inside of a HMD could also serve to minimize one of the biggest problems most users of HMD have, motion sickness, that's mainly caused by an asynchronism between the user's head movement and the scene movement (low frame rate, high latency).

This problem could be addressed by using a technique called foveated rendering. Which consists of only rendering the scene with full detail on the area of gaze and the rest with lower detail, this minimizes the required processing power necessary to render the scene, making it more responsive.
As above stated, on the market there are some commercial solutions that provide eye-tracking capabilities for popular HMD, but do so at a prohibitively high cost for most researchers, and require extensive modification of the units. As such the objective of this thesis is to develop a low cost, open source, solution to track the users gaze inside the popular Oculus Rift Development Kit 2 (DK2), to be used on the study of emotional activations inside VR environments.

1.4 Thesis Structure

On the second chapter the hardware development phase is discussed. On this chapter the several prototypes developed are shown and their strengths and flaws discussed. The third chapter deals with the software development phase, the eye-tracking algorithm is presented and some applications of this work are shown. Finally on the fourth chapter the conclusions are presented and future work is discussed.

Chapter 2

Hardware

Adding a Eye tracker to the Oculus Rift Development Kit 2 (DK2) requires the transmission of video from the head mounted display (HMD) to the main processing computer, this can be done by adding cables, that run along side the ones already present (The Oculus is connected to the computer with two cables, one for the HDMI port and one for the Universal Serial Bus (USB) interface, which it uses for power and to transmit inertial data), or transmit the video feed wirelessly.

Another aspect to consider, as in most computer vision applications, is illumination. A system with good illumination greatly simplifies the process of image segmentation, by reducing noise caused by the lack of light entering the camera sensor and by increasing the contrast between the areas that are of interest and those that are not. When the user is using HMDs the illumination of the eyes is weak, so as not to cause discomfort, and is highly dependant on the scene that's being displayed on screen, as such, a constant, more powerful light source is necessary. In eye tracking applications the use of Infrared (IR) illumination is widely used [29], because of the dark pupil effect (see image below), that increases the contrast between the pupil and the iris, and because it does not interfere with the undergoing visualisation process. So from a early stage in development it was decided to use IR illumination on the prototypes.

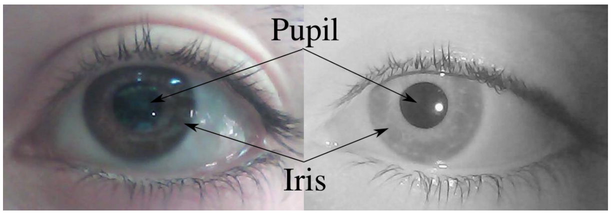

Figure 2.1: Left: Eye under visible light illumination. Right: Eye under IR light

Several solutions were implemented, which are discussed on this chapter, detailing their advantages and disadvantages.

2.1 IR light safety considerations

By using IR illumination some safety considerations should be taken into account. The IR light emitted by the Light Emitting Diode (LED)s, used for typical eye-tracking applications, emit on the IR-A band, which extends from 700 nm to 1400nm. On this band the dangers for the optical system include: Thermal injury to the retina of the eye (400 nm to 1400 nm) and Near-infrared thermal hazards to the lens (approximately 800 nm to 3000 nm). [30] When skin is exposed to this type of radiation it gets warm providing a sort of warning mechanism against damage, however the eyes don't have that sort of reaction. As such care should be taken when exposing the eyes to IR radiation, as prolonged exposure can lead to cataracts and burns. The American Conference of Governmental Industrial Hygienists (ACGIH) and the International Commission on Non-Ionizing Radiation Protection (ICNIRP) recommend a maximal daily corneal exposure (for periods of more than 1000s) of $10mW/cm^2$ total irradiance for wavelengths 770-3,000 nm [30].

2.2 Solution A

The first proposed solution consisted on creating a wireless eye tracking system. This system should transmit one or more camera feeds to the host computer for processing, and if possible to process the feeds remotely, thus freeing the computer of some processing time.

2.2.1 Implementation

To implement this solution a processor unit is required, it should have a low cost, small size and be lightweight. The Oculus itself weights about 440 grams, so adding much more weight to the unit is not advisable, so as not to cause discomfort on the users neck after prolonged use. The Raspberry Pi (model B) portable computer, had all this characteristics, so it was chosen as the processing unit. To test it's wireless video transmission capabilities the following setup was used:

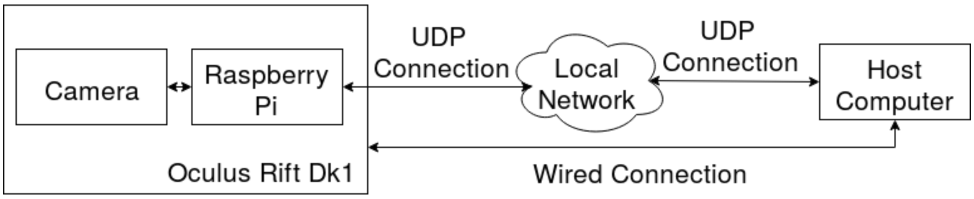

Figure 2.2: Video transmission diagram (to replace)

The Raspberry Pi uses a WiFi dongle to transmit the video feed, over a User Datagram Protocol (UDP) connection, to the host computer. As for the camera, the Raspberry Pi camera module was used. This camera allows the recording of video up to 90 Frames per second (FPS), with VGA quality (640x480 pixels). In order to illuminate the eye a single IR LED was used.

Figure 2.3: Eye tracking hardware setup using the Raspberry Pi

This initial prototype was designed for Oculus Rift DK1 that differs in several aspects from DK2, in particular in terms of available space.

2.2.2 Results

Advantages:

- High frame rate of the camera allows for the detection of saccades.
- Being wireless no extra cables are necessary.

Disadvantages:

- The system presented a latency of about 300 ms, which for eye tracking application is far from great. During saccades the eyes can move at speeds of up to $900°/s$ per second , so by the time the image is transmitted over to the main computer and processed, the point of fixation can be completely different. Rendering the data useless.
- Given the size of the camera Printed Circuit Board (PCB) (25x24 mm) the only place it could be placed, inside the HMD, was on the outermost part of the lenses (see image above). And on this position only a small portion of the eyes movement can be tracked. On the particular case of tracking the left eye of the user, only when he/she was looking at the left part of the HMD screen, could the camera get a clear image of the pupil.
- In order to get binocular gaze data another camera is needed, and transmitting video from two cameras at the same time would only increase the latency of an already delayed system.
- The use of a single IR LED proved insufficient for proper illumination (see image below).

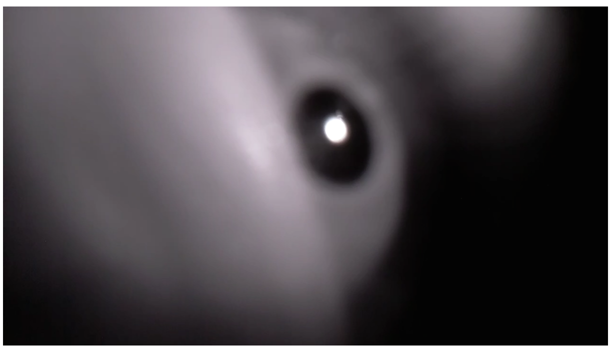

Figure 2.4: Image acquired using this setup

The main focus of this implementation was determining if the wireless transmission of video was good enough for gaze tracking, but since the results proved fairly negative, on the latency side of things, no major experimentation was made on using the Raspberry Pi to pre process the frames, and on powering the unit "wirelessly". The Raspberry Pi was powered by an external USB charger. If the results were positive the plan was to either use a power bank or to get power from the HMD itself.

2.3 Solution B

This solution will present a wired system where the cameras will be connected to the host computer via a USB interface and the illumination will be provided by an array of IR LEDs surrounding the lenses of the HMD. Note that this solution was developed for the Oculus Rift DK2 and not DK1 as the previous one.

2.3.1 Implementation

Given the space constraints, inside the Oculus HMD, a small camera was necessary, so the choice felt on endoscopy type cameras. This type of camera is typically used for visual inspection of places where a conventional camera can't fit, be it inside the human body or inside the engine of a car, to give some examples. The really small ones are those used for medical applications, where sizes can go as low as 1mm, however these can normally only be acquired in bulk and come with a high cost. On the other hand, non medical ones can be acquired for less than 10€, however their footprint is bigger.

After some research the following pair of cameras was chosen: two XINFLY XR-IC2MHD. These cameras have a size of 7,8x39 mm, a maximum image resolution of 640x480 pixels (VGA resolution), a refresh rate of at most 30 Hz, are accessed using a USB interface and were priced at about 8€. They also came with a removable lens, which will leave room for future modification. These cameras, like most commercial ones, came with tiny filter that blocks IR light. This had to be removed, but given their small size, about 2 mm in diameter, special care had to be taken.

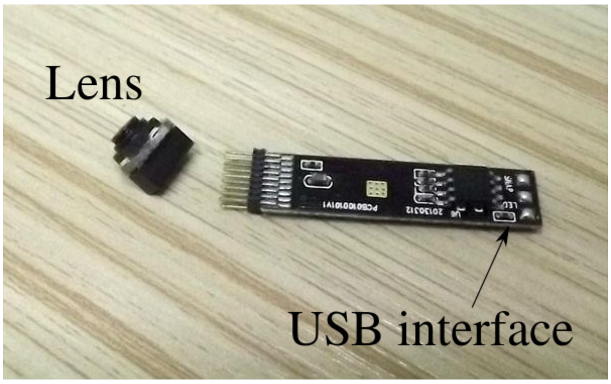

Figure 2.5: Endoscopy camera used

With new cameras available some tests had to be made on their positioning, so a temporary illumination plate was designed, without supports for the cameras. This plate had nine repurposed IR LEDs (taken from household appliances) mounted on it, placed in a circular pattern around the HMD lens. And was powered by the USB port of the HMD, it had a total power consumption of about 60 mA.

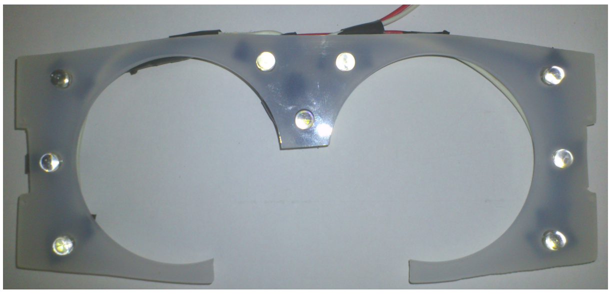

Figure 2.6: Acrylic plate with IR LEDs

Several positions were tried for the cameras, some on the inside of the HMD and some on the outside, and the one that provided the best results was the following.

