High performance and low cost eye tracking using differential stroboscopic lighting

Frank Helbert Borsato

THESIS SUBMITTED
TO THE
INSTITUTE OF MATHEMATICS AND STATISTICS
OF THE
UNIVERSITY OF SÃO PAULO
FOR THE
DOCTORAL DEGREE IN SCIENCE

Program: Computer Science

Advisor: Professor Carlos Hitoshi Morimoto

This research is supported by Fundação Araucária (projeto DINTER UTFPR/IME-USP)

São Paulo, August, 2016

High performance and low cost eye tracking using differential stroboscopic lighting

This is the original thesis written by Frank Helbert Borsato,
submitted to the Committee Members

Abstract

An eye tracker is a device that measures eye movements and is commonly used to estimate the point of gaze. Common techniques used for eye tracking in the 90's include electrooculography, contact lenses, and video-based methods. Today, video-based eye trackers have become dominant.

Despite recent advancements in eye tracking technology, improving overall performance and reduced size, low cost high performance eye trackers are not yet available. In this thesis we introduce a novel eye tracking technique we call Differential Stroboscopic Lighting (DSL), and show two possible implementations using low end webcams, one synchronous and one asynchronous. DSL is an extension of the differential lighting (DL) technique developed in the 90's for analog cameras. DL uses two set of infrared light sources to alternately generate bright and dark pupil images. The pupils are then robustly detected from the difference of two consecutive video frames. Though DL is a simple and computationally efficient solution for robust pupil detection, its implementation using low cost digital cameras poses challenges due to the lack of external video synchronization and the use of rolling shutters that causes each scanline to be shifted in time. DSL overcomes these problems by synchronizing the stroboscopic lighting with the video frames using lightweight image processing algorithms.

Because DSL demands a relatively low computational load, it is appropriate for high speed cameras and low end computers such as mobile phones and tablets. The use of stroboscopic lighting also enhances the image quality, significantly reducing motion blur and artifacts from the rolling shutter. As a result of better image quality, DSL based eye trackers also achieves better accuracy and precision. In this thesis, we also propose an accurate and efficient pupil contour refinement algorithm to take full advantage of DSL properties in an actual eye tracker. Our experimental results using the DSL eye tracker prototype, that uses a low cost PS3-Eye camera that can achieve frame rates near 200 Hz, shows that the proposed method can outperform existing state-of-the-art algorithms in terms of accuracy and computational performance.

Keywords: eye tracking, high performance, low cost, asynchronous stroboscopic light, eye safety, rolling shutter, synchronization, clock drift, pulse minimization, Kalman filtering, human-computer interfaces

Resumo

BORSATO, F. H.. **Rastreador de olhar de alto desempenho e baixo custo baseado em iluminação estroboscópica diferencial**. 2016. 144 f. Thesis (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2016.

Um rastreador de olhar é um dispositivo que mede os movimentos do olho e é comumente utilizado para estimar o ponto observado pelo olhar. Técnicas comuns nos anos 90 incluiam eletro-oculografia, lentes de contato especiais, e métodos baseados em análise de vídeo. Nos últimos anos, a análise de vídeo se tornou dominante.

Apesar dos avanços recentes na tecnologia de rastreamento de olhar, melhorando o desempenho e reduzindo o tamanho, dispositivos de baixo custo e alto desempenho ainda não estão disponíveis. Nesta tese nós introduzimos uma nova técnica de rastreamento de olhar que chamamos de Iluminação Estroboscópica Diferencial (IED), e mostramos duas possíveis implementações usando câmeras de baixo custo, uma síncrona e outra assíncrona. IED é uma extensão da técnica de Iluminação Diferencial (ID), desenvolvida nos anos 90 para câmera analógicas. ID utiliza duas fontes de iluminação por infravermelho para gerar imagens alternadas de pupila clara e escura. A pupila é então detectada de maneira robusta pela diferença entre imagens consecutivas. Embora ID seja uma solução simples e computacionalmente eficiente para detecção da pupila, sua implementação utilizando câmeras digitais de baixo custo apresenta desafios devido a falta de uma saída de sincronização e o uso de obturadores digitais de cortina, que geram linhas deslocadas no tempo em um mesmo quadro. IED supera esses problemas sincronizando iluminação estroboscópica com os quadros da câmera utilizando algoritmos leves de processamento de imagens.

Considerando que a técnica IED requer uma baixa carga computacional, ela é apropriada para câmeras de alta velocidade e computadores de baixo desempenho, como celulares e *tablets*. O uso de iluminação estroboscópica melhora a qualidade da imagem, reduzindo significativamente artefatos de movimento e artefatos do obturador. Como resultado de imagens melhores, rastreadores de olhar baseados em IED alcançam uma melhor acurácia e precisão. Nesta tese, também propomos um algoritmo acurado e eficiente para o refinamento do contorno da pupila para obter o máximo das propriedades da IED. Nossos resultados experimentais usando um protótipo de rastreador baseado em IED, que utiliza uma câmera PS3-Eye que alcança taxa de quadros próxima a 200 Hz, mostra que o método proposto pode superar os algoritmos do estado-da-arte em termos de acurácia e desempenho computacional.

Palavras-chave: rastreamento de olhar, alto desempenho, baixo custo, iluminação estroboscópica diferencial, segurança para os olhos, obturador de cortina digital, sincronização, sincronia de relógio, minimização de pulso, filtro de Kalman, interfaces humano-computador

Contents

Acronyms

ADSL	*asynchronous differential stroboscopic lighting.*
AE	*auto exposure.*
ANOVA	*Analysis of Variance.*
CR	*corneal reflection.*
CRT	*Cathode ray tube.*
DL	*differential lighting.*
DoG	*derivative of gaussian.*
DSL	*differential stroboscopic lighting.*
EGT	*eye gaze tracker.*
ESR	*Equivalent Series Resistance.*
ET	*eye tracker.*
ETS	*extended temporal support.*
FWHM	*full width half maximum.*
HCI	*human-computer interaction.*
HCI	*human-computer interface.*
IC	*integrated circuit.*
ICNIRP	*International Commission on Non-Ionizing Radiation Protection.*
IEC	*International Electrotechnical Commission.*
IR	*infrared.*
LED	*light-emitting diode.*
MOSFET	*metal-oxide-semiconductor field-effect transistor.*
MPE	*maximum permissible exposure.*
NIR	*near-infrared.*
PoG	*point of gaze.*
pp	*polypropylene film.*
ppm	*parts per million.*
QVGA	*quarter video graphics array.*
RANSAC	*RANdom SAmple Consensus.*
RC	*resistor-capacitor.*
RMS	*root mean square.*
ROI	*region of interest.*
SMA	*simple moving average.*
TCR	*temperature coefficient of resistance.*
USB	*Universal Serial Bus.*

UVC *USB Video Class.*
VGA *video graphics array.*
VOG *Video-oculography.*

List of Symbols

$\mathbb{E}$	Exposure value, driver specific
Δe	Exposure period, in seconds
t_0	Start of frame integration
t_{strobe}	Lag between strobe and t_0
Δ_{strobe}	Strobe length, in μs
Δt	Frame period, in seconds
S	Total number of scanlines in a sensor
N	Visible scanlines
D_0	Invisible scanlines before N
D_z	Invisible scanlines after N
t_{line}	Time a scanline takes to be read
h_{stripe}	Height of stripe, in scanlines
β	Exposure conversion coefficient
$\hat{\beta}$	Estimated value for β
δ_{is}	Inter-frame setup time, in μs
Δ_{clke}	External clock period
Δ_{lag}	Difference between t_0 and t_{strobe}
D_{stripe}	Stripe position, in scanlines
Δ_{drift}	Difference between Δe and Δt
ξ	Time to move the stripe to visible scanline range
$\dot{\xi}$	Time to move the stripe to invisible scanline range
$\mathbb{H}$	Inverted cumulative histogram
ϱ	CR center of mass weighting constant
υ	Ray size multiplier constant
η	Translation threshold for ETS
E	Irradiance, given in W/m^2
Φ	Radiant power, given in W
α	Planar angular subtense, given in radians

List of Figures

List of Tables

Chapter 1

Introduction

The first devices constructed to measure the eye movements date back the late 19th century. Being mostly mechanical, they were highly invasive and uncomfortable. Only on the very beginning of the 20th century the photographic principle was introduced, becoming the dominant technique for recording eye movements (Holmqvist *et al.*, 2011). Despite the great discoveries they enabled, they were still large and complex, constraining studies to laboratory settings (Buswell, 1935; Wade, 2010; Yarbus, 1967).

In the last decades the scenario has changed considerably. Estimating the eye movements have become more accessible, growing in popularity among researchers from many disciplines. Usability analysts, sports scientists, neurophysiologists, game developers, and others, benefit from eye movement data for different purposes (Duchowski, 2002).

Though a number of techniques have been investigated along the century to estimate the eye movements, *video-oculography* (VOG) have emerged as dominant and is wide spread in research and commercial applications. A VOG is a video-based device for the measurement of eye movements (Kim *et al.*, 2006). By means of a calibration process, it is also possible to estimate the *point of gaze* (PoG). This later device, capable of telling the point of gaze, is also referred as an *eye gaze tracker* (EGT) (Morimoto and Mimica, 2005b). In the rest of this thesis, we will refer to these devices as *eye trackers* (ETs) since it seems to be the most common used name.

A typical video-based ET is composed by a video camera to record eye images and a computer to estimate the movements, store, and analyze the gaze data. They can be used in either a remote or head-mounted configuration. In a remote system, the camera is fixated, typically bellow or close to the computer monitor being used in the experiment. In a head-mounted configuration, the camera is fixated in eyeglasses or helmet-like apparatus designed to be worn. While remote systems are less intrusive, they are typically less flexible, as the allowed movements are restricted by the field of view of the camera being used to image the eye. Head-mounted systems on the other hand, enable free head movement, allowing the observation of gaze behaviors in largely unconstrained settings.

Video-based eye tracking methods typically fall within two categories: model-based and feature-based methods (Hansen and Ji, 2010a). Model-based methods find the best fitting model for a given image. For example, Ivins and Porrill (1998) use a five-parameter deformable iris model which is adjusted using non-linear least-squares minimization over the image data. Hansen and Pece (2005) present an active contour model based on image statistics which combines particle filtering with the expectation maximization algorithm. Model-based approaches have the potential to be very accurate, but typically trade-off accuracy by computational speed. They also tend to be sensitive

to the initial model state.

Feature-based methods, on the other hand, depend on the detection and localization of eye features within the image. Though the features vary across algorithms, intensity gradients are commonly employed to locate eye corners (Zhu and Yang, 2002), *corneal reflections* (CRs), the pupil (Ohno *et al.*, 2002) or the limbus, the boundary between the iris and sclera (Zhu and Yang, 2002).

In the work from Ohno *et al.* (2002), pupil candidates are segmented iteratively as dark circular regions surrounded by brighter pixels. Pixels are then scanned radially from the center of the best candidate pupil, and those brighter than a calculated threshold are considered as pupil border locations. Goni *et al.* (2004) and Keil *et al.* (2010) use a histogram-based threshold calculation with the pupil center calculated as the centroid of the segmented region. Recent works provide more elaborate solutions. Świrski *et al.* (2012) use the convolution response of a Haar-like center-surround feature to obtain a gross estimation of the pupil position, which is then segmented using a threshold estimated using k-means clustering of the image histogram. The pupil is assumed to be darkest element in the image. A similar approach is used by Kassner *et al.* (2014b).

After the best pupil or iris candidates are computed, their contour can be estimated by ellipse fitting (Fuhl *et al.*, 2015a; Li and Parkhurst, 2005; Ohno *et al.*, 2002; Świrski *et al.*, 2012; W. Fuhl, 2016; Zhu and Yang, 2002). To improve the robustness of the estimation to outliers, statistical methods such as *RANdom SAmple Consensus* (RANSAC) have been used (Fuhl *et al.*, 2015a; Li and Parkhurst, 2005; Świrski *et al.*, 2012).

Feature-based methods tend to make strong assumptions about the image quality, and accordingly, are typically implemented using active illumination. Controlling the illumination makes feature-based methods attractive because the pupil and CRs become relatively easy to find, and are directly related to gaze (Hansen and Ji, 2010b). In addition, feature-based methods are typically less complex and easier to implement than model-based approaches. Accordingly, methods employing active illumination are predominant in both the literature and commercial systems (Hansen and Pece, 2005), a few examples include Ebisawa (1995); Keil *et al.* (2010); Morimoto *et al.* (1999); Pupil-Labs.com (2016); SensoMotoric Instruments (2016); Tobii AB (2016).

Video based ETs typically utilize *near-infrared* (NIR) light to improve image quality and highlight the pupil. The illuminators can be arranged to produce dark or bright pupil images. Bright pupils are created using coaxial illumination that causes retro-reflection at the retina. The effect is responsible for 'red eyes' on conventional flash photography, and has been exploited by medical devices and ETs for a long time (Charlier and Hache, 1982; Ebisawa and Satoh, 1993; Morimoto *et al.*, 1999; Zhu *et al.*, 2002). By using structured lighting to produce both bright and dark pupil images, ETs can robustly detect the pupil at a low computational cost (Ebisawa, 1998; Morimoto *et al.*, 1999). The technique is called *differential lighting* (DL), and employs the difference image to highlight the overlap of the bright and dark pupil images in consecutive frames.

1.1 Motivations

Despite recent developments in eye tracking technology, with improvements on size, usability, performance, and cost, mobile eye tracking technology still faces several challenges. Commercial head-mounted ETs are still expensive for general use and typically rely on closed source software

and hardware (Kassner *et al.*, 2014b). Particular research domain and studies require multiple eye trackers, making the price especially prohibitive (Kim *et al.*, 2014).

As an alternative to commercial devices, affordable eye tracking solutions have been suggested (Ferhat *et al.*, 2014a; Kassner *et al.*, 2014a; Kim *et al.*, 2014; Lukander *et al.*, 2013), but they are typically limited to 30 Hz or 60 Hz. Higher sample rates might be used to create more accurate and precise eye trackers (Hennessey *et al.*, 2008), and are necessary to estimate fast eye movement trajectories (saccades) during eye gestures used for communication and interaction. They also give researchers more flexibility for data filtering and more reliable event detection using velocity-based algorithms. According to Andersson *et al.* (2010), low sampling frequency results in uncertainty that demands more data and post-processing, and might not be available in a mobile environment. Duchowski *et al.* (2016) argue, based on the Nyquist-Shannon sampling theorem, that a 60 Hz sample rate limits the observation of saccades of at least 33 ms of duration, while Enright (1998), provides evidence that saccadic peak velocity cannot be accurately estimated at $60Hz$ for saccades shorter that 10°, typical of reading. Faster eye trackers might also create more responsive gaze interfaces by reducing latency.