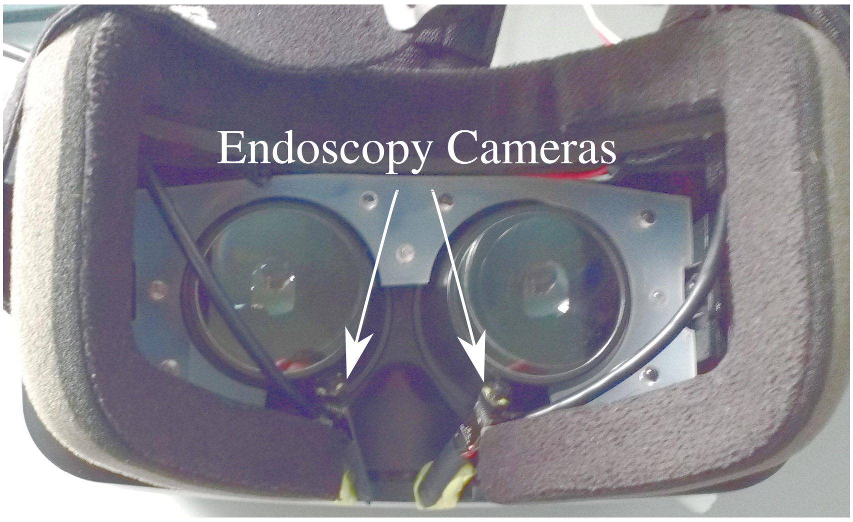

Figure 2.7: HMD with illumination plate and endoscopy cameras

With a camera position chosen, a new plate was designed with some brackets to hold the cameras in a fixed position.

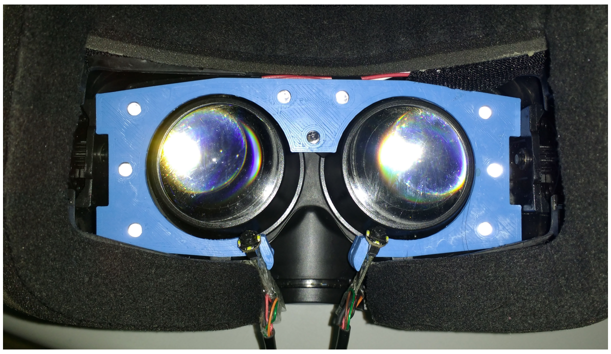

Figure 2.8: New plate with both illumination and camera brackets

In order not to add extra cables to the HMD, a USB hub was used between the desktop computer and the HMD. On this hub the webcams, the illumination and the HMD USB cable where connected.

2.3.2 Results

Advantages:

- Low cost. The two cameras came at under 20€, the LEDs cost less than 0.50 cents each, and the plate can easily be made with a common 3D printer. As a result, the total cost sums to about 30€.

- The cameras are able of acquiring the full range of motion of the user's eyes, while wearing the HMD.
- The IR LED array provides good clear illumination to the eyes, thus we are able of getting images with no shadows, simplifying image segmentation.

Disadvantages:

- With a frame rate of only 30 Hz, saccades can't be tracked, and fast eye movements will result in blurred images, where pupil segmentation is not possible.
- The utilization of the Oculus USB cable proved impractical for binocular video transmission, because of the high bandwidth demand of the cameras. They are incapable of transmitting compressed video and each one will request the full bandwidth usage of the USB controller, even when using a smaller resolution than standard VGA. Testing the same system configuration with other webcams present around the lab (two "Logitech QuickCam® Orbit AF") no such problems existed, and dual video transmission was possible on a single cable. So this seems to be a problem exclusive to these endoscopy cameras. One solution for this is to connect each of the cameras to a different USB controller, to do this at least one of them has to bypass the USB hub and be connected directly to the desktop computer.
- These cameras tend to heat up during operation, and being so close to the users nose and eyes, they tend to dry up the eyes and start to "burn" the nose even after a short usage period of less than 5 minutes.
- The distance between each of the camera brackets is only 3 cm, so users with noses wider than that can not use the HMD properly and can have difficulty breathing.
- Video stability was also a problem. If the HMD moved abruptly the cameras would shift position and required re-adjustment. Also when the HMD was removed, or placed, on the user's head some camera shifting tended to occur.
- The user's eyes reflect what the HMD is displaying causing variations in illumination and adding artefacts to the pupil segmentation procedure. Notice on the image below how the screen reflections, particularly on the left eye affect the colour of the pupil.

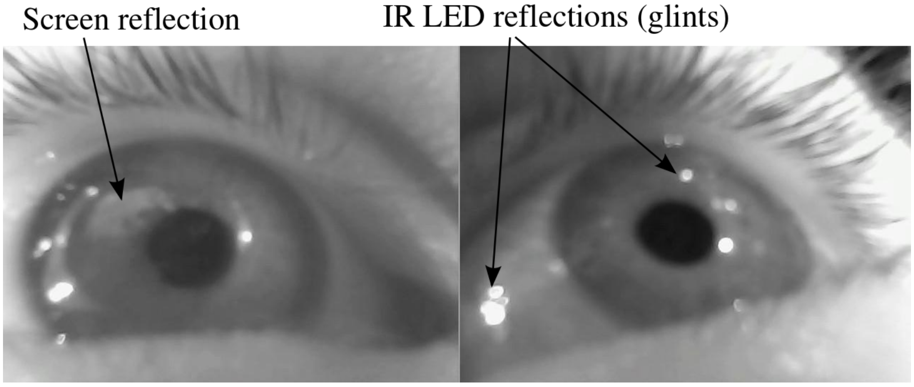

Figure 2.9: Grayscale eye footage acquired using the endoscopy cameras

2.4 Solution C

The Oculus rift development kit 2 comes with two sets of lens cups, named A and B. These two sets of lenses have the same optics but the A set is taller than the B one by about 2mm (A lens: 21mm, B lens: 19mm). This height difference influences the focal point of the HMD and is intended to compensate, in some way, for the users vision acuity. The B lenses aimed at people with some level of short-sightedness, or myopia, allowing them to use the HMD without glasses. This solution takes advantage of this height difference between both lens, to use a spacer to elevate the B lens to height of the A ones, leaving a opening were a camera can be inserted. This way cameras can be placed inside the HMD without permanently modifying the unit.

2.4.1 Implementation

See appendix A for details on the spacer construction.

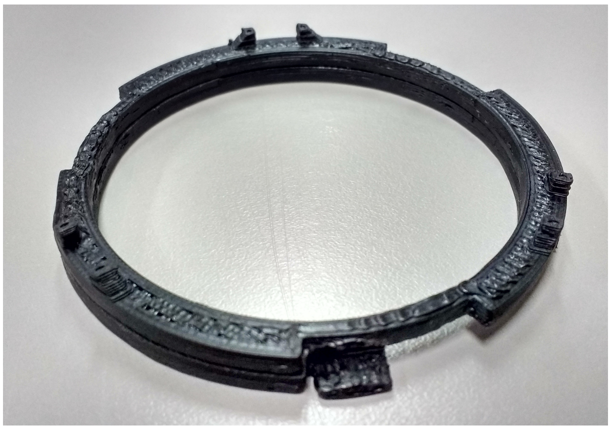

Figure 2.10: Spacer used to place the camera under the lens cup

This first prototype of the spacer increased the height of the B lens by about 4 mm, an increase of 2 mm over the standard A lens, causing some blurring of the borders of the display, but is not detrimental to the immersion. On the image below a detail of the underside of the spacer shows the opening for the camera (left).

Figure 2.11: Detail of the spacer showing the camera positioning

In order to re-use the endoscopy cameras, some modifications had to be made to them. The camera connector had to be bent $90°$, as seen in fig.2.12. This way the camera faces the user's eye, while it's PCB is inserted through the spacer opening.

Another modification to the cameras was the inclusion of a visible light filter, placed in front of the camera CMOS sensor. This filter will block all visible light emitted by HMD screen, thus eliminating screen reflections. The filter used was the LEE filter 87, a polyester filter, that blocks all visible light transmission below 730 nm of wavelength.

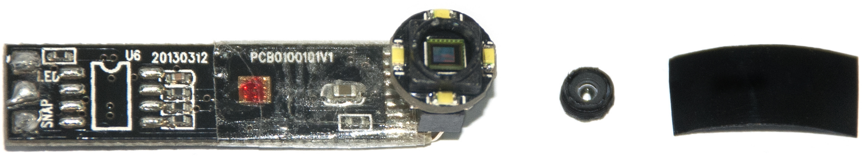

Figure 2.12: Camera with lens connector bent 90 degrees, showing the CMOS sensor and the visible light filter

On the figure 2.13 I present two successive frames of a video recorded with the two cameras mounted on the HMD. (Note that these particular cameras show the IR spectrum with a red tint) The camera on the left had the visible light blocking filter and the one on the right didn't. As can be seen the use of a visible light blocking filter greatly improved the image quality by eliminating screen reflections. Thus simplifying the pupil segmentation procedure, as will be described on the next chapter.

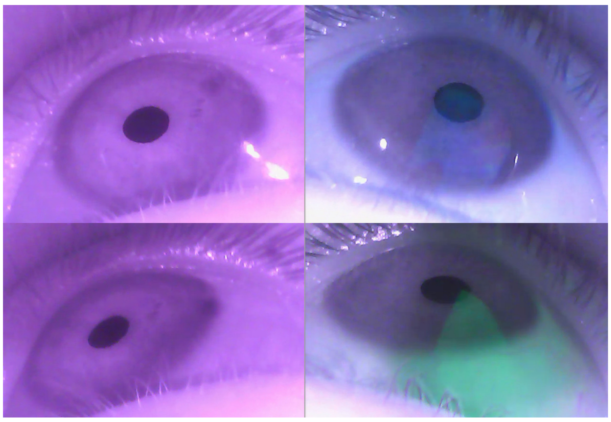

Figure 2.13: Comparison between the use, and not, of a visible light filter

Since this solution uses individual spacer adaptors for each of the lenses, a decision was made of also developing individual illumination rings for each of the eyes. This decision was made to add the possibility of only building a single eye gaze tracking system, and thus only require the manufacture of a single lens adaptor and illumination ring, if need be. In the previous prototypes the LEDs used where all through hole ones, salvaged from old appliances. They where bulky and their technical specifications where mostly unknown. In order to provided safe illumination for the user, and taking into account the inclusion of a visible light filter, new IR LEDs were selected, taking into account the safety considerations presented previously. The LEDs wavelength had to be greater than 730nm (the cut off wavelength of the visible light filter). The combined total irradiance of them should be lower than $10mW/cm^2$. And finally since they are mounted on the borders of the eyes, their viewing angle should be big enough so as to properly illuminate them.

The Vishay VSMY1850ITX01 LEDs met these requirements and where chosen for the illumination rings. They have a wavelength of $850nm$, the sum total irradiance of all 16 LEDs $E_{total} \approx 8mW/cm^2 < 10mW/cm^2$ (worst case scenario), and each costed about 0,63€. For details on the construction of the rings read appendix B. The resulting IR illumination rings are shown on the next image.

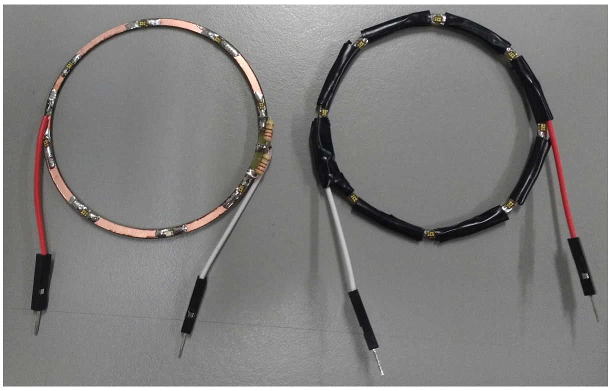

Figure 2.14: Illumination rings. Left: bare copper, right: insulated with tape

Next images of the fully assembled unit are presented.

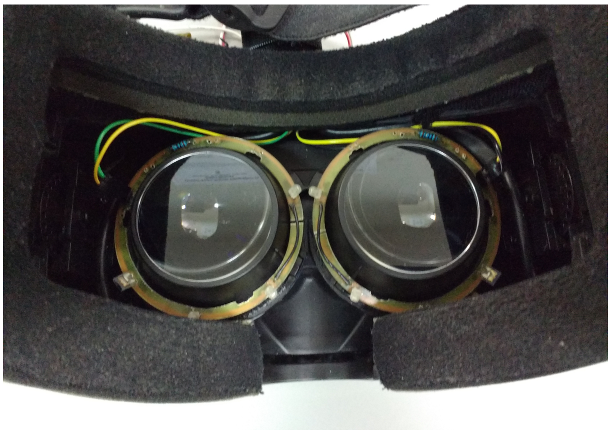

Figure 2.15: HMD with both cameras mounted and with the IR ring

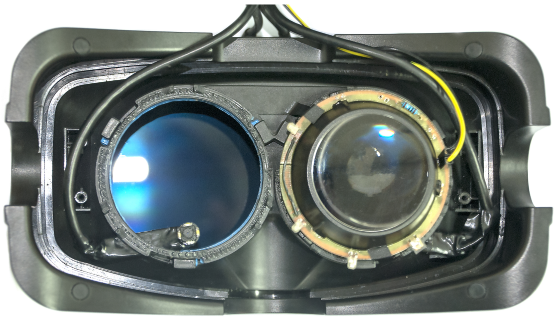

Figure 2.16: Detail of the HMD with one lens removed to show the camera

2.4.2 Results

For this solution the same cameras as before were used, and the advantages and disadvantages of this solution with respect to the previous one are listed below.

Advantages:

- After installation no further readjustment is necessary. Making this an install and forget solution.
- Good video stability, even during fast head movements.
- No heating issues. Since the cameras are placed away for user's skin, and covered by the HMD lens, excessive heating of the cameras no longer prevents long term operation.

Disadvantages:

- The increase in height of the lens, caused by the spacer, increases the blurriness of the image, particularly on the borders of the visual field of view. This effect is not very pronounced, but for some users can prevent a comfortable experience. One way to solve this is by making the holes for the cameras directly on the lens unit, but this entails some risks, since the plastic cover of the lenses can break and replacements for them are unavailable for purchase.
- The user is capable of viewing the cameras, although, this only happens when he/she is looking directly down.

Chapter 3

Software

To successfully track the gaze of an user we need an algorithm that is able, on a first instance, to segment the eye image leaving us only with the pupil, and afterwards to determine it's center and use this information, in conjunction with a previous calibration procedure, to transform the coordinates of the pupil center into viewed scene coordinates. Hereafter follows the developed algorithm.

3.1 Algorithm

The eye-tracking algorithm can be summarized by the following steps:

Figure 3.1: Algorithm steps diagram

Before starting the segmentation of the pupil some preprocessing on the raw image of the eye is performed in the form of a Gaussian blur filter, as a way to attenuate image noise, and only then a binary threshold is applied to the resulting image. Choosing a proper threshold value is key to obtaining a good segmentation of the pupil, too low/high value can lead to poor accuracy. Initially I've used the same approach as Miguel Brito [32]. This approach consisted on converting the image to grayscale, equalizing the histogram and then use the histogram's median as the threshold value. The problem with this approach lies with the fact that it requires the pupil to be the only "dark" portion of the image to successfully segment the pupil. If the eyelashes and/or eyebrows are thick they tend to create artifacts on the segmented image, as seen below. This artifacts can led to tracking errors and as such need to be removed. One way to do this is to use a mathematical morphological operation[1], such as: opening, erosion, etc., on the segmented image. However this is somewhat computationally intense and can increase (or decrease) the size of the pupil, thus leading to more tracking errors.

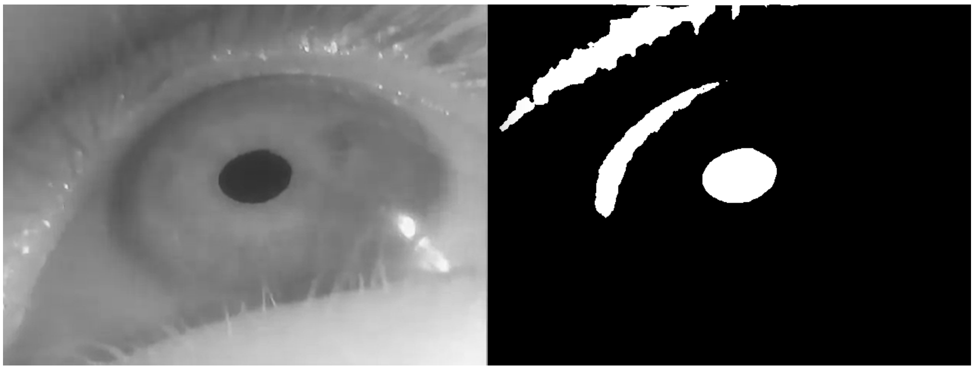

Figure 3.2: Segmentation using Miguel Brito's method

As such a new approach was necessary. So I've decided to analyse the colour histogram of the raw image. By using one of Matlab's computer vision applications, as seen on the image below, we can see that there are small spikes on the histograms that correspond to the colours associated with the pupil. Using a threshold value based upon these spikes allows a good segmentation of the pupil, as seen below, without the need for further post-processing.

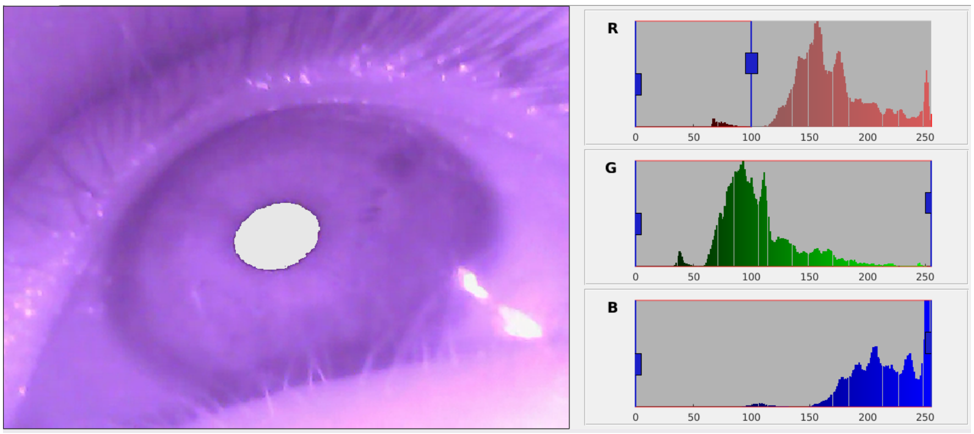

Figure 3.3: Pupil segmentation using Matlab's "Colour Thresholder" application (threshold using red colour channel)

[1]A mathematical morphological operation is the process of convoluting a binary image with a binary structuring element and then selecting a binary output value depending on the thresholded result of the convolution. [33]

Converting the image to grayscale and then looking at it's histogram also showed a corresponding spike, and it could also be used to successfully segment the pupil. However for the particular cameras we're using on the HMD, the red colour channel provided the best results, what is quite expectable due to the proximity to the IR band. Depending on the hardware being used for eye-tracking, one or the other threshold method can be selected.

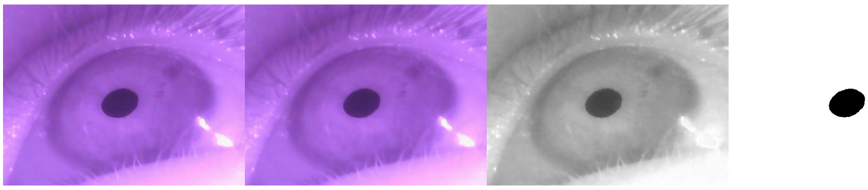

Figure 3.4: Pre processing of the raw eye image

From left to right: Raw image, image after applying the Gaussian Filter, red colour channel and segmented pupil.

After segmentation comes the process of determining the center of the pupil. The first step is to find the contours of the segmented image, for this the OpenCV function "findContours" was used, which is based on the algorithm described in [34].

Depending on the location of the pupil, in relation to the camera, it can have a circular form (looking at the camera axis) or it can have the shape of an ellipse. On the HMD the cameras are placed in a off axis position in relation to the user's viewing axis, as such they will show the pupil with an ellipsoidal shape. The center of this ellipse is obtained using the OpenCV ellipse fitting function "fitEllipse" whose algorithm is based on [35].

Figure 3.5: From left to right: Segmented image, contours and finally ellipse fitting of the pupil contours.

This gives us good tracking of the pupil and given the absence of major artifacts after the segmentation stage blink detection is fairly accurate.

The major problem of this setup comes from the low framerate of the cameras used, the 30 frame per second these cameras provide are not enough to get clear eye images during eye saccades, and as such tracking can be momentarily lost and be confused with a blink. To prevent this false blink detection a counter is used, and a blink is only validated if no ellipses are found during "x" successive frames.