Increasing the sample rate aggravates an open issue with video-based ETs, the considerable processing power that video processing requires. While in a laboratory setting, a remote ET is allowed to run in a separate powerful machine, mobile ETs must conserve power to meet operating times required for long-term studies in research and commercial applications (Bulling and Gellersen, 2010). The processing demands limit the sample rate and response time of mobile devices. For example, the ET developed by Nagamatsu *et al.* (2010) was capable of a sampling rate of 10 Hz, using a separate notebook with 2.2 GHz microprocessor. Another example is the work from Ferhat *et al.* (2014b), in which a 3 Hz sample rate was achieved using a Raspberry Pi platform. Moreover, studies report the attainable frame rate of the system, overlooking momentary processing demands or the need to run the application that will use the gaze data (see for example Lukander *et al.* (2013) and Kassner *et al.* (2014b)). Considering that mobile devices are intended to allow free head movement, such as on natural eye movement research, head movement compensation is required. The most common modality is the analysis of the video of a scene camera (Kinsman *et al.*, 2012), which adds more weight to the already overloaded processing pipeline.

Given the gains a high sampling rate might enable and the importance of gaze studies on naturalistic setups, commercial head-mounted systems are increasing in performance. SMI recently launched a mobile binocular solution capable of 120 Hz sample rate, using a Samsung Galaxy Note 4 as remote unit (SensoMotoric Instruments, 2016). Tobii also provides a high speed wearable solution, reaching 100 Hz and 120 minutes battery recording time (Tobii AB, 2016). A recent open software (and semi-open hardware) initiative designed to be a compromise between price and performance is available from Pupil-Labs, with the 120 Hz sample rate head-mounted solution available from € 800 to € 1500, for the monocular and binocular versions, respectively (Pupil-Labs.com, 2016).

1.2 Objectives

One of the original aims of this project is to create a low cost and high performance eye tracker for mobile gaze enhanced applications and make it available as an open source software and hardware for the international eye tracking community.

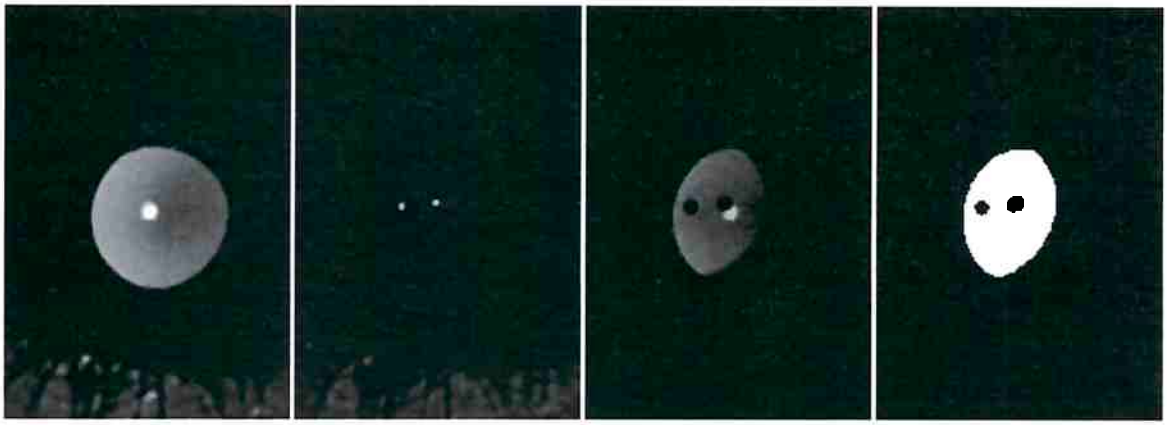


Figure 1.1: *Pupil detection using the differential lighting technique. a) Bright pupil; b) Dark pupil; c) Difference image; d) Thresholded difference image showing the pupil overlap region.*

In this work we investigate alternative ways to the development of low cost and high performance video-based ETs. The cost is kept low by allowing the use of consumer grade cameras. Such cameras lack the hardware synchronization means necessary for the original DL to work. We propose extensions to the DL that allows the structured illumination to be synchronized with the camera frames by means of image processing algorithms. We also propose a simpler ET design that allows the structured lights to run asynchronously, reducing the assumptions about the illumination behavior.

With respect to the performance, the improvements are associated to the temporal and spatial resolution, coupled with reduced computational resources. This goal is pursued by exploiting the lightweight pupil detection provided by the DL technique, associated to high quality images provided by the use of stroboscopic light (very short light pulses). As the effective exposure time is reduced, sharper images are expected during high speed eye movements. As a result, a higher accuracy is expected. By developing computationally efficient algorithms, we aim at reducing processing that video usually demands, increasing the potential of high speed video-based gaze tracking integration into mobile environments and wearable devices.

1.3 Challenges

The DL technique assumes each frame is illuminated by only one source, producing either a bright or dark pupil. This was assured on the analog implementations by triggering the lights alternated on the even and odd fields of interlaced cameras (Morimoto *et al.*, 2000). Using the difference of two consecutive frames (Figure 1.1c), high contrast regions are detected as pupil candidates as seen in Figure 1.1d. The detected region in the difference image corresponds to the overlap between the dark and bright pupils, and is the result of further refinement to remove noise artifacts and outlier regions.

In this work we exploit the effect using modern low cost cameras, which in majority employ a mechanism to capture images called rolling shutter. It allows reduced costs in chip production and added sensitivity, as the integration process continues during acquisition. The drawback is a shift in time between each image line. This results in well known effects of wobble and skew when fast moving objects are imaged. Particularly during saccades, the pupil is one of such high speed

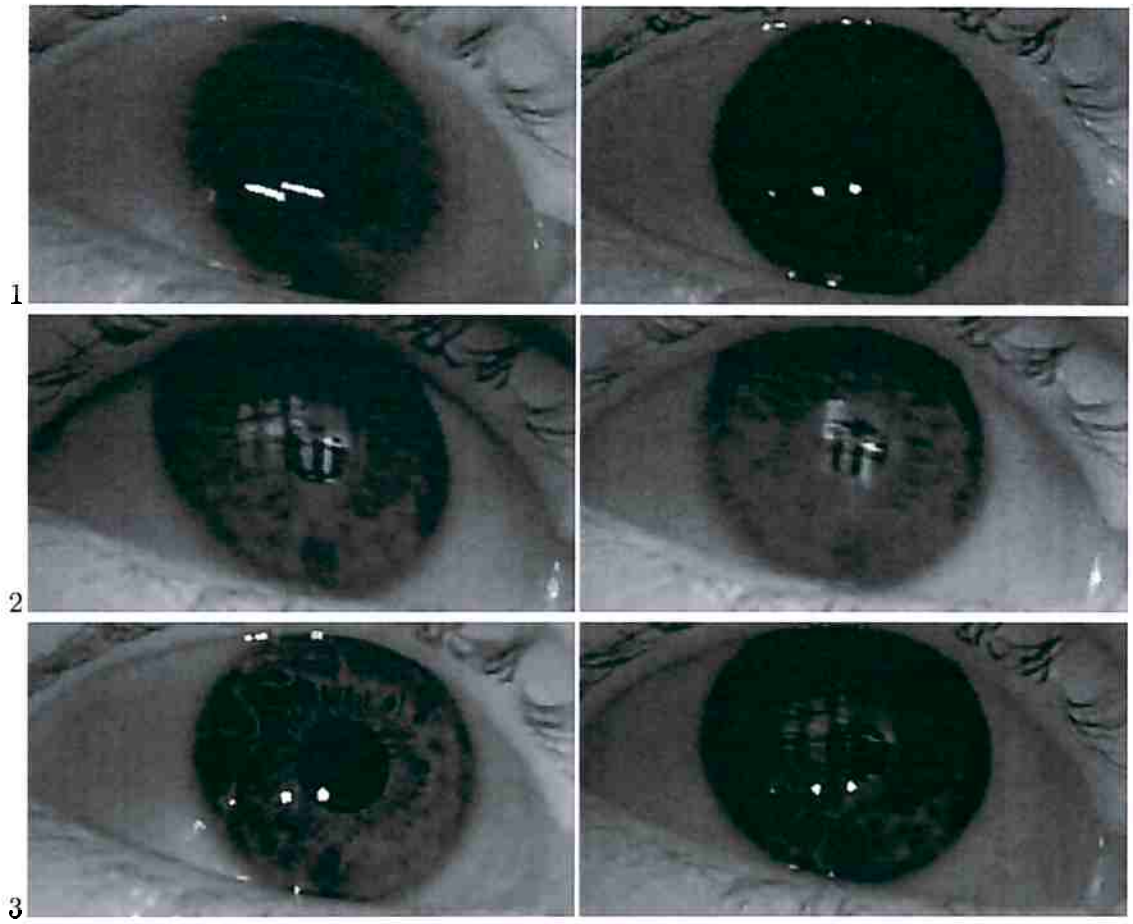


Figure 1.2: *Eye captures during large amplitude saccades (greater than 50°). The gain and exposure are the same, except for image 2a, where the exposure was reduced. 1ab) dark room (measured 20 lux at eye) and continuous NIR light; 2ab) next to window in a sunny day (measured 1900 lux at eye) and no artificial illumination; 3a) dark room and stroboscopic NIR illumination; 3b) next to window and stroboscopic NIR illumination.*

moving objects, and thus, suffer from deformations and blurry silhouette.

To improve the image quality, stroboscopic illumination is used (compare Figure 1.2-3a, taken at 30 Hz and stroboscopic illumination with Figure 1.2-1b, taken at 187 Hz and continuous illumination). The artifacts such as skew are also reduced (compare Figure 1.2-2a with Figure 1.2-3a). Figure 1.2 shows frames taken with different camera and illumination setups using a Sony PlayStation Eye camera (Sony, 2014). The Figure 1.2-1a presents both blur and skew, and is the common setup used by low cost ETs. In Figure 1.2-1b, the blur is reduced due to the low exposure time, but as the lines are shifted in time, the pupil and iris appear deformed. In Figure 1.2-3a, we see the stroboscopic light result. Despite the 33 ms integration time, the image is sharp because the effective illumination took only 800 μs. The quality is comparable to a capture done at 1250 Hz. As expected, the image quality is superior as the frame taken at 187 Hz with continuous illumination (Figure 1.2-1b).

To produce images as in Figure 1.2-3a, the pulse of light, also referred as strobe, must be triggered on the right instant. While it is a trivial task for high grade cameras (such as the Blackfly series from PointGrey (Point Grey, 2015)), which provide a synchronization output, it is not as

easy for regular webcams. Interesting though, is that most camera chips provide the output (see for example sensor OV7725 from OmniVision (OmniVision Technologies, Inc., 2007)), which is not made available to customers of low cost devices by the camera manufacturers.

In this work we synchronize the light source with consumer-grade cameras using efficient image processing algorithms, without relying on particular camera interfaces. A separate circuit is responsible for triggering the lights alternately. By carefully adjusting the camera exposure and strobe length, a visible feature can be tracked on the frame images. The feature takes a form of a dark stripe for easy detection. By allowing the circuit to communicate with the computer, the pulse periods can be adjusted based on the stripe translation until synchronization with the camera is achieved.

1.4 Contributions

The *differential lighting* technique allows the implementation of lightweight pupil detectors but is particularly difficult to use on modern, low cost cameras, which are based on the rolling shutter mechanism. In this context, a key contribution of this work is a software-based synchronization means, which coupled with an external stroboscopic light controller is able to provide high quality images from off-the-shelf hardware. The technique has the potential to adjust the camera parameters automatically, such as to take full advantage of each frame. We show the technique can work well in a typical eye tracking experiment and is computationally lighter than other dark pupil based techniques.

Another key contribution is the possibility to exploit the DL without explicit light synchronization, greatly simplifying the external hardware implementation and setup, as the illuminator is decoupled from the system. This allows the temporal requirements of the illumination to be achieved even with analog components, which might enable more people to exploit our findings. A typical eye tracking experiment suggests the performance of the asynchronous technique is similar to the software-synchronized counterpart.

A new pupil refinement algorithm is also proposed. The candidate points to the pupil contour are computed with sub pixel resolution, and outliers removed without the need to fit an ellipse. The method iteratively finds the thresholds for segmenting the pupil and CR when necessary, an important feature as the bright pupil effect is idiosyncratic and varies with the gaze angle. The temporal support was also increased, instead of using two adjacent frames to detect the pupil, a larger distance is allowed. With such, computations needed for detection and refinement were reduced.

A particularity of the techniques proposed is that some frame lines might not get illuminated, result of the pulsed nature of the light and particular combinations of camera parameters and timings. The pulse length is one of the factors which when reduced, contribute to minimize the lines compromised. Because the reduction in pulse length is accompanied by increase in power, we must carefully consider eye safety. Issues related to the eye exposure to NIR are thoroughly considered.

1.5 Organization

The remainder of this thesis is organized as follows:

Chapter 2 shows how the DL technique can be extended to work with rolling shutter cameras. Particularly, it describes the method developed to synchronize an external strobe generator to the camera frame rate, such that frames are even illuminated, without artifacts and with improved image quality. Chapter 3 relax the assumptions of the method. Notably, the strobe is allowed to run independently of the camera, allowing the construction of simpler timing circuits. As a drawback, the pulse length of the stroboscopic light must be reduced. Appendix C describes how this can be accomplished and possible controller embodiments are presented in Appendix D. Chapter 4 describes how the result of the image differencing can be refined to obtain an accurate pupil contour. Chapter 5 describes the eye tracking experiments performed to assess the technique and compare to state-of-the-art counterparts. The prototypes employed are described in Appendix E. Chapter 6 presents the results of the eye tracking experiments. Additional results which involve the presentation of multiple images, such as the whole tracking sequence of a saccade, are presented in Appendix A. Conclusions are drawn in Chapter 7. Additionally, the limitations of the proposed techniques are discussed, as well as insights on future directions the development might take.

The techniques described are evaluated with different prototypes. They can raise concerns about the safety to the eyes due to different reasons. While one employs powerful *light-emitting diodes* (LEDs) to compensate for many losses on the light path due to particularities on the construction, a second prototype employs powerful LEDs to achieve extremely short pulses. Before use, they were assessed using the latest standard on the subject. Additionally, we have considered the *International Commission on Non-Ionizing Radiation Protection* (ICNIRP) guidelines on eye safety as well. A summary of the eye safety standards and guidelines is presented in Appendix B.

Chapter 2

Differential stroboscopic lighting

2.1 Introduction

Differential lighting (DL) synchronizes the two light sources with the camera frames to generate alternate bright and dark pupil images (Ebisawa, 1998; Morimoto *et al.*, 2000). From the difference of two consecutive frames (Figure 1.1c), high contrast regions are detected as pupil candidates as seen in Figure 1.1d. The detected region in the difference image corresponds to the overlap between the dark and bright pupils, and require further refinement to track the real pupil contour, as suggested by Hennessey *et al.* (2006).

In Morimoto *et al.* (2000), the light controller is connected directly to the camera which provides the synchronization signal to trigger the lights at the right instant. Figure 2.1 depicts such arrangement. The limitation however is that most consumer cameras do not provide such output. This is particularly true for webcams designed for computer usage. As this work targets off-the-shelf components to allow affordable designs, this limitation must be overcome with alternative approaches.

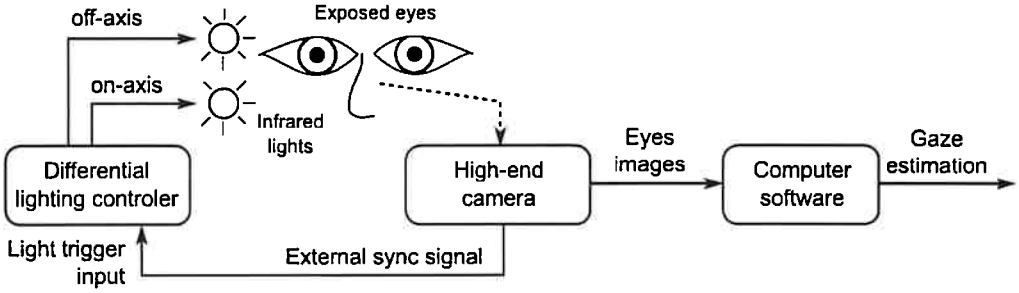


Figure 2.1: *Traditional DL component arrangement. The camera has a synchronization output which can be used to trigger the illuminators at the right time.*

To allow the DL technique to be used with low cost cameras, we propose a software-based synchronization method. We developed computer vision algorithms able to compute the lag between the camera frames and the activation of the structured illumination. With the help of an external circuit, we are able to control the moment the lights are fired, such that synchronization is achieved. As a result, we are able to obtain clear pupil images with alternate colors at frame rate.

The remaining of the chapter is organized as follows. The next section (2.2) introduces the problem of using low-cost cameras to eye tracking. Section 2.3 describes the method employed to keep external pulsed light sources synchronized with low cost cameras. Section 2.4 describes the

image differencing which allows the pupil to be robustly detect. Section 2.5 gives implementation details of the technique. Section 2.6 describes the experiments performed and presents the results. Section 2.7 closes the chapter with a discussion on the results.

2.2 Problem statement

One problem that reduces eye tracking accuracy is the lack of consistent light (Holland and Komogortsev, 2012). In dim lit environments, cameras typically compensate poor illumination by increasing the exposure and gain. The former increases motion blur of fast moving objects while the latter increases noise (Figure 1.2-1a shows blurring). In bright places, the exposure can be reduced as well as the gain. This results in better images, with low noise, and less motion blur. Another problem caused by the use of rolling shutters, which is employed by the majority of CMOS video cameras today, is the artifacts such as skew and smear, that affects the image quality around fast moving objects such as the eye (Figure 1.2-2a shows skew on the pupil and iris).

The use of structured lighting to locate the eyes improves the illumination but poses further challenges to image processing using rolling shutter cameras since the exposure of one frame might include the illumination from different lights. This problem is only aggravated with the use of low cost high speed cameras, so that switching lights on and off with a frequency close to the frame rate of the camera becomes impractical. To overcome the limitations of rolling shutter cameras, Theobalt *et al.* (2004) have used stroboscopic lighting to capture high speed motion of a baseball using standard still cameras and high power stroboscope, and Bradley *et al.* (2009) have used stroboscopic light to synchronize an array of consumer grade cameras. Kim *et al.* (2014) used a infrared *light-emitting diode* (LED) tied to a simple independent timer to synchronize eye and scene image streams manually.

2.3 Camera software synchronization

Regular cameras do not have a synchronization output. Without that, it is hard to get structured light to work properly, as the majority of the frames will suffer from double exposure. A frame suffering from double exposure is characterized by lines illuminated by different light sources. To keep the structured light synchronized with a regular camera, we have developed computer vision algorithms that adjust the firing of the illuminators independently of the camera frame rate, that minimizes double exposure in the center of the image. Figure 2.2 depicts the proposed solution. The synchronization link between the camera and the lights (Figure 2.2a) is replaced by a link to the host computer (Figure 2.2b).

2.3.1 Camera model

We adopt a rolling shutter camera model similar to the one of Bradley *et al.* (2009). Figure 2.3 illustrates the capturing process within this model. Observe that if a vertical line is drawn perpendicular to the time axis, it may cross two integration areas from different frames (it will certainly happen if the exposure is set to maximum). Each pixel exposure starts after its contents are cleared and ends with the reading. The slope of the clear and read lines represent the shear of the exposure intervals along the time axis. The slope is a function of the frame period Δt and

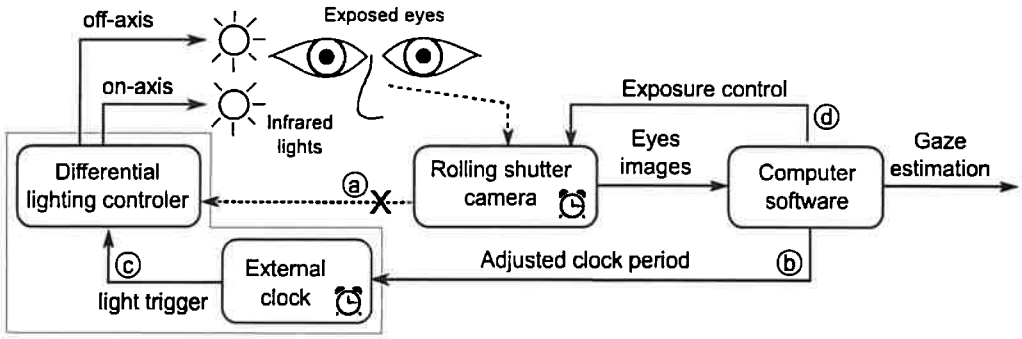


Figure 2.2: *Physical setup of the proposed system. Link (a) is the hardware sync, while links (b) and (c) correspond to the software controlled alternative. Link (d) is responsible for the exposure automatic adjustments.*

total number of scanlines S. Just before actual pixel data is read out, the sensor works on optical black and dummy pixels for D_0 scanline periods, in a so called vertical blanking time. After the blanking time, N valid scanlines are output, followed by D_Z dummy scanlines at the end of the frame. This is a general model compatible with most camera chips currently available from several vendors (OmniVision Technologies, Inc., 2007; ON Semiconductor, 2014) and image sensor receivers (Texas Instruments Incorporated, 2015). Note that D_Z can be zero without loss of generality. We will adopt the term scanline to refer to sensor lines, which include the invisible ones, and the term line when referring to the visible ones, which form the actual frame.

To obtain even exposed frames using pulsed light, the lights must be well synchronized. Let t_{line} be the time taken to read a scanline, which is given by $\Delta t/S$. Let Δ_{strobe} be the on period of the stroboscopic light. The value of Δ_{strobe} must be

$$\Delta_{strobe} \leq \Delta e - (N \cdot t_{line}) \tag{2.1}$$

with Δe in the open interval $(N \cdot t_{line}, \Delta t)$.

Consider the start of a frame at t_0 and t_{strobe} as the onset of the illumination. Given Δ_{strobe}, t_{strobe} is bound to

$$t_0 \leq t_{strobe} \leq t_0 + (D_0 \cdot t_{line}) - \Delta_{strobe} \tag{2.2}$$

for $\Delta_{strobe} \leq (D_0 \cdot t_{line})$. Otherwise, t_{strobe} must be triggered before t_0.

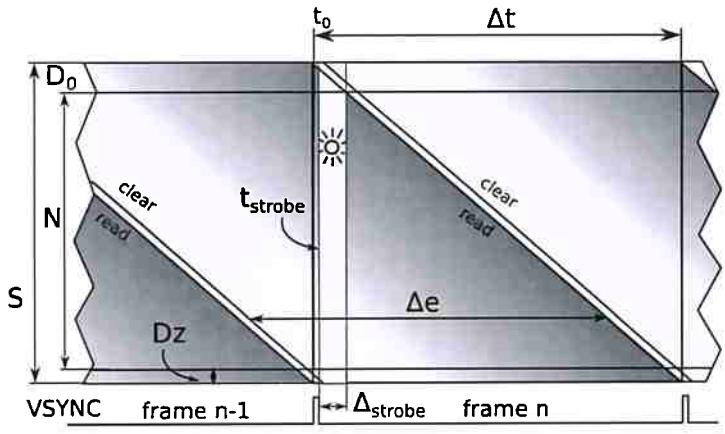


Figure 2.3: *Producing an even frame illumination with stroboscopic light in a rolling shutter camera.*

When the hardware synchronization output is available, avoiding double exposure is a matter of limiting the maximum Δ_{strobe} and the minimum exposure time (Δe), such that both (2.1) and (2.2) hold. As the stroboscopic light is triggered by the synchronization output of the camera, in general, t_{strobe} can be configured to be very small.

Nevertheless, to obtain frames without double exposure by means of software synching, the camera exposure must be manipulated to generate particular image patterns (including double exposure), which with the help of computer vision algorithms, enable the estimation of the lag between the camera clock (t_0) and the clock used to generate the light pulses (t_{strobe}).

2.3.2 Synchronizing the camera and stroboscopic lighting

The stroboscopic lighting is assumed to be driven by an independent clock with period Δ_{clke} that can be manipulated by software. Despite having control over the period, the difference in time (here named as lag), between a strobe and the start of a frame from the camera depends solely on their clocks. Therefore, to match them, one must be temporarily manipulated until the synchrony is achieved.

Thus, two independent steps are necessary to synchronize the camera with the lights: 1) adjust Δ_{clke} in order to match the camera period (Δt); 2) manipulate Δ_{clke} temporarily to induce a drift between the clocks such that the lights pulse approximately at the start of a frame (i.e. $t_0 \approx t_{strobe}$). Both are done by computationally processing frame images.

If t_0 is not matched to t_{strobe}, the produced image might contain lines illuminated by different sources. Figure 2.4 shows a specific timing diagram where the stroboscopic light causes double exposure, illustrated by the eye image in the same figure. The image is composed by scanlines from two different exposures of the strobe light, fired approximately Δt apart from each other. Region **A** shows the dark pupil effect while region **D** shows the bright one. The dark stripe in the middle of the image is the result of less stroboscopic light reaching the sensor during those scanlines.

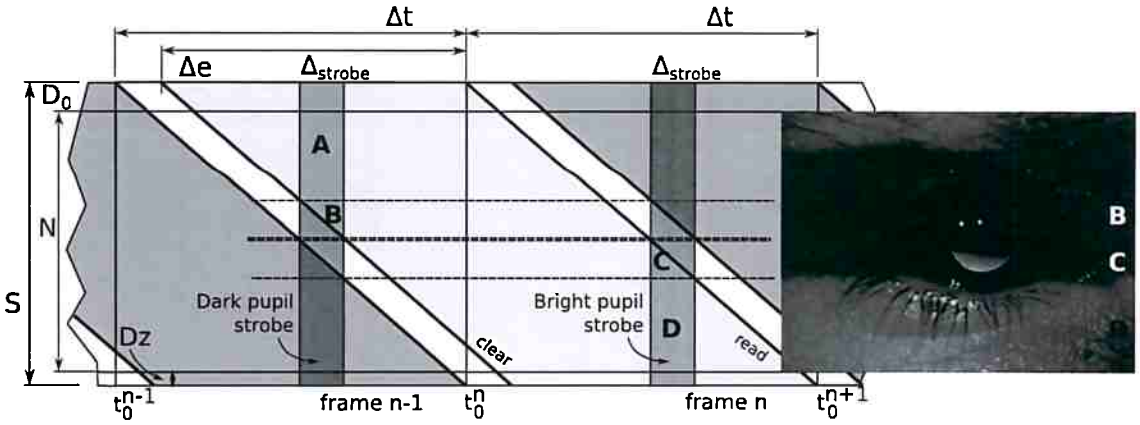


Figure 2.4: *Rolling shutter camera with a delayed strobe and actual frame capture presenting double exposure.*

The position of the stripe on image indicates how much one clock is delayed with respect to the other. This is what we call the lag between the clocks at a given time. By observing the stripe at different instants, the difference on the clock periods can be estimated. How much the stripe moves in a given time is a measure of the drift between the clocks. Then the external clock can be adjusted until its period converges to the one of the camera.

2.3.2.1 Estimating the lag between the clocks

To estimate the lag between the clocks, the position of the stripe in the frame image is necessary. The stripe can be defined by its height, h_{stripe}, and position D_{stripe}. By manipulating the camera exposure (Δe), the frame illumination is changed and accordingly, the height of the stripe.

To easy the detection, the exposure is set such as to maximize both the frame illumination coverage and the gradients produced, as follows

$$\Delta e = \Delta t - (\Delta_{strobe} + t_{line}). \tag{2.3}$$

With such exposure, stripes with exactly one scanline not illuminated by any strobes are produced. The stripe height (h_{stripe}) is given by

$$h_{stripe} = \frac{2\Delta_{strobe}}{t_{line}} + 1. \tag{2.4}$$

A smaller exposure results in more scanlines not illuminated, consequently, reducing the useful image area. A larger value reduce the stripe height and increase its brightness, which might result in a difficult detection.

The lag between the camera clock t_0 and t_{strobe} can be expressed as follows:

$$\Delta_{lag} \approx (t_{line} \cdot (D_0 + D_{stripe})) \tag{2.5}$$

where D_{stripe} is the line to receive no light from strobes. Its estimation is described in Section 2.5.1. The value D_0 is needed, but typically, only the value of $D_0 + D_z$ which is equal to $S - N$ is available. In such case, a good approximation is to take $D_0 = (S - N)/2$.

Note (2.3) gives the exact value described only if $\Delta t = \Delta_{clke}$. As the camera period (Δt) is empirically estimated, it will hardly be the case. The method described in Section 2.5.1 is robust to deal with the resulting differences however. In (2.5), Δ_{lag} is an approximation even when $\Delta t = \Delta_{clke}$, as in practice D_{stripe} might start within a scanline.

2.3.2.2 Strobe triggering adjustment (drift minimization)

The difference between Δt and Δ_{clke} causes the stripe to move over time. The movement indicates a drift between the clocks which causes Δ_{lag} to change over time. By minimizing the drift, we approximate Δ_{clke} to Δt. We propose to tackle the problem of minimizing the drift using variable steps. An initial estimation of the difference on the clock periods is done and this value is used as bootstrap value for Δ_{clke}. After the external clock is adjusted, a new estimation of the difference is performed. This process is performed until the clocks converge.

The idea is to adjust Δ_{clke} until the stripe moves less than a determined amount in a given time. This happens when $\Delta t \approx \Delta clke$. Due to the physical factors such as supply variations and different temperature coefficients, the perfect match is impractical. The difference between the clock of the camera and the external one being controlled will be called Δ_{drift}. Its estimation can be done as follows

$$\Delta_{drift} = \Delta t \cdot \frac{\Delta_{lag}(t_n) - \Delta_{lag}(t_{n-1})}{\Delta T} \tag{2.6}$$

where $\Delta T = t_n - t_{n-1}$, with t_n and t_{n-1} representing two moments corresponding to the delivery of a frame to the computer (not necessarily adjacent), $\Delta_{lag}(t_n)$ is (2.5) calculated for the frame at t_n. Note that for the purpose of calculating Δ_{drift}, the value of D_0 can be ignored in (2.5), as it will be canceled in (2.6).