Figure 3.6: From left to right: Frame captured during a saccade and algorithm result

3.2 Calibration

The objective of the calibration step is to establish a relationship between the detected coordinates of the eye pupil center and corresponding fixated locations on the Oculus Rift display. By creating a image with several targets on known positions and recording the positions of the pupil while looking at those targets it's possible to to estimate the necessary mapping function. However those coordinates are not the pixels on the display but those of the displayed image (texture), since this one is going to suffer the mapping by a homographic projection followed by radial distortion. If we want to establish a relation with a particular scene region, corresponding to our focus of interest, we must convert these coordinates into scene coordinates, eventually through the creation of a cone that can be used to detect the intersection with other scene objects. This conical object represents all the scene points projected on the users gaze zone. This zone can be represented by a circle or ellipse which has a radius correspondent to the eye-tracker's uncertainty. At this stage we are only going to do the calibration calculus in relation to the texture and not in relation to the display or scene.

3.2.1 Calibration targets

When selecting the number of calibration targets two considerations should be made. The time it takes to complete the calibration and the accuracy that can be achieved with them. The software allows for two methods of acquiring the calibration targets coordinates: a manual and an automatic one. For the manual case, circular targets, see image below for details, are shown one at a time using a black colour over a white screen. The user looks at them and presses a key, this records the pupil coordinates for that target and presents the next one. This is done until all 9 targets are registered.

For the automatic one the procedure is similar but instead of waiting for the user to press a key to change the target, each one is shown for approximately 1.5 seconds before moving on to the next, and the pupil coordinates are recorded just before the target change is made. In order to quickly attract the user gaze to the targets, and to provide good fixation on them, they exhibit some dynamic behaviour, by decreasing their sizes along time until vanishing. [29]

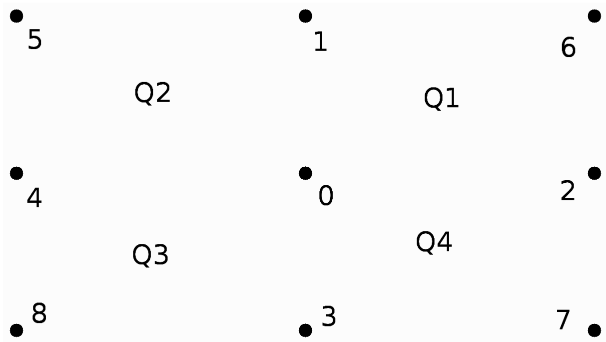

Figure 3.7: Calibration targets

On the image above the calibration targets are presented, with the number representing their appearance order. For reasons to be explained below, the screen was divided in 4 quadrants which are represented by the labels Q1..Q4 on the above image.

3.2.2 Calibration method

With the coordinates pairs acquired now comes the time to obtain the mapping function f, which is used like this:

$$\mathbf{s} = f(\mathbf{p}) \tag{3.1}$$

Where $s = (s_x, s_y)$ are the screen texture coordinates of the point the user is looking at and $p = (p_x, p_y)$ are the corresponding image coordinates of the pupil center. To note that left and right eyes will have different mapping functions, mainly due to the dominance of one of the eyes.

Initially a linear approximation was tried. The homography between both point pairs was calculated using two different approaches. The first one calculated a single homography for the whole screen (using 5 and 9 points) and the second calculated a separate homography for each of the four quadrants, using the corresponding calibration points for each one (for example, for the first quadrant, points 0, 1, 6 and 7 are used).

To test these methods for accuracy, besides the previous mentioned 9 calibration points, an additional 16 points where collected in between the first ones. All these points made a 5x5 grid as shown below. This grid has a width, and height, of 320 pixels.

Figure 3.8: Point locations used for the calibration testing

It is important to note that the grid showed above is undistorted. When viewing it through the Oculus lenses it presents some form of radial distortion, however since at this point we are only mapping the pupil to texture coordinates, this is not important. Using the OpenCV function "findHomography" we calculate the homographies using the original 9 points, and then all 25 points are transformed using the previous mentioned equation. And the Euclidean distance between the real screen coordinates and the estimated ones was calculated, as a way to obtain the error in these approximations.

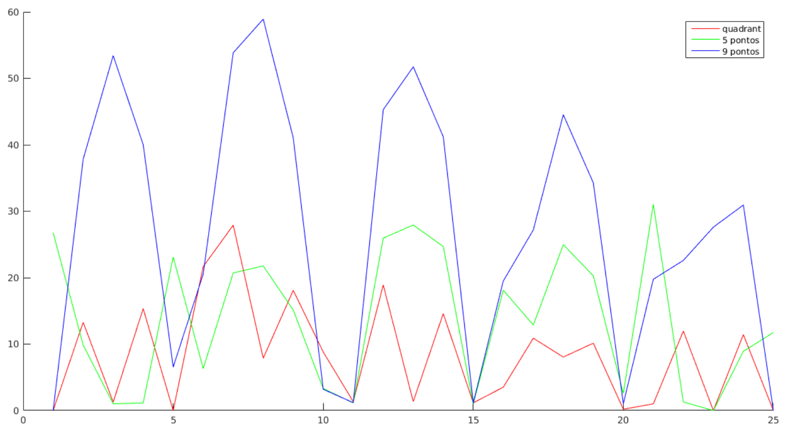

Figure 3.9: Euclidean distance error between real and estimated coordinates using homographies

Table 3.1: Euclidian error using homographies

Homography Method	Mean error	Abs Max error
5 points	13.6670	31.0429
9 points	27.3331	58.9140
quadrant	8.3500	27.8960

Looking at this data it is interesting to note that using 9 points to obtain the homography gives worse results for the tested point locations than by using only 5. This happens because the distortion caused by the Oculus lenses and the geometry of the eye is more evident on the borders of the field of view. So when using only points 0 to 4 we are obtaining a homography that better represents the inner most portion of the screen. Versus when using points 0 to 9 that

tend skew this results. However in both cases the error is still to great to precisely determine where in the screen the user is looking at. By using the quadrant based approach we get slightly better results, but with a maximum error of about 28 pixels gaze detection is still not possible. However these results are sufficient, for example, to subdivide the screen in a grid, of at most 5 by 5 spaces, and to say at which one the user is looking at.
With a linear approach to coordinate mapping giving poor results we started experimenting with non linear approximations. For this we used Matlab's "fitgeotrans" function to obtain a polynomial approximation for our coordinate transformation needs. Using the same method to calculate the error, the following results where obtained.

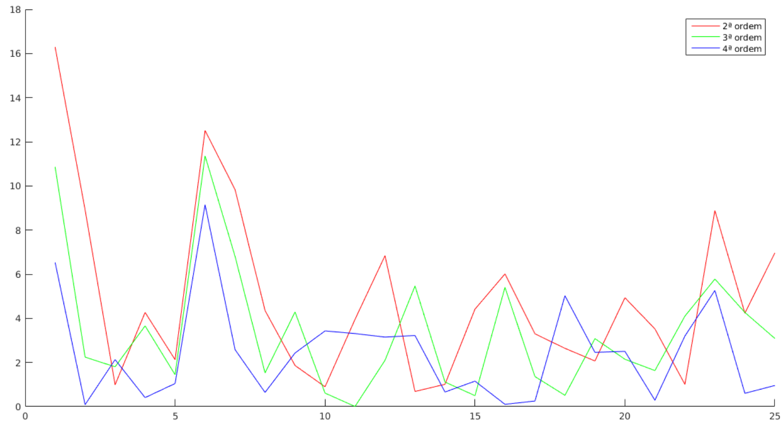

Figure 3.10: Euclidean distance error between real and estimated coordinates using polynomials

Table 3.2: Euclidian error using polynomials

Polynomial order	Mean error	Max error
Second order	4.9010	16.2896
Third order	3.4066	11.3599
Fourth order	2.4244	9.1497

With a second order polynomial our estimation errors where halved, with the greater 10 plus pixel errors present on the outer most points. Third and fourth approximations where also tried given, and as expected gave better results.

3.3 Technical Discussion and Implemented applications

The eye-tracking algorithms and methods where coded onto a C++ class, that was added to the OpenAR (Open Augmented Reality) engine that's currently being developed by professor Paulo Menezes group at the Instituto de Sistemas e Robótica (ISR) lab of the Departamento de Engenharia Electrotécnica e de Computadores (DEEC) of the University of Coimbra. This class was developed under a Unix environment and uses the OpenCV library for image manipulation operations. Some applications where developed to showcase some of the possible uses of Eye-tracking in Virtual Reality (VR) environments, and are showcased on this chapter.

3.3.1 Avatar Eyes

Figure 3.11: Avatar Eyes Application

This application shows how the eye-tracker can be used to control the eyes of a virtual avatar. The user runs the calibration procedure and afterwards the avatar will mimic the apparent gaze of the user's eyes. To do this the application uses the position of the eyes in relation to the calibration points to apply a rotation to a sphere object that represents the eye of the user. If a blink is detected the texture of the eye sphere is replaced with one that matches the users skin tone. On figure 3.12, several examples of these transformations are shown, using an avatar created using the Structure Sensor.

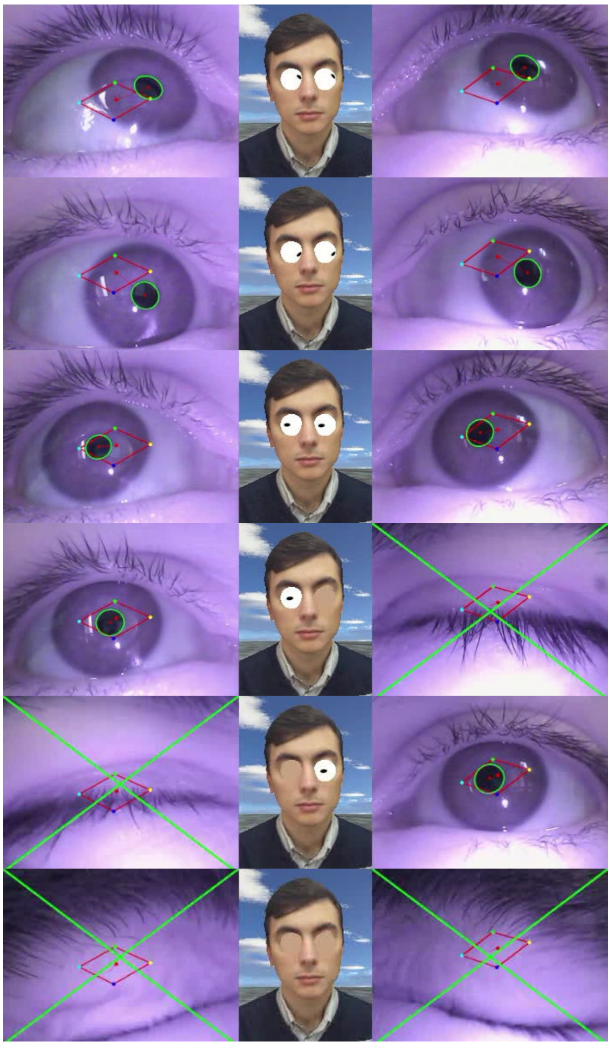

Figure 3.12: Avatar Eyes example transformations (eye size exaggerated for demonstration purposes)

This type of application can be useful, for example, for increasing the immersion in scenarios where several users interact inside a virtual reality environment. These scenarios far extend the traditional game use, and could include therapeutic uses. One such example is a scenario where users with social anxiety could interact with each other, being rewarded by looking at each other in the eyes. Typically the therapist asks patients to perform such exercises on the 'real world', however they lack a away to know for sure if the patient has actually done them, and must rely on the word of the patient. By performing these exercises in a VR environment the therapist is able to have a greater control of what the patient is exposed to, and to analyse it's emotional responses in real time.

3.3.2 Scene Explorer

Figure 3.13: Scene Explorer Application

This application allows the user to explore a scene using the HMD, while overlapping it's apparent gaze location, over the scene, to the intructor on a secondary screen. This application is controlled by the instructor who can adjust the threshold parameters on the fly and recalibrate the application, if necessary. As soon as the calibration is performed the instructor is able to visualize the apparent gaze of the user, using a circular or cross shaped object (figure 3.14 first and second images), whose radius represents the error calculated for the calibration. In addition to these, a "paint" like visualization is included that reveals the areas of the scene that the user has looked at (third image on figure 3.14). In addiction to this the application also saves the gaze data, including a timestamp, that the instructor can use for later analysis.

Figure 3.14: Gaze visualization options available

In order to have a one to one correspondence between the calibration space and the displayed scenes, the screen coordinates in both situations should the same. To obtain this a same size texture image is created for both situations, and subsequently sent to the OpenAR engine, that applies the radial distortions needed for proper viewing inside the HMD. In the case of the secondary screen the raw texture is displayed with the apparent gaze target overlapped on top of it. This type of application is useful to access the response of users to certain stimuli, ranging from phobia therapies to learning scenarios where, for example, the instructor needs to know if the user looked at certain parts of the scene in a certain order.

3.3.3 Cerebral Palsy Keyboard

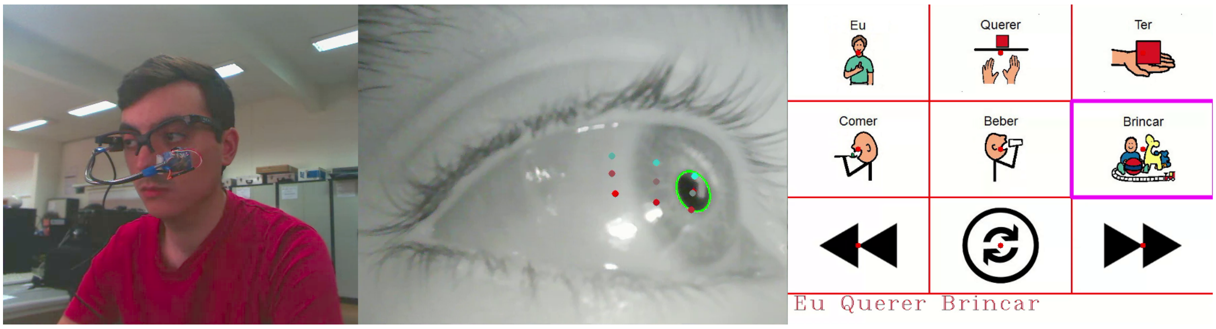

Figure 3.15: From left to right: User wearing the eye-tracking glasses, view of the users eye and virtual keyboard seen by the user

Using some of the visual assets used for the Cerebral Palsy interaction system developed by Galante and Menezes [36], I've developed an application that allows people with some form of physical and/or speech impediment, such as cerebral palsy, to construct sentences, and synthesise them, using their eyes as sort of pointing device. Whereas Galante and Menezes used a (expensive) commercial Eye-tracking system this application was demonstrated using only an off the shelf webcam, whose IR filter was removed, and some form of IR illumination to provide contrast for the pupil (if the working area is well lit by the sun no additional illumination is required, else a single lower power IR LED suffices). On figure 3.15 the webcam and the IR LED are mounted a glasses frame.

The graphical interface consists of 9 buttons displayed in a grid, as shown on figure 3.15, and the bottom part of the screen is used to display the sentence being constructed. The top 6 buttons represent common words the user can use to constructs sentences, the bottom left and right ones are for navigating between different word screens and the bottom middle one is used for deleting the previous selected word from the sentence being constructed. When the user looks at a certain button, that one is highlighted, and when a key is pressed the corresponding action is performed. If a word is selected, besides adding that to the phrase being constructed, the application also synthesizes its sound.

For this application the calibration procedure is slightly altered. Instead of displaying the targets on the edges of the screen they are placed on the center of each button, thus allowing for better button selection accuracy even when using a low frame rate/resolution camera (the webcam used in figure 3.15 had a framerate of about 10 fps and a resolution of 320x240 pixels).

The disadvantage of this system in relation to Galante and Menezes one is that it requires the user to remain still in relation to the screen, some solutions for this include the use of a second camera that looks at the screen and compensates head movement based on screen pose, or the use of inertial sensors on the glasses to track head movement.

Chapter 4

Conclusion and Future Work

This thesis was aimed at developing an eye-tracking system to be used inside the Oculus Rift development kit 2. It's goal was to complement other bio-signal data (Heart Rate (HR), Respiratory Rate (RR), Electrodermal Activity (EDA), etc), already being analysed on the ongoing study of emotional activations inside VR environments. During the course of this work several prototypes where developed, with the goal of being able to compete with the commercial solutions available. On the accuracy side of things they lagged behind what's commercial available at the moment. But the overall cost of about 1% of the commercial ones, this makes them an attractive solution for researchers, who wish to start studying the gaze of users, without spending a lot of money. This work resulted on an accepted paper to be presented at the 14th International Conference on Remote Engineering and Virtual Instrumentation (REV2017). As for future work, on the hardware side of things a wider field of view camera system could improve the tracking of the users eyes, by increasing the range of motion of the eye that can be observed. If such a camera can be found at a low cost it could extend the capabilities of the system. A better frame rate camera would also allow for the study of saccades. On the software side of things the use of a more complex calibration procedure that takes into account the radial distortion of the lenses could increase the accuracy of the system, putting it at a more competitive level against more expensive solutions. As part of the effort to minimize the effects of motion sickness we also intend to add foveated rendering to the OpenAR platform. Foveated rendering intends to simulate the way our eyes see the world, by applying more detail at the area of gaze fixation. This could also help to minimize the computational effort of rendering a virtual reality scene by reducing the amount of detail needed to render. As part of a collaboration with the department of psychology we also intend to start using the eye-tracker on exposure-based therapies. Particularly on the study of phobias, where we want to analyse the visual response of the user to phobia inducing stimuli.

Bibliography

[1] Wikimedia Commons. (1962) Yarbus eye tracker. [Online]. Available: https://en.wikipedia.org/wiki/File:Yarbus_eye_tracker.jpg

[2] A. L. Yarbus, *Eye Tracking Techniques*. New York: Plenum, 1967 (Russian original 1962).

[3] Wikimedia Commons. (1962) Yarbus the visitor. [Online]. Available: https://en.wikipedia.org/wiki/File:Yarbus_The_Visitor.jpg

[4] Robert J. K. Jacob and K. S. Karn, *Eye Tracking in Human–Computer Interaction and Usability Research: Ready to Deliver the Promises (section commentary)*. Elsevier Science BV, 2003, pp. 573–605.

[5] Seeing Machines. Seeing machines. [Online]. Available: https://www.seeingmachines.com

[6] SmartEye. Antisleep. [Online]. Available: http://smarteye.se/wp-content/uploads/2016/10/anti-sleep.pdf

[7] Leo Kelion (BBC). Caterpillar backs eye-tracker to combat driver fatigue. [Online]. Available: http://www.bbc.com/news/technology-22640279

[8] Toyota. Toyota enhances pre-crash safety system with eye monitor. [Online]. Available: http://www.toyota.co.jp/en/news/08/0122.html

[9] ——. (2008) Outline of pre-crash safety system eye-monitoring feature. [Online]. Available: http://www.toyota.co.jp/en/news/08/image/0122a.gif

[10] Senso Motoric Instruments. Market research. [Online]. Available: http://www.smivision.com/en/gaze-and-eye-tracking-systems/applications/market-research.html

[11] Tobii. Marketing and consumer research. [Online]. Available: http://www.tobiipro.com/fields-of-use/marketing-consumer-research/

[12] Marketing Research Association. Eye tracking heat map. [Online]. Available: http://www.marketingresearch.org/sites/default/files/styles/large_preview/public/article_image/eye.jpg

[13] LC Technologies. Communicate with the world using the power of your eyes. [Online]. Available: http://www.eyegaze.com/eye-tracking-assistive-technology-device/

[14] LusoVu. Eyespeak. [Online]. Available: http://www.myeyespeak.com/

[15] Wikipedia. Eye tracking. Last Accessed: 2016-09-14. [Online]. Available: https://en.wikipedia.org/wiki/Eye_tracking

[16] A. Duchowski, *Eye Tracking Techniques*. Springer, 2003, pp. 55–65.