If (2.6) returns a negative value, the external clock period Δ_{clke} is lower than the one of the camera, and the stripe is moving upwards. If otherwise, it is positive, Δ_{clke} is higher and the stripe is moving downwards. The external clock is then adjusted using the difference found

$$\Delta_{clke}(t_{n+1}) = \Delta_{clke}(t_n) - \Delta_{drift} \tag{2.7}$$

where $\Delta_{clke}(t)$ represents the period of the external clock at a given time t.

The algorithm converges quickly. Initially, the out of synchrony might be big, and the value of ΔT is selected to be low (equivalent to a frame period Δt). If in that period, the stripe did not moved, ΔT is increased in Δt increments.

The stop criterion employed is to wait the equivalent in time to 200 frames (a little more than $1.0s$ at $187Hz$) for a scanline drift, if it not occurs, the algorithm is stopped and the clocks are assumed to be synchronized.

The estimation of the drift depends on the correct detection of the stripe. The stripe can only be detected correctly if the whole profile is on the visible scanline range. When the stripe is partially hidden, the period of the external clock Δ_{clke} is temporarily biased to allow the stripe to quickly move to the visible scanlines. The value added to Δ_{clke} depends on the sign of the last calculated Δ_{drift}, the sensor scanlines, the stripe height, and the frame rate, as follows

$$bias = \Delta t \cdot t_{line} \cdot [(S - N) + h_{stripe}] \cdot \xi^{-1} \tag{2.8}$$

where $|\xi|$ is the desired time to move the stripe to the visible scanline range, being positive if the last calculated Δ_{drift} was positive, and negative otherwise. After $|\xi|$ seconds from adding *bias* to Δ_{clke}, the stripe is on the visible scanline range and Δ_{clke} can be recovered to the old value and the adjustment routine resumed.

After the clocks are on synchrony, the stripe must be moved to the invisible scanline range. This is done by adding a bias to the external clock for a user defined period of $\dot{\xi}$ seconds, as follows

$$bias = \Delta t \cdot [(t_{line} \cdot D_{stripe}) - \Delta t + \Delta e] \cdot \dot{\xi}^{-1} \tag{2.9}$$

where D_{stripe} is defined as in (2.5).

Once the external clock is matching the camera and ticks approximately at t_0, the images can be used with minimum artifacts. Due to the limited clock resolution ($62.5ns$) and different temperature coefficients, some drift of the stripe is expected however. To avoid any frame loss, the system must check, from time to time, the position of the stripe and if necessary, temporarily increase or decrease the external clock period (Δ_{clke}) as needed.

2.4 Image differencing

With the clocks synchronized and the stripe close to the frame border, the difference between consecutive frames can be used to detect pupil candidates. This procedure is also called pupil

candidate estimation, with the best candidate further refined as defined in Chapter 4.

When no previous information is available, the contour of the pupil overlap is estimated using an adaptive thresholding technique using the whole frame. An inverted cumulative histogram is calculated to help the algorithm on converging faster. A minimum pupil size (p_{min}) is used to bootstrap the algorithm with the initial threshold using (2.11).

The inverted cumulative histogram is defined as follows:

$$\mathbb{H}_i = \sum_{j=i}^{k-1} h_j, \quad i = 0..k-1 \tag{2.10}$$

where both i and j are bin numbers, h denotes the histogram, h_j is the number of pixels that falls into the intensity interval defined by bin j and, k is the total number of bins.

The number of bins used is equal to the quantization ($k = 256$). The first threshold level is defined by

$$\text{threshold} = \underset{i}{arg\,min} |\mathbb{H}_i - A| \tag{2.11}$$

where $A = \pi \cdot p_{min}^2$.

To speed up the convergence, the algorithm tries to reduce the threshold in large steps. A new threshold is found as the intensity which provides enough pixels to fill up an ellipse that encloses the contour of the largest area traced from the binarized image on the current iteration.

The ratio between the area of the largest candidate and the sum of the area of the other candidates is used as stop criterion. This is similar to the criterion used in Starburst (Li and Parkhurst, 2005) to detect the corneal reflection. Initially, it is expected that only the true pupil overlap and some other small regions are segmented. The ratio initially increases as the threshold is lowered, as the area of the most prominent candidate grows faster. The algorithm stops when the ratio drops for the first time. Intuitively, this happens when the candidate corresponding to the pupil overlap grows slower than the other candidates. A list of candidates corresponding to regions segmented in the iteration immediately before the drop are returned for additional checks. Note that the ratio is only calculated when more than one disjoint region is found, which can happen in one iteration and do not in the next.

Eventually, the computer might skip a frame, leading to a condition in which both the actual and the previous frame correspond to the same pupil color (i.e. dark or bright). In this case, the resulting difference image will be basically noise. Running the method just described will result in lots of candidates which, if no action is taken, will eventually grow to a single region. To stop such cases the earliest, there are additional parameters, such as the maximum candidate count and maximum pupil radius.

The adaptive threshold returns a list of candidate regions. Though it is very likely that the candidate with the largest area corresponds to the pupil overlap, additional tests are done, such as checking the ratio between the major and minor axis of an enclosing ellipse. A candidate might also be dismissed if it is far from the last calculated pupil position, if available. Sanity checks are also done to eliminate bordering regions. The candidate closest to the last pupil position is considered as representative of the current pupil overlap.

Running an iterative algorithm which includes several thresholding and contour tracing for every

frame, for the whole frame, is computationally expensive. Thus the candidate estimation is done only for a *region of interest* (ROI) defined using the radius and position of the last valid pupil, if available. The ROI is defined as a square region centered on the last pupil center and side $\aleph$ times the pupil radius. We also exploit the processing already needed for Section 4.2 which deals with the pupil contour refinement, to calculate the average intensities of the pupil and iris from the traced rays. The additional processing is negligible and provides an image-aware threshold based on the most recent frames. At high frame rates, the change in luminance from one frame to another is considerably small, thus making it feasible to use a threshold based on such.

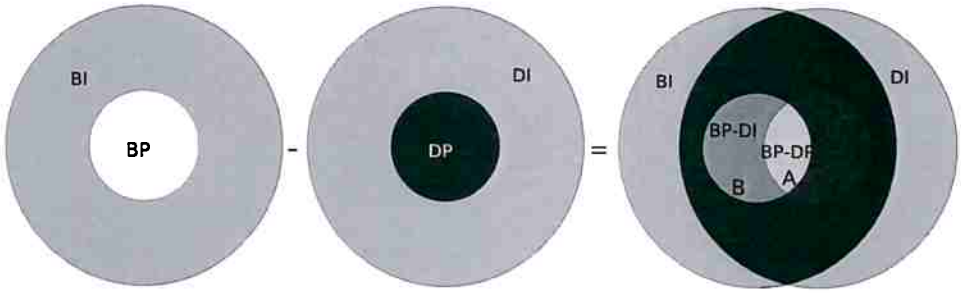


Figure 2.5: *Diagram showing the pupil (P) and iris (I) for a bright (B) and dark (D) pupil frame and the resulting difference. In blue are regions A, B and C, corresponding to the intersection of the bright with the dark pupil, bright pupil with the dark iris, and bright iris with the dark pupil, respectively.*

Considering Figure 2.5 and the regions A, B and C it defines, considering also that the average intensity at region X is denoted by $\overline{X}$. Let $\overline{BI} \approx \overline{DI}$, if it holds that $\overline{A} > \overline{B}$ and $\overline{A} > \overline{C}$, a good choice for the threshold to detect region A from the difference image is as follows:

$$\text{threshold} = \frac{\overline{A} + max(\overline{B}, \overline{C})}{2} \tag{2.12}$$

When available, the algorithm starts with the threshold in (2.12) and runs only one iteration. This will be the case most of the time, except when the tracking is lost and shortly after, such as during a blink.

2.5 Implementation

A system based on the *differential stroboscopic lighting* (DSL) technique is composed by a camera, a structured light source, a controller to trigger the lights and a computer. Instead of connecting the lights directly to the camera, a separate computer-configurable controller is used (see Figure 2.2 for details).

The experiments described in this chapter employed the prototype described in Section E.1. As keeping the synchronization becomes harder as the frame rate is increased, the camera employed was the PS3 Eye operating at 187 Hz. A 16-bit timer of an Arduino Leonardo board was used to trigger the external clock. The timer was configured without prescaling, giving a resolution of 1/16 MHz (62.5 ns).

The next section describes how the stripe position is determined using lightweight image processing algorithms. The images used were collected using the prototype described in Section E.1.

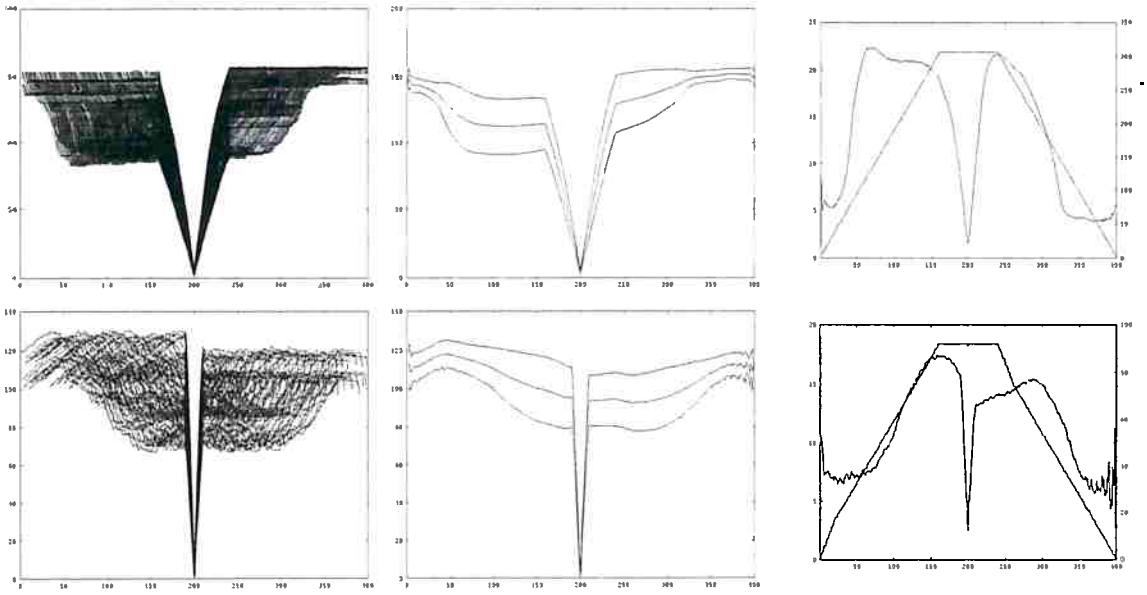


Figure 2.6: *Stripe alignment result using the line with minimum intensity. On top $\Delta_{strobe} = 800\mu s$ and the eye of USER2 is imaged, on bottom, USER1 is imaged with $\Delta_{strobe} = 200\mu s$. a) Average column image aligned by the minimum intensity. b) Mean $\pm SD$ of pixel intensities c) Number of images considered in each position and SD of the pixel intensity.*

2.5.1 Stripe detection

The stripe is very likely to be formed by gradients. Thus, its detection can be done by applying a gradient operator over the average column image, such as a *derivative of gaussian* (DoG), and pick the location of the strongest sign change in the convoluted image as the stripe central point.

Another approach is to simply select the darkest line on the frame. The intensity of the line will vary depending on the subject, the camera position, the ambient light, but it is very likely to be the darkest on the frame, particularly for the eye tracking task, in which the depth variation is small.

Both methods were tested offline using 893 frames from USER1 and USER2 (Section 5.3.1). For each frame, the average column was found and the images aligned by the line with minimum intensity. In order to keep the whole stripe, images in which the minimum were closer to half the stripe height from bottom or top were discarded.

Figure 2.6 shows the alignment result. Note how the standard deviation is progressively smaller when it approaches the line with minimum intensity. Note also that the stripe is consistently darker than the rest of the frame. This characteristic is expected to be maintained for a range of ambient illuminations, as the intensity of a scanline is a function of the exposure and strobe length[1].

Figure 2.7 shows the convolution result and the placement of the minimum intensity scanline as calculated by taking the mean position between the strongest positive and negative responses of the filter. The kernel employed was an approximation of a DoG kernel of size 15. Note how the response is better for USER1, in which the strobe length is smaller, and thus, the spatial support of the filter is proportionally larger. For $\Delta_{strobe} = 800\mu s$, the stripe height is about 41 lines, while

[1]Particularly true for small depth variations, such as the eye tracking task, in which part of the face is imaged. Might not hold for other tasks.

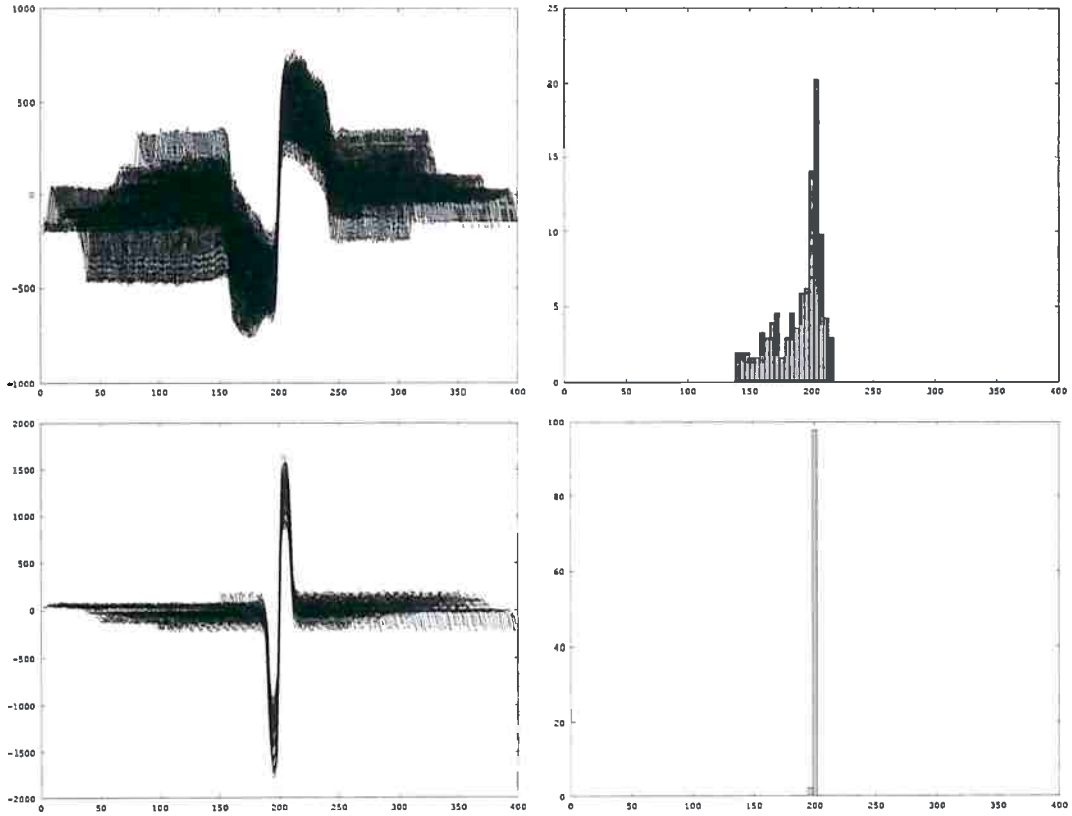


Figure 2.7: *Convolution of the column images from Figure 2.6 by an approximation of a DoG kernel of size 15. On top, images from USER2 and bottom, from USER1. a) Plot of the convolution for each image; b) Histogram of the calculated minimum intensity scanline using the peaks from convolution.*

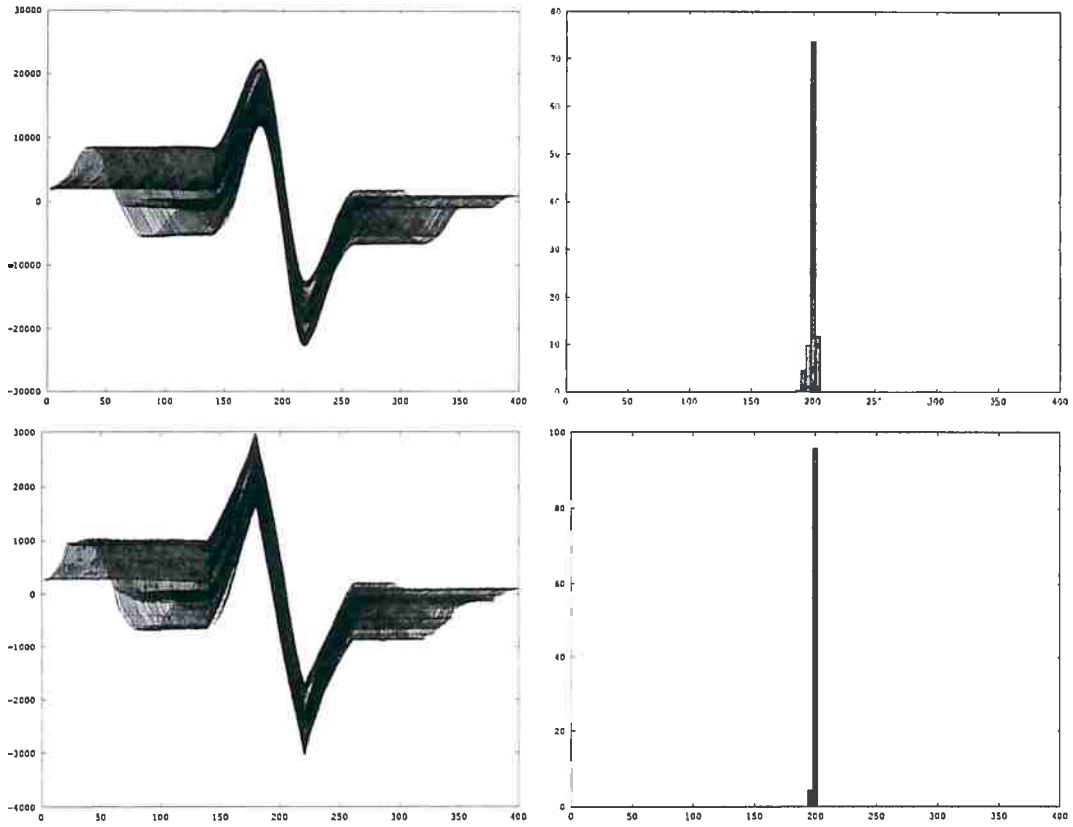


Figure 2.8: *Convolution of the column images from Figure 2.6a(top). On top, a kernel of size 41 defined as in (2.13) were used, while in bottom, only the two extreme values where kept nonzero. a) Plot of the convolution for each image of USER2; b) Histogram of the calculated minimum intensity scanline using the peaks from convolution.*

for $\Delta_{strobe} = 200\mu s$, the stripe is formed by approximated 10 lines.

Increasing the kernel support to the stripe height improve the results. Figure 2.8 shows the convolution with a kernel of size 41, equal the stripe height. The kernel employed is a first order function with unitary slope, defined as follows:

$$kernel(i) = i - \lfloor h_{stripe}/2 \rfloor \tag{2.13}$$

where $||kernel|| = 2 \cdot \lfloor h_{stripe}/2 \rfloor + 1$. The graphs at the bottom of Figure 2.8 were filtered using a kernel with only two nonzero values, the ones at the extremes. Note that a custom convolution can be used and is much faster than a traditional one using a smaller kernel.

By observing Figure 2.6a and comparing with Figure 2.7b and 2.8b, it looks advantageous to just take the scanline with minimum intensity from the average column image. However, when the stripe is hidden on the invisible scanlines, the minimum might not come from the stripe, thus, resulting in a wrong estimation. In such case, the accurate estimation of the stripe position is indeed not necessary, as the frame will not be used for the strobe triggering adjust (Section 2.3.2.2). However, being aware of such case is important to temporarily bias the clock and make the stripe move back to the visible scanlines.

The approach adopted here is to use both the convolution peaks and the line with minimum intensity. If the minimum is between the strongest peaks, the estimation is safe to be used as the

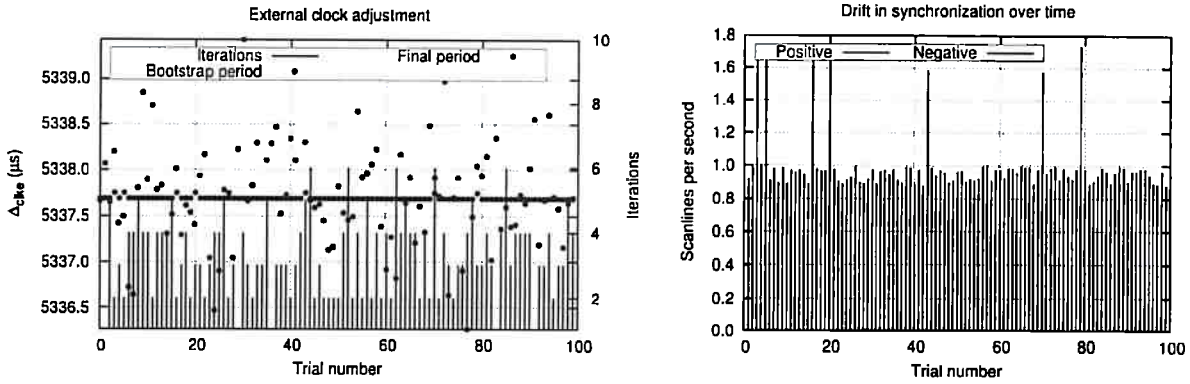


Figure 2.9: *Synchronization of the external clock on repeated trials for the PS3 Eye camera at 187Hz. a) The estimated camera period used as bootstrap for the iterative adjustment, the adjusted period, and the number of iterations necessary; b) The resulting drift in synchrony expressed in scanlines per second.*

true stripe location (D_{stripe}), if not, the stripe is assumed to be hidden. The convolution is done using a kernel of at least the size of the stripe, using values as defined in (2.13). The value of D_{stripe} is only considered valid if the peak locations of the convolution were compatible with the whole stripe being on the visible scanlines.

2.6 Experiment definition and results

Three volunteers were asked to wear the head mount apparatus described in Section E.1 while the procedure described in Section 2.3 were performed several times. Before each run, the external clock period was bootstrapped with a new estimation performed based on the delivery rate to the computer.

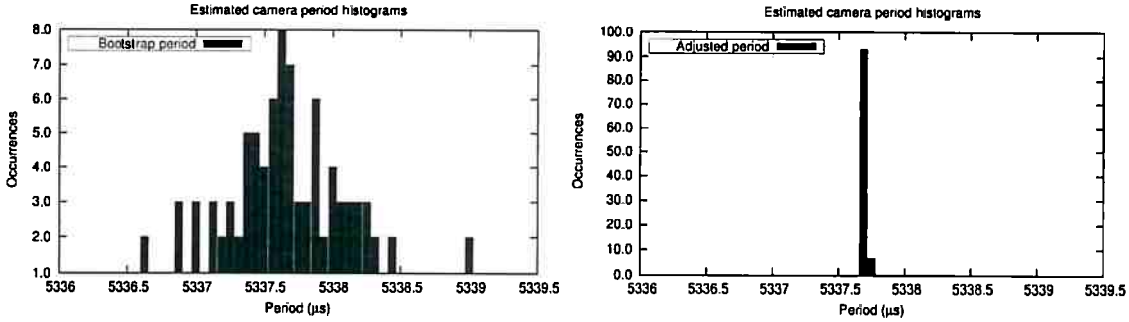
To assess the repeatability of the method, one hundred independent trials of the strobe synchronization were done. Figure 2.9 shows the results. As can be noted, the mostly found Δ_{clke}, which results in about one scanline per second of stripe drift, is the one that makes the camera most in synchrony with the strobe. A period just 62.5 ns longer, makes the stripe movement increase to more than 1.6, and the same amount shorter, makes it increase to 3.6 scanlines per second.

To assess the resulting synchronization, the movement of the stripe produced on the frames were measured on the long run. On each trial, after the synchronization was done, we waited for a stripe translation of at least 20 scanlines. The number of frames captured were then multiplied by the total scanlines and divided by 20 to obtain the data on Figure 2.9b.

In the actual implementation, the stripe detection takes only an average 405 μs (SD = 35.7 μs) of processing time. Note that with the results found, the stripe detection can be run in intervals of more than one second.

Figure 2.10 shows the histograms of the computer estimated camera periods and the periods estimated using the procedure from Section 2.3.

Figure 2.11 show images captured while a saccade is performed. The images correspond to volunteer USER1 (Section 5.3.1), and are sharp due to the effective small integration time of just 800 μs. Note a gradient to dark on the frame bottom, result of less strobe light impinging the respective lines. With the stop criterion defined in Section 2.3.2.2, the stripe will move by



a) b)

Figure 2.10: *a) Histogram of the computer estimated camera period, based on the delivery rate; b) Adjusted clock period.*

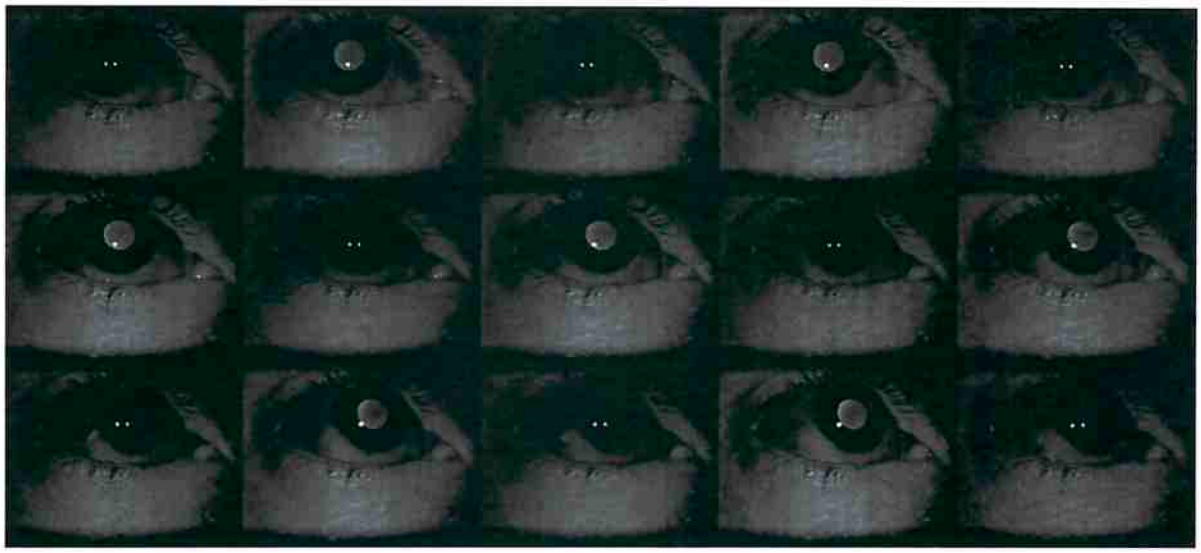


Figure 2.11: *Example capture of volunteer USER1 (Section 5.3.1) with a saccade ongoing, frames 784 to 798.*

approximately a line in a second.

2.7 Discussion

The original DL technique uses light sources synchronized with the video frames to create bright and dark pupil images. The method was originally developed for interlaced analog cameras and, therefore, was limited to 60 Hz. The DSL technique uses very short light pulses that will, most likely, be captured during only one frame. When the pulse is close to the beginning or end of the frame, our method detects a stripe that is used to delay the next pulse, i.e., no external synchrony is required from the camera. Thus, our technique can be used with any digital camera.

Another advantage of our technique is the reduction of motion blur during fast eye movements, which contributes to a higher accuracy. The effective exposure is given by the strobe length, which is typically small.

The effective processing needed to keep the synchronization by software is very small, and can be run only from time to time. The results suggest a drift between the clocks responsible for no more than two lines of stripe movement per second.

Chapter 3

Asynchronous differential stroboscopic lighting

3.1 Introduction

Previous chapter described a software-based solution to the synchronization of light pulses with off-the-shelf cameras. Such solution allowed the *differential lighting* (DL) technique to be used with cameras that do not provide a synchronization output, thus extending the potential usage to low-cost devices. The technique required a strobe controller capable of communicating with the system to allow adjustable triggering periods. From an image-based drift estimation, the necessary adjustments to keep synchrony over time could be accomplished.

A key issue with the software-based solution is the need to keep an open link with the light controller. Although many engineering solutions might be proposed to reduce the circuit footprint and possibly switch to a wireless link, the underlying complexity might be cumbersome. Maintenance can also be more complex, as there are more failure points.

In this chapter we introduce the *asynchronous differential stroboscopic lighting* (ADSL). The lights are allowed to pulse on their own, simplifying both the hardware and the software. The illuminator is completely decoupled from the computer system, being able to run independently. The computer software changes its role, from actively working to sustain the synchronization, to just keeping track of the asynchrony as it develops.