[17] Metrovision. Electro-oculography by metrovision. [Online]. Available: http://www.metrovision.fr/images/480x300/EO_01.jpg

[18] Chronos Vision. Scleral marker. [Online]. Available: http://www.chronos-vision.de/img/news_scleral_marker.jpg

[19] V. I. P. L. Florida State University. Eye tracker with head rest. [Online]. Available: http://www.fcrr.org/vip_lab/lab/participant_.jpg

[20] Senso Motoric Instruments. Smi eye tracking glasses. [Online]. Available: http://files.ctctcdn.com/ee922129001/3e9683f5-9154-423a-a2cf-23243be7f002.jpg

[21] ——. Smi applications. [Online]. Available: http://www.smivision.com/en/gaze-and-eye-tracking-systems/applications/overview.html

[22] Tobii. Tobii: Fields of use. [Online]. Available: http://www.tobiipro.com/fields-of-use/

[23] Senso Motoric Instruments, “Eye tracking hmd upgrade package for the oculus rift dk2,” 2015, last Accessed: 2016-09-14. [Online]. Available: http://www.smivision.com/fileadmin/user_upload/downloads/product_flyer/prod_smi_eyetracking_hmd_oculusDK2_screen.pdf

[24] ——, “Eye tracking hmd based on htc vive,” 2016, last Accessed: 2016-09-14. [Online]. Available: http://www.smivision.com/fileadmin/user_upload/downloads/product_flyer/SMI_ET_HMD_HTC_Vive_screen.pdf

[25] ——, “Smi mobile eye tracking hmd: based on samsung gearvr,” 2016, last Accessed: 2016-09-14. [Online]. Available: http://www.smivision.com/fileadmin/user_upload/downloads/product_flyer/SMI_Mobile_ET_HMD_SamsungGearVR_screen.pdf

[26] Arstechnica. Oculus rift with eye tracking. [Online]. Available: http://cdn.arstechnica.net/wp-content/uploads/2014/06/DSC_2066-640x425.jpg

[27] “Fove: Technical illustration based on fove prototypes,” 2016, last Accessed: 2016-09-14. [Online]. Available: http://www.getfove.com/

[28] “Starvr,” 2016, last Accessed: 2016-09-14. [Online]. Available: http://www.starvr.com/tag/eye-tracking/

[29] Donegan et al., “D3.2 report on features of the different systems and development needs,” Communication by Gaze Interaction (COGAIN), Tech. Rep., 2006.

[30] F. Mulvey, A. Villanueva, D. Sliney, R. Lange, S. Cotmore, and M. Donegan, “D5.4 exploration of safety issues in eyetracking,” Communication by Gaze Interaction (COGAIN), Tech. Rep., 2008.

[31] The Linux Information Project. (2006) Pipes: A brief introduction. Last Accessed: 2016-09-14. [Online]. Available: http://www.linfo.org/pipes.html

[32] M. de Abreu Ferreira da Silva Brito, “Estimação da pose da cabeça e da direção do olhar de um utilizador,” Master’s thesis, FCTUC, DEEC, Portugal, 2012.

[33] R. Szeliski, *Image processing*. Springer, September 3, 2010 draft, pp. 127–129.

[34] S. Suzuki *et al.*, *Topological structural analysis of digitized binary images by border following*, April 1985, pp. 32–46.

[35] A. W. Fitzgibbon and R. B. Fisher, "A buyer's guide to conic fitting," in *In Proceedings of the 6th British conference on Machine vision*, 1995.

[36] A. Galante and P. Menezes, "A gaze-based interaction system for people with cerebral palsy," in *4th Conference of ENTERprise Information Systems – aligning technology, organizations and people (CENTERIS 2012)*, 2012.

[37] ClassicGOD. Parametric rift dk2 ipd adjuster +/- 4mm. [Online]. Available: www.thingiverse.com/thing:543445

Appendix A

Spacer construction

To construct the spacer we start by 3D printing two different parts for each lens. They where based upon the "Parametric Rift DK2 IPD adjuster", published on "www.thingiverse.com" by "ClassicGOD" [37], used for changing the inter pupilar distance of the lenses.

Figure A.1: Top parts (Rendering)

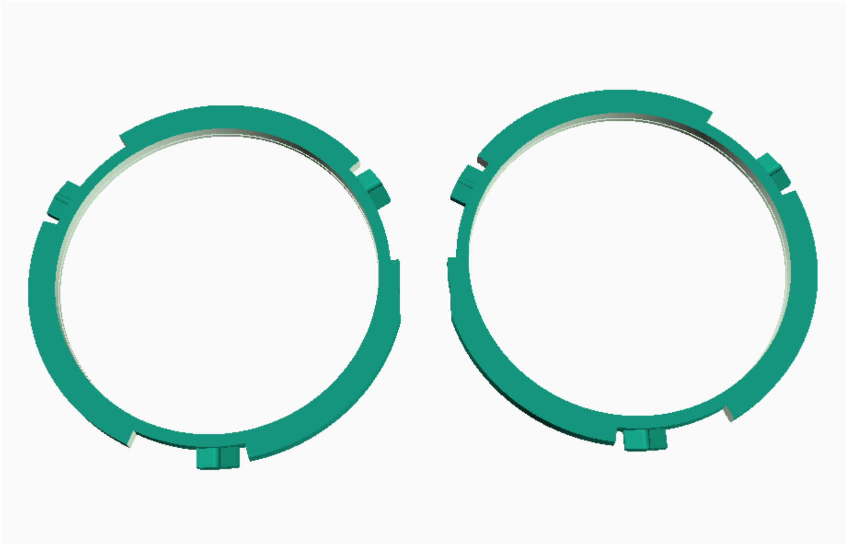

Figure A.2: Bottom parts (Rendering)

[Links to files]

Note that the left and right parts are mirrors of each other and that they have a flat edge on the axis of symmetry. This flat edge is necessary for the spacers to fit correctly on the HMD, and as such when both spacers are placed on the HMD the left and right one should face each other.

A.1 Assembling the pieces together

After printing a small modification needs to be done on the bottom parts. Part of the piece needs to be cut off in order for the camera to fit through the spacer. Several camera positions where tested and the one that achieved the best results was the one shown below.

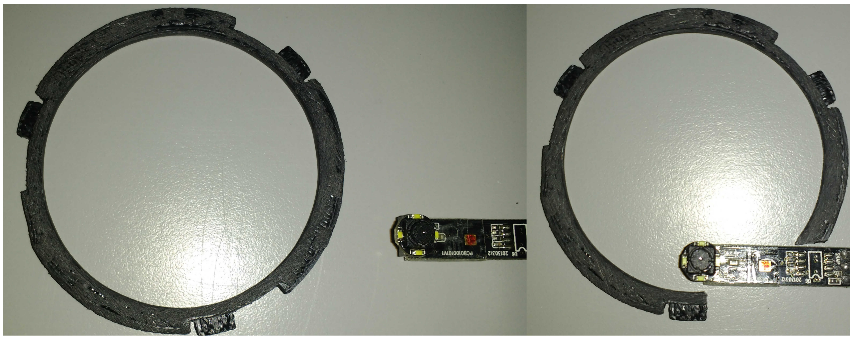

Figure A.3: Bottom part (right) before/after the cut is made to fit the camera

Important note: Remember that the cut on the left bottom part is mirrored!

Side note: I've decided not to include this modification on the deliverable files because the size of this cut will vary depending on the camera being used, and more importantly because trying to print the piece with the cut made in software tended to lead to warping of the printing material during manufacture. By printing the full piece the structural integrity of the material is preserved and the resulting piece fits nicely on the HMD and lenses.
Now we can glue the two pieces together by matching the top and lower parts, as shown below.

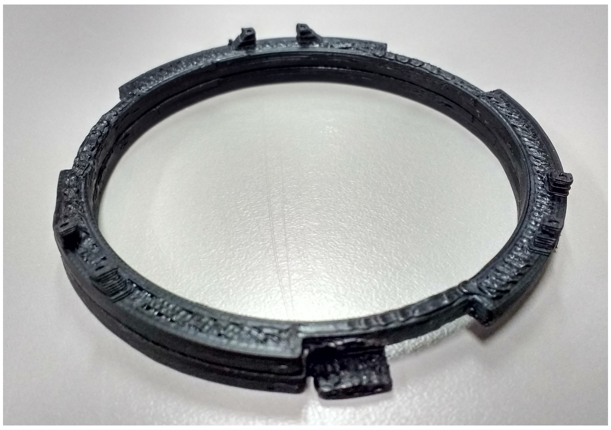

Figure A.4: Completed spacer

Appendix B

IR LED ring

In order to provide a shadow free illumination to the user's eyes, 8 IR LEDs where placed equidistantly from each order on a circular PCB. In order to meet the safety requirements of using IR illumination some calculations needed to be made. After some experimentation it was decided to power the LEDs with a current of $15mA$. Which for the Vishay VSMY1850ITX01 LEDs means they will produce a radiant intensity of about than $I_e = 1.5mW/sr$. This will be rounded up to $2mW/sr$, because given the safety concerns discussed previously is best to consider a worst case scenario. The distance from them to the pupil is greater than $2cm$, so the irradiance, for each LED, is about $E = I_e/d^2 = 0.5mW/cm^2$. Since the LEDs are very close to each other the sum of their average irradiance must be taken into account. For each ring this gives $E_{total} = 8.E_{IR} = 4mW/cm^2$, which is less than the $10mW/cm^2$ recommendation, for periods of over 1000s [30]. If we consider the contribution of both illumination rings the total irradiance is still under this limits. It's important to note that these calculations assume a constant radiant intensity independent of the angle between the LEDs and the pupil. As with most LEDs the radiant intensity decreases as the angular displacement increases. For instance, at 30º the radiant intensity decreases by 10% and at 40º by 20%. As the pupil is at an angle between 30 or 40º, in relation to the LEDs, the real world values are less than those calculated. Supplied with $15mA$ the forward voltage of each LED is about $1.4V$, so the circuit in figure B.1 was designed to powered them.

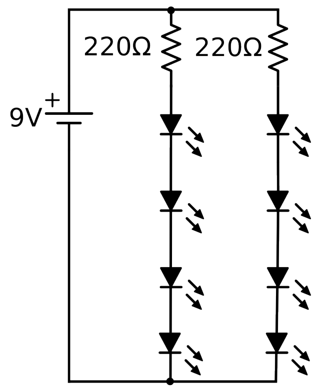

Figure B.1: Ring electric circuit

On the figure B.2, a suggestion of the PCB circuit is presented. This design was chosen so as to simplify the manufacture process. Where "V+" and "V-" represent, respectively, the positive and negative voltage terminals, and "R" represents the resistors.

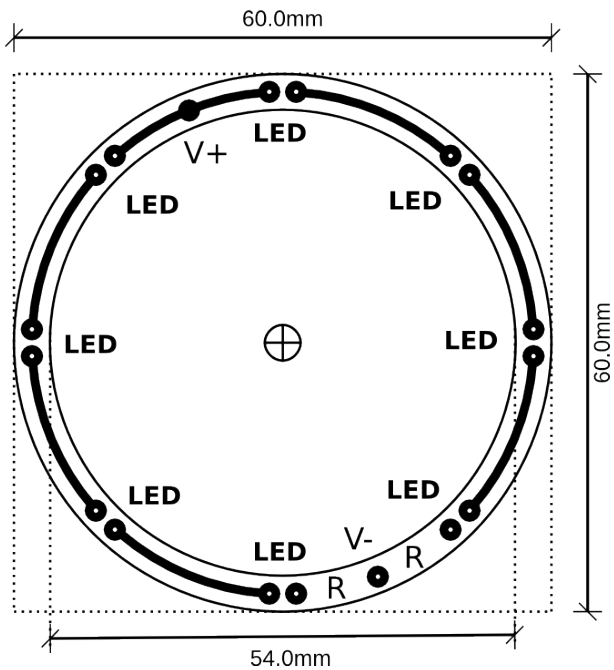

Figure B.2: PCB layout (image not to scale)

Due to their circular, and hollow shape (which leads to a high loss of material), the commercial PCB manufacturers we've contacted, placed a considerable high price on their manufacture (upwards of 30€ per eye, without components). Considering the low cost of raw PCB, and the simplicity of the circuit, a cheaper alternative is to do it in house. Locally, in Portugal, the PCB cost per cm^2 is between 1 or 2 cents, depending on the size of the board. The rings have a diameter of $6cm$, so their cost will be far less than 1€ per ring. The figure B.3 shows the jig used to hold the PCB in place, while the cuts where made using different sized hole saws. Note that for the inside cut of the rings, the hole saw must be chosen measuring it's outside diameter, and for the outside cut the opposite applies. If they are not available a coping saw or a jigsaw could be used, to give some alternatives.

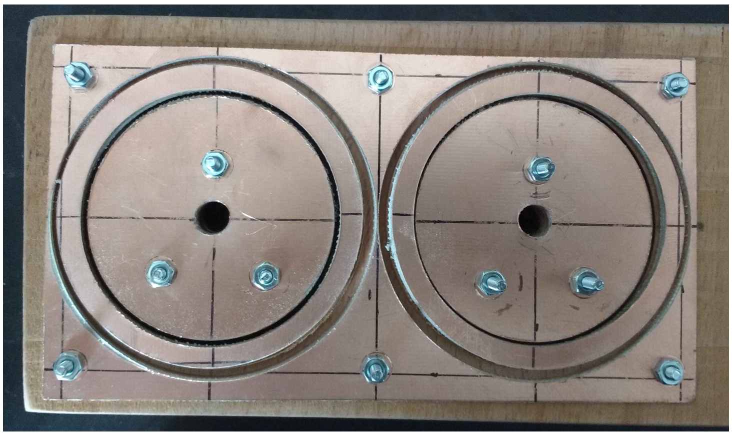

Figure B.3: Jig to cut the PCB rings

Since the PCB layout consists of simple interruptions of the copper layer, etching can be replaced with the use of a blade or file to remove the material. The following image shows the finished rings with the components soldered.

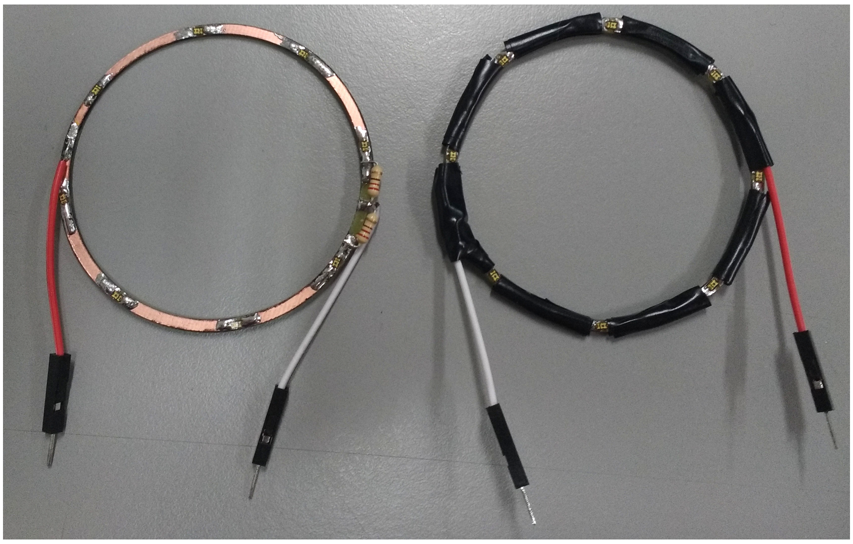

Figure B.4: Finished illumination rings. Left: bare copper, right: insulated with tape

Their total cost was about 20€, of which 8€ was the shipping cost of the LEDs.

Appendix C

Paper to be presented at REV2017

The Importance of Eye-Tracking Analysis in Immersive Learning - A Low Cost Solution

Paulo Menezes, José Francisco, Bruno Patrão

Institute of Systems and Robotics
Department of Electrical and Computer Engineering
University of Coimbra, Polo II, 3030-290 Coimbra, Portugal,
`PauloMenezes@isr.uc.pt`,
WWW home page: `http://www.isr.uc.pt/~paulo`

Abstract. This article discusses the use of Virtual Reality as a tool for supporting learning and some of its opportunities. The importance of using gaze tracking in immersive learning setups is also discussed. This serves as a motivation for the construction of an low-cost eye tracker adapted to an head mounted display (Oculus Rift DK2), which is described. The algorithm for eye tracking as well as the calibration procedure is described, with some results presented.

1 Introduction

Immersive learning systems are attracting the interest of may people and naturally that of teachers and students. The most appealing features of these systems are, on the one hand, the interactive level of control that the trainee/student has on the virtual environment, and on the other hand, the amount of information that it can provide about the user, which can be recorded for later analysis or processed in real-time if needed. To acquire such information a number of sensors may need to be installed in the surrounding space, on the user's body, or both. For the systems where the user can freely move in space, a combination of body tracking system and head mounted display (HMD) is typically required, to provide accurate data of user's body and head movements and poses. For other systems where the user is sitting down performing the experiment, only the HMD's information may be enough provide data related with the evolution of the user's head position and orientation along time. However, in order to track the user's gaze direction, special purpose modifications to the HMD are required and this is normally a challenge due to spatial constrains, illumination artifacts caused by the displayed scenes, etc. In most learning scenarios it is very important to track the user's gaze direction inside the virtual environments, because of its importance to the instructor/teacher, as gaze patterns may help to determine if the trainee/student struggled to understand the subject, and act accordingly, or if he just payed attention to the important details in the right time. Some examples of the use of this can go from adjusting the displayed information, i.e. giving extra information to users who seem to be struggling [7], to providing the instructor/teacher with information of what areas of the text, or visual assets, need to be reworked in order to improve their comprehensibility [3,4]. In the literature we can find many studies on the work of eye-tracking devices, but still there are very few dedicated to HMD-based immersive systems. Only recently companies like SMI and NVIDIA, among others, have worked on this subject but mostly with the purpose of improving the visualization experience through new techniques like foveated rendering [5].

In this paper we address the importance of eye tracking in learning environments and experimentation, followed by the presentation of a low cost eye-tracking solution for HMDs, which includes both hardware and algorithm description, followed by the calibration procedure and presentation of results.

2 Virtual Reality in Learning and Experimentation

The use of Virtual Reality as a learning tool is becoming increasingly accepted given the advantages and possibilities it introduces. Using it, the concept of virtual laboratory may gain another dimension, where the student or trainee beyond learning to control the experiment or system, may feel the sensation of being in the presence of it.

Training of tower crane operators was studied before, and is an example of the cases that may have clear benefits from the use of this technology to avoid financial losses resulting from accidents that may occur during the training process of novices [6].

Virtual labs, also introduce the possibility of extending practice beyond school time, enabling the students to prepare in advance some experiments, or repeat them to later at home. Certain types of experiments may require the presence of a specialist to verify that the appropriate manipulations are being done properly, or just for the sake of safety of the participants, or even to prevent damaging some equipment or expensive materials. Here simulated experiments although not providing the same type of guidance, solve intrinsically the problems related with personal, equipment and material safety.

In what concerns simulations the most common is to have them visualised via the computer screen as plots, charts, tables, or animations, but with the availability of VR/AR systems we will see certainly a change towards their use, and there are very good justifications for that. The first is that they make possible to train manipulation abilities, in order to learn not only the sequence of the procedure, but also train the exact gestures to be performed. Examples are the preparation of medicine students to gain suturing or palpation skills. This can be done using a VR system composed of an head mounted display (HMD) and some haptic devices that enable student to not only see but also sense the forces and vibrations of the procedure, as if he/she was handling the needle holder. Another reason to use VR or AR for learning is linked with the known advantage of learning by doing versus learning by seeing. In reality this create opportunities for the student to (virtually) touch, or observe in place by exploring the subject from every possible viewing angle as in the real world [2].

3 Learning to look at the right things at the right times as part of some training processes

It is common sense that vision plays a central role in any interaction activities developed by humans and most animals. One of the reasons is that through vision we can learn very different types of information about a subject without reaching and touching it. In addition it enables us to sneak a glance at some interesting object and continue performing some main task without interruption. Let's consider an example: While driving a car, we keep our eyes on the road, but as some speed limit sign appears we need to check if our current speed is below the limit or not. For this we peek at the speedometer on the dashboard to get the intended information, but return immediately the attention to the road.

The design of panels and dashboards is normally guided by the knowledge of the human vision related cognitive processes. This influences the choice of the distribution of instruments, or the way to attract the attention to the right information at the right moment.

In complex systems the number of gauges, together with the complexity of the task itself, may render the operator activities very stressful and demanding. In many of such cases it is not viable to have beeps or blinking lights to attract the attention to the information displays, as they tend to turn into noise that may contribute to degrading the driver performance, and lead to a desensitisation process that makes the operator to stop noticing them.

For this reason during the training process the operators/pilots must learn some mandatory procedures which include the periodic verification or reading of some displays or gauges in predefined sequences. This is typically important to confirm if the related quantities are within acceptable limits or if some corrective operation must be executed.

During normal training sessions, the instructor may observe directly if the trainee does, as expected, the periodical checking of the important displays, or consults the appropriate one prior to do some action or manoeuvre, as a way to choose the appropriate commands or verify the related safety.

3.1 Immersive Learning Systems

It is known that immersive systems based on HMDs may be an important learning tool, given that it introduces the possibility of repeating at will the training in simulated situations, without physical, financial or health risks, neither for the trainee or for third parties. Using this type of systems, the training process, that typically is done in the presence of an instructor, may now also be done individually, eventually at home. Nevertheless, these systems are still some limitations, like: (1) The resolution of the display devices and field of view, although having been improved a lot recently, are still below the human capabilities. For this reason and to have sufficient detail to enable the easy reading of the elements, either their minimal size or maximum distance has to be limited. This has the consequence of reducing the visible area of the dashboard for a given static head pose, unlike in the real world, where a pilot or operator can see most, if not all, of the dashboard, just by moving the eyes left and right. The use of current HMDs still imposes the rotation of the head to visualise the remaining parts, as a result a trainee may easily forget about reading what is out of sight. (2) The HMD occludes the eyes of the trainee, so the instructor cannot see them. This indeed does not allow an instructor to know where the trainee is looking at. As a result it is not possible to know if the appropriate instrument readings were done when they were supposed to happen.