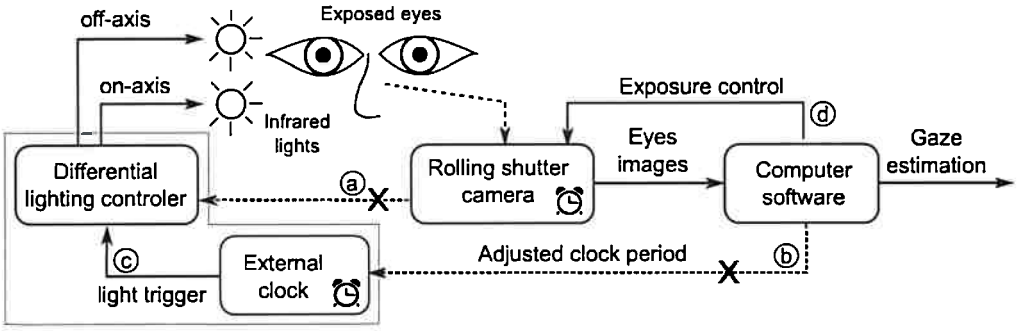


Figure 3.1: *Physical setup of the proposed system, which is an extension of the technique presented in the last chapter.*

Figure 3.1 depicts a system implementing the ADSL. Note that link (b) which allowed the

computer to control the clock was removed, and the differential lighting controller is completely decoupled.

The remainder of this chapter is organized as follows. The next section (3.2) introduces the problem of allowing the lighting to work independently from the camera. Section 3.3 describes the proposed method and defines minimum and desired operating conditions. Section 3.4 describes the method used to track the asynchrony such as to allow the image differencing to work (Section 3.5). Section 3.6 presents an system overview followed by Section 3.7, with experimental results. Finally, Section 3.8 concerns a discussion on current limitations of the method and an alternative way to deal with frames illuminated by two strobe firings.

3.2 Problem statement

Allowing the strobe to run independently of the camera, possibly at a different frequency, results in images with patterns that can range from partially lit frames, dark stripes, bright stripes and banding on the frame. Such patterns can take several minutes to cross the visible scanline range or just a couple of frames.

We are particularly interested in the formation models of bright, dark and invisible stripes. The conditions which lead to particular models are described in Section 3.3. For now, the stripe will be considered the smaller region on the sensor scanline space, which is delimited on both ends by fully lit scanlines. Additionally, each end must be illuminated by a different source of the structured lighting. In Figure 3.2, the stripe in *frame n* is given by the outer dotted lines.

If only a small difference in frequency is allowed, the time the stripe takes to cross the frame is limited and the stripe height can be readily controlled. The stripe, even if not visible, causes problems if over the pupil or *corneal reflections* (CRs), as it is composed, on the best case, by scanlines impinged by two different strobe lights. This results in a gradient in a dual-colored pupil and a mixture of CRs hard to predict.

As the stripe will eventually pass over the pupil (otherwise we have the synchronous case or our software-synchronous approach), the best option is to track its movement accurately, such that a special attention can be given to the affected area. We assume the stripe moves at constant velocity, which implies that the camera and external clocks maintain their differences over time.

In practice, a very small frequency drift is expected from both clocks. Temperature changes, supply fluctuations and on the long-run, component wear can cause such effect. The drift causes the stripe to violate the constant velocity hypothesis, making the position estimation more difficult and error-prone. In Section 3.8.1 we discuss in what extent capacitors and resistors used on *resistor-capacitor* (RC) timing circuits can influence the drift. If the external clock employs a crystal driven oscillator, the drift is expected to be much lower.

3.3 Asynchronous model

On software or hardware synchronized versions of the DL, a frame is characterized by a single pupil color (Ebisawa, 1998; Morimoto *et al.*, 2000). On the asynchronous technique, a frame might contain both pupil colors in one frame. Figure 3.2 shows the integration time of a frame (bright green area) in which two light pulses illuminated the scanlines (the orange and yellow ones). The frame

is the result of two snapshots taken by different strobes at different instants, which might result in a dual-colored pupil. As most of the frames will be illuminated by more than one light, we will introduce a convention to describe the pupil colors or illuminators active for a given frame. Starting at the top, a bd frame is a frame lit by the bright illuminator, followed by the dark illuminator on the rest of the frame. A db frame is the opposite.

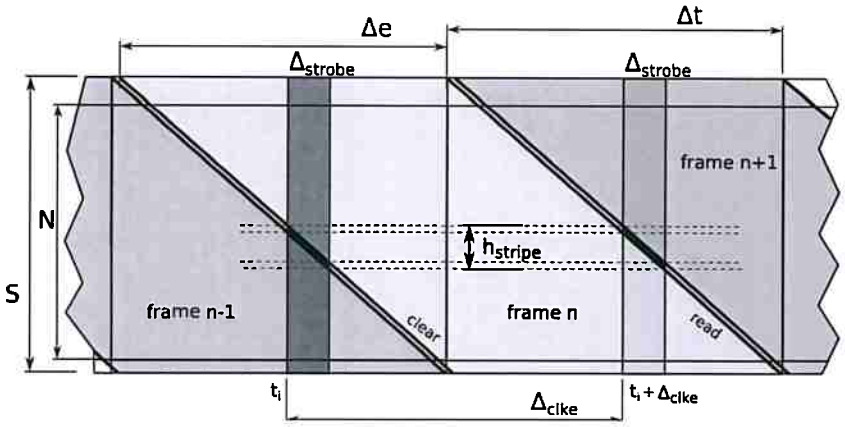


Figure 3.2: *Capture model using asynchronous stroboscopic light.*

On the long run, the time the pupil is covered by stripes is constant (assuming the pupil with a constant size (p_d) during the interval), and depends on the pupil size (p_d), stripe height (h_{stripe}) and total number of scanlines (S), as follows

$$D_c = \frac{2 \cdot h_{stripe} + p_d}{S} \tag{3.1}$$

where D_c is the proportion of time in which the pupil is covered. S, p_d and h_{stripe} are given in scanlines.

We will define the term stripe frequency (f_b), which is the inverse of the time the stripe takes to traverse all scanlines. The stripe frequency is given by the difference in frequency between the camera clock and the external clock. This happens because the stripe is suffering a phenomenon similar to the temporal aliasing. The sampling frequency (sample rate) is close to the signal (strobe) such that the resulting sample (the stripe in this context) represents the phase delay.

To be able to detect the stripe and estimate its movement, we limited the stripe movement such that at least three consecutive frames present the whole stripe, before it hides and appears again on the opposite side. Figure 3.3 depicts the maximum stripe movement. With only two we would not be able to estimate the direction of movement. The stripe profile must be entirely on the visible scanlines, thus,

$$n_{stripe} = \frac{S}{N} \cdot \frac{4 \cdot h_{stripe} - 1}{h_{stripe}} \tag{3.2}$$

where $4 \cdot h_{stripe} - 1$ means we must have almost four stripes on the visible scanlines, except for one scanline. This is to guarantee three whole stripes every any four adjacent frames. The term S/N scales the stripes to the invisible range. The allowed difference in frequency Δf, between the clocks is given by

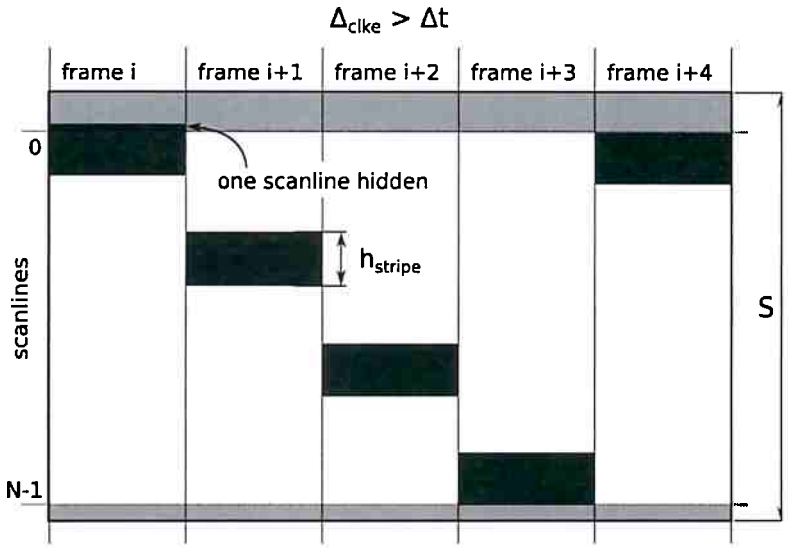


Figure 3.3: *Example of maximum stripe translation between frames to allow the estimation of the direction of movement. Shown are consecutive frame excerpts showing the stripe position.*

$$\Delta f = \frac{1}{(n_{stripe} - 1) \cdot \Delta t} \tag{3.3}$$

where Δt is the camera period in seconds.

Every time the stripe crosses the pupil, the pupil color sequence is inverted. If the stripe runs fast over the frame (large differences in clock, i.e. $S/n_{stripe} > h_{stripe}$), it may eventually be before the pupil in one frame, and after in the next. As a result, the pupil will have the same color in both. Such a condition impairs the detection based on image differencing.

The probability of the described condition becomes higher than zero only if $p_d \leq (S/n_{stripe} - h_{stripe})$. If this is the case, then the probability of two consecutive frames with the same pupil color is given by[1]

$$P(p_d|n_{stripe}) = \frac{1}{n_{stripe}} \cdot \left(1 - \frac{p_d}{(S/n_{stripe} - h_{stripe})}\right) \tag{3.4}$$

where $P(p_d|n_{stripe})$ gives the probability of a pupil with diameter p_d having the same color in consecutive frames given n_{stripe}.

For a sufficiently small pupil, this generates a situation where we simply cannot count on the detection, as it will fail eventually. Thus, we restrict the external clock range, such that $n_{stripe} \geq S/h_{stripe}$.

Assuming the exposure at maximum, i.e. $(\Delta t - \Delta e) \leq \epsilon$, with ϵ very close to zero (less than t_{line}) and considering $\Delta_{clke} = \Delta t$, the stripe have a height (h_{stripe}) dependent only on the total scanlines (S), the frame period (Δt) and the strobe length (Δ_{strobe}). Only in such case, the stripe height h_{stripe} is given by $(\Delta_{strobe} \cdot S)/\Delta t$. However, the frequency of the external clock must match the sample rate. As this is not the case, we must account the change in the stripe and adjust the range of the external clock.

If $|\Delta t - \Delta_{clke}| \neq 0$, the stripe height will increase linearly according to the magnitude of the difference. The stripe height is given by

[1]This is the equivalent to say that the stripe will be before the pupil in one frame, and after the pupil on the next.

$$h_{stripe} = \frac{\Delta_{strobe} \cdot S}{\Delta t} + \left| \frac{\Delta e - \Delta_{clke}}{t_{line}} \right| \tag{3.5}$$

If $\Delta_{clke} < \Delta t$, i.e., the external clock is at a higher frequency, the stripe will become brighter than the rest of the frame. It is simple to remedy however, if the frequency of the external clock is known. Just set the exposure length to be equal the external clock period (Figure 3.4e).

$$\Delta e = \Delta_{clke} \tag{3.6}$$

If $\Delta_{clke} > \Delta t$, i.e., the external clock is at a lower frequency, the stripe will become darker, no matter the exposure (the exposure length is limited to Δt), see Figure 3.4c. Thus, it is preferable to setup the external clock in a way to avoid frequencies lower than that of the camera.

Figure 3.4 shows the illumination impinged on the scanlines given the relation between Δ_{clke} and Δt. The clocks are assumed to have about 60% phase delay (or lag). The gray area in Figure 3.4b represents ϵ, which was exaggerated for visualization purposes. In orange and yellow, the two different strobes, responsible for dark and bright pupil effects. Note the frame is the result of two strobe firings, Δ_{clke} apart from each other (see Figure 3.2), which were put together to help on the visualization of the outcomes. In blue, the theoretical intensity of an average column image taken under the circumstances immediately to the left. In dark blue, scanlines lit by two strobe firings, with the arrow indicating the direction the stripe is moving over time.

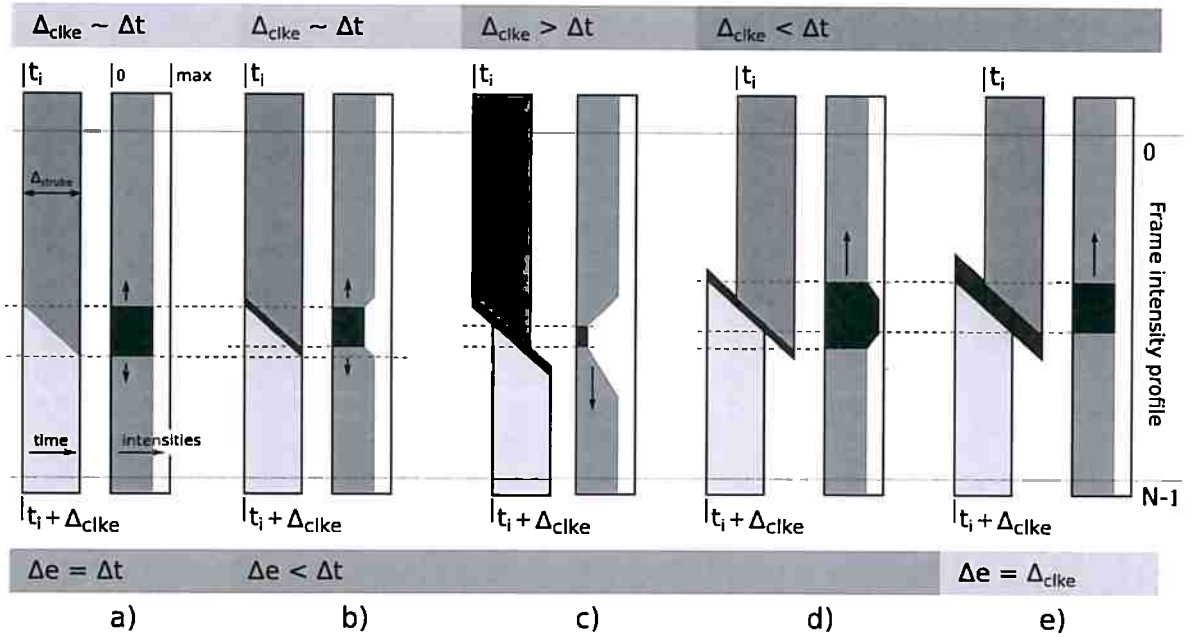


Figure 3.4: *Scanlines lit by stroboscopic light given the relation between Δ_{clke} and Δt, and corresponding frame intensity profiles. The yellow regions were translated by $-\Delta t$ to help on visualization.*

3.3.1 A 1×1 design

So far we have considered the minimum requirements for the external clock to avoid impairing conditions. Configuring the external clock with tight tolerances is not difficult however, even with analog circuits. Using a microcontroller makes the task even easier. Appendix D presents possible

implementations. In this section we will account the effect of different parameters on the produced image.

As already shown in (3.1), the time the pupil is covered by stripes is dependent on the stripe height. Suppose we start with $\Delta_{clke} = \Delta t$, and gradually decrease Δ_{clke}. The exposure length is adjusted to match Δ_{clke}, thus the stripe stays the same during the process. Using (3.1), we find the same pupil coverage for any Δ_{clke}. This is in fact true on the long run and assuming constant pupil size. However, the speed at which the stripe moves over the pupil changes, and plays a fundamental role on the detection.

The detection of pupil candidates is done by locating high contrasting areas between two consecutive frames. The bright and dark pupils form one of such regions. What we call a 1×1 design is the triggering of the bright pupil illumination followed by the dark one and so on, which is what is traditionally done. However, when the stripe is over the pupil, it becomes dual-colored. The triggering sequence can be described using regular expressions as (bd)*, producing frames following (db bd)*[2]. Faster the stripe moves from one frame to the next, greater the pupil's overlap with the same color, resulting in low to none contrast in such regions.

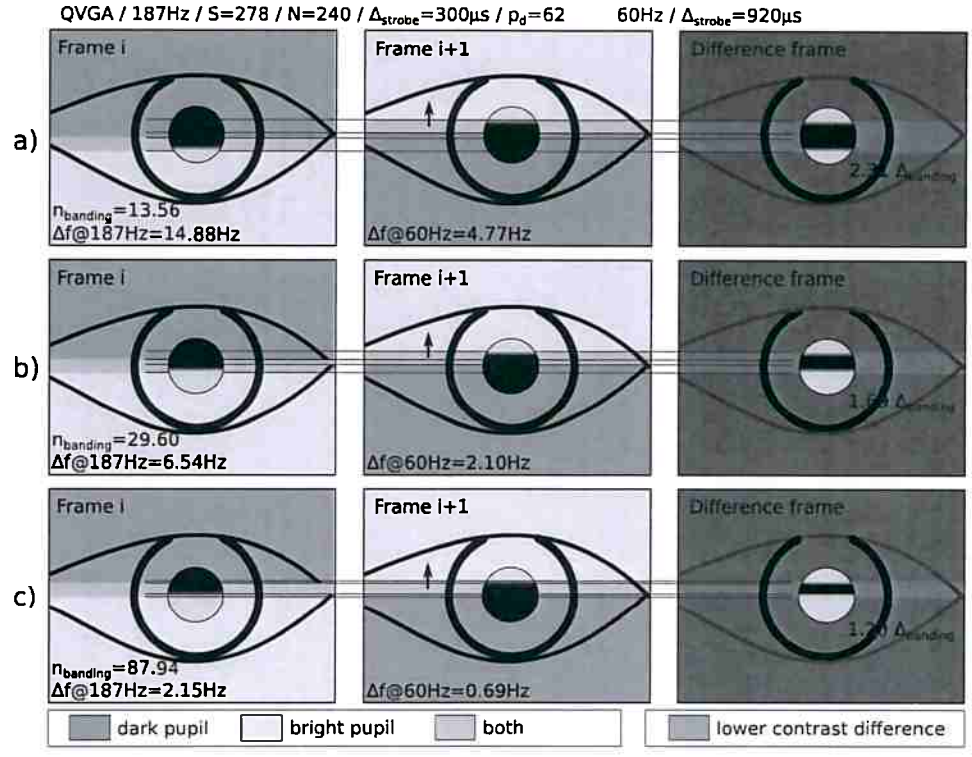


Figure 3.5: *The effect of asynchrony on the difference image.*

The number of scanlines affected by the stripe and its movement is given by

$$\hat{h}_{stripe} = h_{stripe} + \frac{|\Delta t - \Delta_{clke}|}{t_{line}} \tag{3.7}$$

where h_{stripe} is found using (3.5) and already incorporates the effect of different clock frequencies on the stripe height.

[2]Eventually, a bd frame might turn into a bb or dd frame when the stripe goes hidden, but those cases occur less frequently $(1 - N/S)$.

Figure 3.5 shows theoretical difference images under three external clock configurations. The picture was made on scale, such that the measure of the colored areas correspond to the values presented. In a), the external clock is at a higher frequency than the one of the camera, $\Delta_{clke} = 1/201.88s$ for a camera at $187Hz$ and $\Delta_{clke} = 1/64.77s$ for a camera at $60Hz$. The stripe takes only $n_{stripe} = 13.56$ frames to scan the whole frame extent. The length of the area affected on the difference image is equivalent to $2.31h_{stripe}$. In b) and c), the difference between the clocks is reduced (still higher in frequency), resulting in a reduction of the affected areas on the difference images.

From (3.7), we note the number of scanlines affected on the difference image is lower bounded by h_{stripe}. From (3.5), which defines h_{stripe}, the variables that can be changed are the exposure length (Δe), external clock period (Δ_{clke}) and strobe length (Δ_{strobe}). The first two are of little help on reducing the stripe height. Reducing the time the strobes are kept on imply increasing the power, which might not be possible for a couple of reasons. Section C discusses the problems related to implementing the shortest possible pulses.

3.3.2 A 2×2 design and higher

In a 2×2 design, the bright pupil illumination is triggered twice, followed by two activations of the dark pupil illuminators. The triggering sequence using regular expressions is (bbdd)*. The frame sequence will be then (db bb bd dd)*. The bb and dd frames will be clean if we use a higher frequency external clock (i.e. $\Delta_{clke} < \Delta t$) and adjust the exposure according to (3.6). The detection can be run on those frames only, and the refinement on all of them.

Figure 3.6 depicts four possible frames using the triggering sequence. The difference used to detect the pupil must now be done between frames $k - 2$ and k. Note that for frames bd and db with the pupil covered by the stripe, the scanlines affected on the difference image grows. In (3.7), the second term must be doubled to account the difference. Note however, that when the stripe is not over the pupil, frames i and $i + 2$ are very similar to frames $i + 1$ and $i + 3$, thus requiring the same computations.

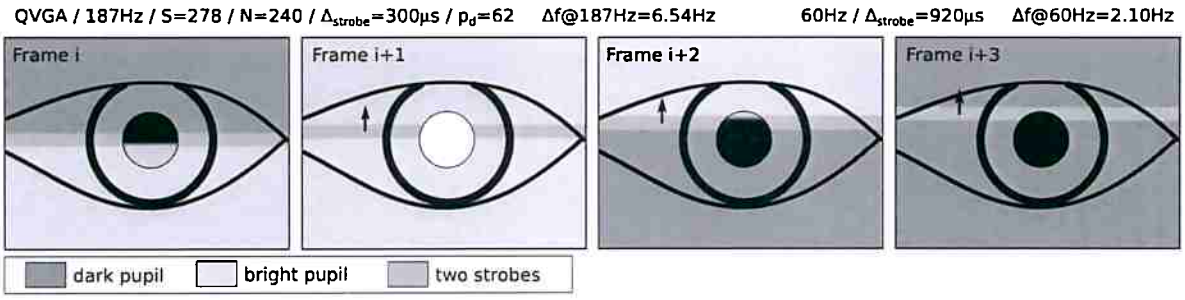


Figure 3.6: *Schematic frame sequence in a 2×2 design.*

A 3×3 design trigger the strobes this way (bbbddd)*. And the frame sequence becomes (db bb bb bd dd dd)*. The detection can be done between any bb and dd frame pairs. However, the differential method has lost power as the time window increased, and the pupils might have moved sufficiently for an empty intersection.

3.4 Implementation

A stripe like the one of Figure 3.5 might impair the eye tracking algorithm when over the pupil, as the affected scanlines have their pixel intensities zeroed or reduced on the difference image, the corneal reflections are distorted, and the pupil color mixed. As the clocks run independently, this will certainly happen, and with a rate given by (3.1). Thus, measures must be taken on each eye tracking step aforementioned. The effectiveness of the measures depend on the correct tracking of the stripe position though.

Taking the data from the previous chapter as example, in which a rectangle equivalent to $28 \times 21mm$ is covered in the field of view, the pupil size can take from 22.8 up to 80 pixels in diameter, for a constrained and fully dilated sample, respectively. The capture was done at $60Hz$ with a pulse length $\Delta_{strobe} = 800\mu s$. Considering the camera parameters, a stripe height of $h_{stripe} = 13.3$ scanlines is expected. Thus, using (3.1), we estimate from 17.7% to 38.3% of the total time with pupils partially compromised.

3.4.1 Estimating the total number of scanlines (S)

Section 3.3 assumed the total number of scanlines of the camera (S) to be known. Such number is essential to calculate the expected stripe size and the exposure. To set the exposure correctly, the external clock period (Δ_{clke}) must also be known in advance (3.6).

We propose finding S by simply estimating the time the stripe is hidden. As we can estimate the movement on the visible scanlines, we can use the speed and time to find the number of scanlines translated while the stripe was hidden. The idea is to record a long sequence of frames, enough to allow the band to pass the visible range several times. Then, estimate the stripe position in each frame, fitting several regressors of the form $\mathbb{B}(t) = a \cdot t + b$, where $\mathbb{B}(t)$ gives the stripe position given time t. The samples for a new linear regressor begin when the position drops while the time increased (or rises, depending on the stripe movement direction). Considering we have k regressors $\mathbb{B}_i(t) = a_i \cdot t + b_i, i = 1..k$, with $i = 1$ being on the earliest samples, we can find $\hat{S}$, the total scanlines weighted by the frequency difference (Δf) as follows

$$\hat{S} = \frac{1}{|\Pi|} \sum_{i \in \Pi} \frac{a_i \cdot b_j - a_j \cdot b_i}{a_j \cdot (i - j)}, \quad j \in \Pi, i > j \tag{3.8}$$

where $\Pi = \{t \in \mathbb{N} | 1 < t \leq k\}$ and j is the greatest index smaller than i in the set. Outliers, such as diverging angular coefficient regressors, can be filtered by excluding the index from the set Π. Figure 3.7 depicts the calculation process.

The total scanlines is calculated as follows

$$S = \hat{S} \cdot \hat{\Delta} f_{/} \tag{3.9}$$

where $\hat{\Delta} f_{/}$ is an estimate of the ratio between the camera clock and the external one. Do not confound $\Delta f_{/}$ with the previously described Δf in (3.3), as the first is a ratio and the second is the absolute difference in Hz. $\Delta f_{/}$ can be estimated as follows

$$\hat{\Delta} f_{/} = \left(\frac{1/f_b \cdot 1/\Delta t}{1 + (1/f_b \cdot 1/\Delta t)} \right)^{\tau} \tag{3.10}$$

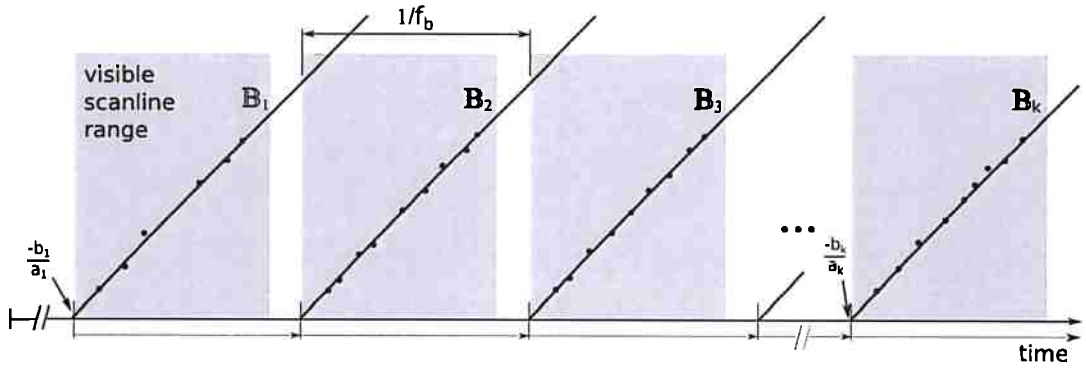


Figure 3.7: *Estimating S with minimal information.*

where $1/\Delta t$ is the sampling rate of the camera, r is 1 for positive $\mathbb{B}$ slopes and -1 otherwise, and $1/f_b$ is the period the stripe takes to travel all the scanlines (in seconds). The value of $1/f_b$ can be found using the regressors by simply replacing a_j by $a_i \cdot a_j$ in (3.8) denominator.

We assume the exposure is adjusted in order to produce a stripe composed by three distinct regions, a gradient to dark, a dark region which can be one scanline wide, and a gradient from dark. We also assume the darkest scanlines belong to the non-illuminated region. The stripe position (D_{stripe}) is then given by the position of minimum intensity in a mean column image.

Adjusting the camera exposure can be straightforward, as in cameras which are *USB Video Class* (UVC) compliant (USB Implementers Forum, 2005). In such cameras, the exposure value is related to the exposure length by a constant $\beta = 0.1 \cdot 10^{-3}$. For the other cameras, the exposure can be in any arbitrary unit, and can also vary with the frame format. A known example is the PS3 Eye camera, in which the exposure range from 0 to 255.

The exposure length (Δe) is not always straightforward to be set, particularly for cameras not UVC compliant. It can be avoided if the user provides two different exposure values that can be easily found by visual inspection. We will call them $\mathbb{E}_{vis}$ and $\mathbb{E}_{inv}$, for the exposure values which render a visible stripe and invisible stripe, respectively. To find $\mathbb{E}_{vis}$, the user must set $\mathbb{E}$ to maximum and reduce it until a stripe composed by three distinct regions become evident (a gradient to dark, a dark region which can be one scanline wide, and a gradient from dark). The value of $\mathbb{E}_{inv}$ depends on the relation between Δt and Δ_{clke}. If $\Delta_{clke} > \Delta t$ then $\mathbb{E}_{inv}$ is given by the maximum $\mathbb{E}$ (Figure 3.4c). If the relation is inverted, starting from maximum, $\mathbb{E}$ must be reduced until a bright band vanishes (Figure 3.4e). Note that in this last case, if β is known, the exposure can be automatically adjusted using Δ_{clke}, which can be estimated using (3.10) as follows

$$\hat{\Delta}_{clke} = \Delta t \cdot 1/\hat{\Delta} f_{/} \tag{3.11}$$

3.4.2 Estimating the stripe position with Kalman filtering

The stripe position detection is very simple and might not be available during some intervals, such as when the stripe enter the invisible scanline range or enter a region where the pupil is expected to be located on the image. As already discussed, the stripe moves at a speed such that its frequency is the result of the difference between the camera clock and the strobe. Therefore, the stripe moves at a constant speed. If this speed is known as well as the stripe position in a given point of time, the position on a new frame can be determined ahead. We propose to use a Kalman

filter to estimate the stripe position, as the detection can be notably noisy.

The Kalman filter is a linear minimum variance of error filter. It is a set of equations which might be used to estimate the position of a moving target from noisy measurements. It involves two steps, prediction and correction or update. The first step uses previous states to predict the current state. The second step uses the current measurement to correct the state (Kalman, 1960).

The stripe position estimation based on Kalman filtering can be formalized as follows. The state of the stripe at each time instance (frame) can be characterized by its position and velocity in the $\mathbb{R}^1$ domain. We assume the constrain of constant velocity, as it is directly related to the drift in frequency between the camera clock and the external one. Let $D_{stripe,k}$ be the position of the stripe on the frame at discrete time k and $D'_{stripe,k}$ be the velocity. The state vector at time k can be represented as $X_k = \left[D_{stripe,k}\ D'_{stripe,k}\right]^T$.