The first mentioned limitation could be overcome from using higher resolution displays, but these impose higher demands on the graphics unit and required bandwidth. As already applied in some HMDs for reducing the cybersickness effects, the detail of the peripheral regions is reduced. But the peripheral region is defined with respect to the display center and not to the gaze direction. This is called foveated rendering and some companies have been working in it [5], still considering very high resolution displays receiving full resolution video, even if a great part of it is blurred. An interesting solution would be to reduce the dimensions of the generated rasterizations and vary their position on the screen so that they are always centred on the user gaze direction. From this it becomes obvious that the introduction of an eye-tracking mechanism would serve as a basis for solutions of both of these problems. Hereafter we will focus on the second case.

4 Eye-tracking in learning

An eye-tracker suitable for use inside an HMD can be a valuable tool for VR-based training processes. For sessions conducted by an instructor, he/she can have a view of what is the focus of attention of the trainee in every instant in time. This can be used, for example, to understand if there are some elements that take longer for the student to understand that require some intervention or explanation from the instructor [1].

With the help of this, the instructor can also observe, either directly, with the help of heat maps, or another method, if the trainee has done the required readings of the instruments at the appropriate times or in the right sequence before some particular actuations, as is defined on the procedures to be learnt.

In the cases that the system is to be used to practice in an unsupervised manner, the procedures to be learnt may be used to generate a set of rules that will serve to evaluate the trainees performances. Here, the gaze information can be used to automatically check if the verifications were done properly, and if the required gauges were read as expected prior to some actuations. This enables the identification of different cases: 1) The trainee did not read the necessary gauges/displays and by consequence did not perform a required action, performed the action not correctly, or even performed the action correctly by chance. 2) The trainee did the required readings but has chosen the wrong actions or incorrect way of doing them. 3) The trainee did the required readings and used them to choose the appropriate actions.

5 A low-cost eye-tracking solution for Oculus Rift DK2

Several researchers and developers have acquires Oculus Rift pre-release development kits DK1 or DK2, to anticipate the development of new applications for these devices or research purposes. When compared with those that preceded them, these devices have very interesting characteristics in terms of field of view and development support. The configuration of the device, in terms of available space and the detachable lenses, creates a perfect opportunity for introducing the necessary modifications to support an eye tracking solution.

5.1 Customization of the HMD for eye tracking

A preliminar analysis of the characteristics of the device and the typical usage conditions was necessary to elaborate a solution that respects the requirements of introducing none or minimal interference to the user during usage.

To get an adequate view of each of the user eyes, a pair of cameras was installed under the respective lenses, in a position that minimises the occlusion of the viewed scene. And in order to provide good illumination to the user's eyes, infrared illumination was chosen, that facilitates the segmentation procedure by increasing the contrast between the pupil and the iris (dark pupil effect). To place the cameras on the HMD a spacer was designed and produced using 3D printing. Figure 1 shows the assembled the lens, spacer and the infrared illumination ring, and the details of the parts used. It should be noted that the chosen cameras are low cost endoscopy-like ones, which as in most cameras on the market, came with an IR suppressing filter. This had to be removed, and in turn replaced with one that attenuates the visible light reflected on the eyes, while allowing the IR light to pass.

Fig. 1. Left: Camera board and sensor with 90 degrees bent connections; Lens; IR filter (film); Lens on spacer with IR ring; Spacer with camera below; Right: Oculus Rift DK2 with left lens out, showing spacer and camera installed; on the right side we can see the illumination ring in mounted on the respective lens.

The resulting setup can be seen on the right side of figure 1, where the left illumination ring and lens was left out to show the respective spacer and camera in place. It can also be seen the cabling of the cameras and illumination ring.

5.2 Algorithmic support for the eye-tracking

Using the installed hardware it's now possible to have clear near-IR images of each of the user's eyes. These images are initially filtered using a low pass Gaussian filter. A common approach is to perform pupil segmentation applying a binarization on grey-level images, which tend to compress to the same range, distinct zones that may produce different responses on the various colour channels of RGB cameras but tend to have the same luminance levels. This typically leads to the generation of artifacts that need to be removed using morphological operations or other.

A careful analysis of histograms obtained from the 3 colour channels showed that (as expected) the red channel could provide the best segmentation results, due to the proximity to the IR band.

After the segmentation, the contours of the pupil are extracted by finding the chain of connected border pixels. Due to the eye movements the shape of the pupil area can vary from a circular to an elliptic one. Therefore the use of an ellipse fitting algorithm enables the extraction of the pupil center at the intersection of the two axes. As no instantaneous variations of the threshold value are expected in normal conditions, failure to detect the elliptical region indicates the occurrence of a blink. The basic algorithm is described in figure 2.

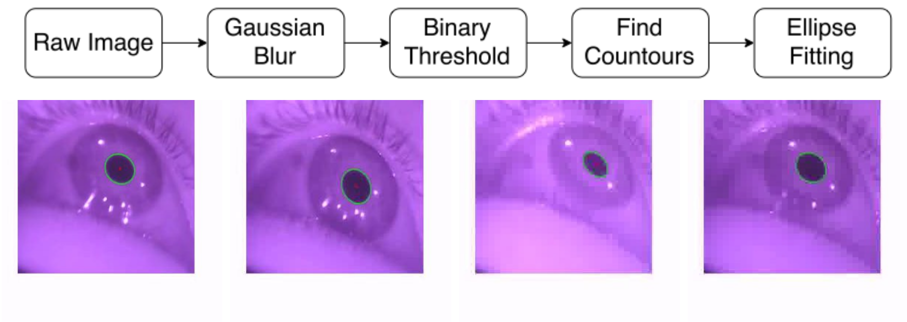

Fig. 2. Top:Block description of the processing algorithm; Bottom: Example result of detection of eye pupil and ellipse fitting

5.3 Calibration procedure

As the interest of the use of this type of eye tracker is to know to which region of the scene the user is looking at, there is a need to obtain a mapping between coordinates of the observed pupil center and the screen coordinates of the point the user is looking at.

For obtaining correspondences between screen coordinates and tracker coordinates, the typical procedure is to show some dot or cross on the screen and ask the user to look at it, and subsequently register the two pairs of coordinates.

In order to improve the stabilization of the user gaze during calibration, instead of displaying a static target we opted to add a shrinking one, whose movementn attracts the attention of the user towards its center.

This calibration has to be performed typically for each usage, as the position of HMD with respect to the eye may vary slightly. For this a set of 9 points are shown sequentially on the "screen" and pupil center coordinates are recorded for each of them. This is performed first for the left eye and after for the right eye, using a dark screen for the opposite eye. From the obtained pairs of coordinates, a mapping function **f** may be estimated depending on the type of mapping chosen (in most cases linear or polynomial). The mapping is then performed as

$$\mathbf{S} = \mathbf{f}(\mathbf{P}), \tag{1}$$

where $\mathbf{S} = (S_x, S_y)$ are screen coordinates of the point the user is looking at and $\mathbf{P} = (P_x, P_y)$ are the corresponding image coordinates of the pupil center. Note that different mapping functions will be obtained for the left and right eyes.

6 Calibration Results and Analysis

The calibration of the eye tracker for obtaining a mapping function between eye- and screen-coordinates was done using both a linear and a polinomial approaches. The linear mapping consists in reducing the above function to $\mathbf{S} = \mathbf{FP}$, where $\mathbf{F}$ is a 3 by 3 matrix, whose coefficients can be estimated using a least squares solution. The calibration errors were then compared for the linear global, linear per quadrant, and polynomial cases. The following tables shows the obtained values for the errors in pixels between the target locations and those obtained via the mapping functions.

Method	Mean error	Abs Max error
Linear global	27.3331	58.9140
Linear per quadrant	8.3500	27.8960
Polynomial 2nd order	4.9010	16.2896
Polynomial 3rd order	3.4066	11.3599
Polynomial 4rd order	2.4244	9.1497

The difference between the linear and the linear per quadrant is that the for the second the display region is divided in 4 quadrants producing 4 separate mapping matrices.

These results show as expected that the gloal linear mapping presents the worst results, but when aplied locally (per quadrant) can improve substancially. The polynomial approach shows as expected much better results. The reason is clearly understood from the fact the iris locations are obtained from the projection of a point from an approximately spherical surface onto a plane through a projective projection.

Using the obtained mapping functions it is now possible to infer the region of the scene the user is looking at, as can be seen in the example shown in figure 3.

Fig. 3. Example of the output of eye tracker (red cross) mapped onto a plane cockpit shown to the user.

7 Conclusion and future work

The paper presents the development of an eye tracker for inclusion in an Oculus Rift DK2, in terms of both mechanical construction and image processing algorithms. Preliminary results in terms of mapping of the estimated gaze onto the screen image plane was shown.

Future work includes testing other types of mapping functions will be evaluated, namely polinomial approximations. Development of method for selection of the zones the user is looking at, at the screen level in an initial stage and in the 3D scene as a later one. For the 3D case it implies the development of models for the gaze directions based on light beams given the uncertainty of the estimates and the discrete nature of the rasterization process.

References

1. Ha, J.S., Byon, Y.J., Baek, J., Seong, P.H.: Method for inference of operators' thoughts from eye movement data in nuclear power plants. Nuclear Engineering and Technology 48(1), 129 – 143 (2016)
2. Menezes, P., Chouzal, F., Urbano, D., Restivo, T.: Augmented reality in engineering. In: 19th International Conference on Interactive Collaborative Learning and 45th International Conference on Engineering Pedagogy (2016)
3. Móro, R., Bieliková, M.: Utilizing gaze data in learning: From reading patterns detection to personalization. In: UMAP Workshops (2015)
4. Oglesby, J.L.M.: An Adaptive Visual Learning Approach for Waterborne Disease Prevention in Rural West Africa. Master's thesis, W. Kentucky University (2016)
5. Patney, A., Kim, J., Salvi, M., Kaplanyan, A., Wyman, C., Benty, N., Lefohn, A., Luebke, D.: Perceptually-based foveated virtual reality. In: SIGGRAPH Emerging Technologies (2016)
6. Patrão, B., Menezes, P.: An immersive system for the training of tower crane operators. In: Exp.at'13: The 2nd Experiment@ International Conference. Coimbra, Portugal (2013)
7. Song, J.C., Nam, S., Song, K.S., Park, S.Y.: Development of a personalized learning system using gaze tracking system. WSEAS Transactions on Computers 14, 264–271 (2015)