The state vector at the next time frame $(k+1)$ is related to the current state by the system model as follows

$$X_{k+1} = FX_k + w \tag{3.12}$$

where F is the state transition matrix and w represents the system perturbation, which is normally distributed as $p(w) \sim \mathcal{N}(0, Q)$ and Q represents the process noise covariance matrix.

The stripe can be detected on the frame image when on the visible scanline range. We will call the detected stripe at time k as Z_k. As the detection is subject to noise, the measurement model is given by

$$Z_{k+1} = HX_k + v \tag{3.13}$$

where vector $H = [1\ 0]$ describes the state to measurement transformation and v represents the measurement uncertainty, which is normally distributed as $p(v) \sim \mathcal{N}(0, R)$ and R represents the measurement noise covariance matrix.

A Kalman filter usually operates in a two-step predict-update manner. The stripe is detected around a region predicted by the projected state (estimated stripe position based on previous Kalman filter state) and its uncertainty at time $k+1$. The detected stripe is combined with the estimation to produce the final stripe position.

Let X^-_{k+1} denote the a priori state estimate at time $k+1$ and X_{k+1} denote the a posteriori state estimate at time $k+1$. The prediction of the state X^-_{k+1} at time $k+1$ is the result of the previous state X_k, and the system model. The prediction is adjusted in the light of measurement data which result in X_{k+1}. The state X_{k+1} relies on both the system model (3.12) and measurement model (3.13). Associated to the state estimates are the error covariance matrices P^-_{k+1} and P_{k+1}, which characterize their uncertainties.

The constant velocity model assumed is well suited to the problem, as the drift of the clocks are expected to be minimal over time. However, the stripe is periodic and might be on the invisible scanline range, impairing the Kalman update step, as no measurement is possible. The Kalman filter prediction can be summarized as follows (Kalman, 1960):

$$X^-_{k+1} = FX_k \tag{3.14}$$

$$P^-_{k+1} = FP_kF^T + Q \tag{3.15}$$

where F is the state transition matrix and is updated every time the prediction is invoked to reflect the time past from the last frame received to the current, as follows

$$F = \begin{bmatrix} 1 & \bar{\Delta}t \\ 0 & 1 \end{bmatrix} \tag{3.16}$$

where $\bar{\Delta}t$ denotes the time past between the current frame and the previous (between discretized time $k+1$ and k), and might not be equal Δt.

If the stripe position on the projected state is greater than $\hat{S}$ in (3.14), it is updated to the difference of the two (not shown in equation). Accordingly, it is summed to $\hat{S}$ if smaller than zero. The filter update is done by means of the three following equations (Kalman, 1960):

$$K_{k+1} = \frac{P_{k+1}^{-} H}{H P_{k+1}^{-} H^T + R} \tag{3.17}$$

$$X_{k+1} = X_{k+1}^{-} + K_{k+1} \left(Z_{k+1} - H X_{k+1}^{-} \right) \tag{3.18}$$

$$P_{k+1} = \left(I - K_{k+1} H \right) P_{k+1}^{-} \tag{3.19}$$

The initial conditions are brought as byproducts of the estimated camera parameters (Section 3.4.1). The initial state X_0 is composed by the estimated stripe position for the current frame and the velocity derived from a median slope from linear equations $\mathbb{B}_i, i = 1..k$. The initial error covariance matrix is initialized with an estimated variance in the stripe calculation and stripe velocity

$$P_0 = \begin{bmatrix} s_{Dband}^2 & s_{\Delta band}^2 \cdot \Delta t/2 \\ 0 & s_{\Delta band}^2 \end{bmatrix} \tag{3.20}$$

where s_{Dband}^2 is the covariance on the stripe position estimation, considering the detected positions and the associated best fitting line. The velocity variance is given by $s_{\Delta band}^2$, and is found by the variance of the first derivatives of the found $\mathbb{B}$ in Section 3.4.1. We assume $R = s_{Dband}^2$ and the process noise matrix is initialized with zeros, because we assume $q = 0$, despite the effect of the estimated $\hat{S}$ in the prediction.

3.5 Image differencing

The difference between consecutive frames is used to detect pupil candidates. This procedure is also called pupil candidate estimation, with the best candidate further refined as defined in Chapter 4.

This procedure assumes the knowledge of the camera total number of scanlines (S), which is a fixed parameter and can be estimated using the procedure from Section 3.4.1. It also assumes the stripe position and height to be known. The first is estimated using the proposed filter while the second can be calculated beforehand using (3.5).

The image differencing consists of finding the overlap between the pupils on two adjacent frames. For single-colored pupils, the difference is found by subtracting the dark pupil frame from the bright

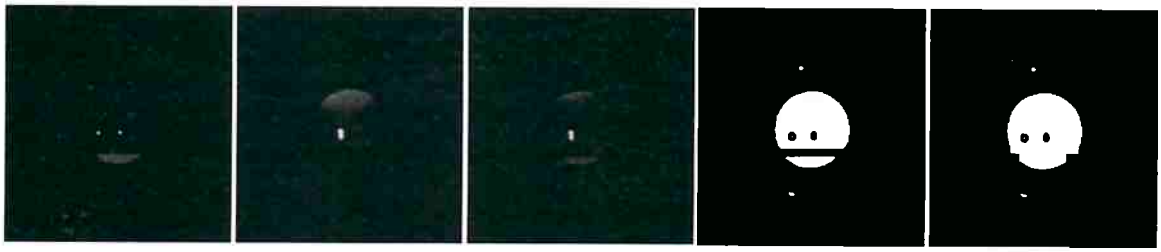


Figure 3.8: *Region of interest excerpts showing the pupil candidate detection for mixed color pupils. a) a db frame; b) the following bd frame; c) image difference; d) result of thresholding; e) line replication result on the predicted stripe position to fill the gap.*

one. With pupils illuminated by two strobes, we must find the difference between the frames, such that the bright pupil portion of the two frames are highlighted. Figure 3.8 shows an example calculation using images taken with $\Delta_{strobe} = 80\mu s$ and strobe firing frequency of $191.072Hz$. In order to reduce processing demands, the difference is calculated only when the stripe enters a *region of interest* (ROI) defined using the last pupil position and size as parameters.

The difference image must be further thresholded. Section 2.4 describes a region growing approach which stops when features detected other than the biggest one start to grow faster. With the stripe over the pupil, two regions start to grow at about the same rate which might result in a premature termination, with a threshold still high, and with only part of the pupil detected (see Figure A.2, frame 56).

To minimize such problems, we simply replicate the lines just before and after the affected region on the thresholded difference image, such that gaps are filled if over the pupil. If the stripe is not on the ROI, the method from Section 2.4 can be used. The affected region is determined by the last and current stripe position. Its size can also be found analytically using (3.7).

3.6 System operation overview

The eye tracking algorithm can only run after the stripe position is successfully tracked, as this information is essential to detect the pupil features correctly as well augment the CR detection. The stripe is tracked with the help of a Kalman filter, which can precisely estimate the stripe position even in the presence of a noisy image-based stripe detection. The filter is initialized with parameters found during the estimation of the camera total scanlines S using the procedure described in Section 3.4.1.

Besides the noise in the stripe detection, the filter is also important because the stripe is not visible on all frames. Besides the invisible scanlines, the stripe is also reduced to minimum when it enters a ROI where the pupil and CRs are expected to lie on. Before entering the cycle of increasing and decreasing the exposure as the stripe moves, the Kalman filter must be allowed to run a few predict and update cycles to allow the internal state to better reflect the system state (the stripe velocity and location).

The filter is allowed to estimate the stripe position only after a frame is received, as the transition matrix is updated with the actual camera period Δt. Note the filter is still discretized, but with variable steps. The period is calculated by the difference between the current frame timestamp and the previous. While this procedure might introduce some fluctuations to the period which is

otherwise fixed, on the long run the stripe is correctly estimated even if a frame is dropped due to an I/O surge, for example.

After the stripe position is predicted by the filter, it might be detected on the frame image if the exposure was configured to $\mathbb{E}_{vis}$ and the predicted position is inside the visible scanline range. In case the detection is done and results in a candidate location, the filter is updated. The stripe position is then used by the eye tracking algorithm. Figure 3.9 summarizes the aforementioned steps.

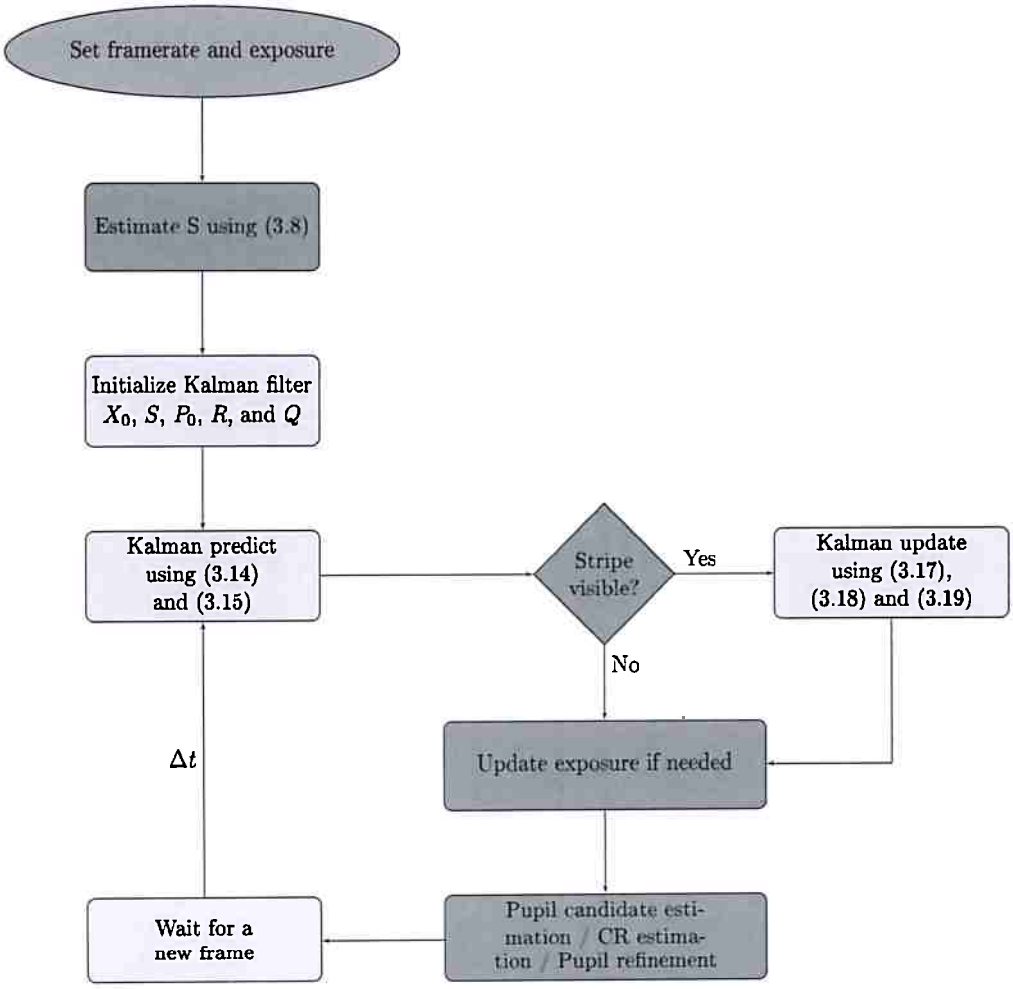


Figure 3.9: *Flowchart of the asynchronous eye tracking system. No interaction with the stroboscopic illumination is needed.*

3.7 Experimental results

The results are divided into two parts, starting by the estimation of the camera total number of scanlines by the simple method presented in Section 3.4.1. Following is the result of the stripe position estimation and filtering, which in addition to S, also employs the noise estimated on the previous part.

The prototype described in Section E.3 is used. The lens employed, at the nominal distance of $13cm$, covers an area of approximated $25 \times 19mm$. This gives about $12.8pixels/mm$. All experiments were done at $187Hz$. While a high frequency can be quite challenging, lower frequencies might

require much more time to get the system working. A discussion is presented after the results, in Section 3.8.

3.7.1 Estimating the total scanlines

The estimation process rely on the detection of the stripe position in a given frame. The stripe is located in the frame column image, a vector containing the average of each image line. Calculating such vector is time consuming, thus we employ a sampled version, in which only 10% of the columns enter the sum. In our simplified version, the stripe is detected as the line with the lowest intensity, as far as it satisfies the condition of being at least three times darker than the median intensity. Such condition lead to some false negative detections, as the stripe is present but not sufficiently dark. However, their effects are negligible compared to possible false positive detections.

The experiment was performed considering five different strobe frequencies, $-1Hz$, $-2Hz$, $+1Hz$, $+2Hz$ and $+2\%$, all with respect to the camera clock. The camera was configured at the nominal frequency of $187Hz$ and the images taken while a subject was wearing the apparatus. The exposure was manually set to produce the minimal visible stripe in each case, as shown in Figure 5.3.

The actual camera frequency was found by configuring the strobes such that the stripe presented the least movement. The external clock granularity is $62.5ns$, and the camera clock was found to be between $187.3251Hz$ (period of $62.5 \cdot 85413ns$) and $187.3273Hz$ (period of $62.5 \cdot 85412ns$), which produced a movement of 0.33 scanlines/s and -0.28 scanlines/s, respectively. The same test repeated 18 days later, resulted in 0.33 scanlines/s and -0.27 scanlines/s of stripe movement. Thus, a drift of $5.31 \cdot 10^{-10}$ occurred between the strobe clock (an Arduino board driven with a quartz resonator) and the camera in 18 days (also equipped with a crystal resonator).

Figure 3.10 shows the calculation for the first three seconds for each strobe period. In blue are the first order curves representing the stripe position at a given time and in red, the estimated S. We also added the final S estimated, found after from 26 to 39.5 seconds. Note that the S estimated using only two or three regressions is a reasonable approximation of the value found after several seconds.

3.7.2 Stripe estimation and filtering

The correct stripe position tracking is an important step for the whole eye tracking system, as the pupil and CR estimation depend on such for a considerable fraction of the total frames processed. Ideally, it must be precise, accurate and lightweight. The most time consuming step is the estimation of the stripe position from the frame image, which involves reading and accumulating all the pixel intensities. A sampled version is used instead, but the simplification might add noise to the resulting stripe position.

A Kalman filter is employed to deal with such noise and to allow the stripe position prediction when the image-augmented detection is not available. The update step of the Kalman filter runs on at most 34.5% of the total frames captured. The hidden scanlines, which are almost 14% of the total scanlines plus 3/5 of the visible scanlines exploit the filter prediction to determine the stripe location. The mentioned fraction of visible scanlines are not used because the exposure is increased to reduce the stripe the most, such that their effects are minimized to the eye tracking algorithm.

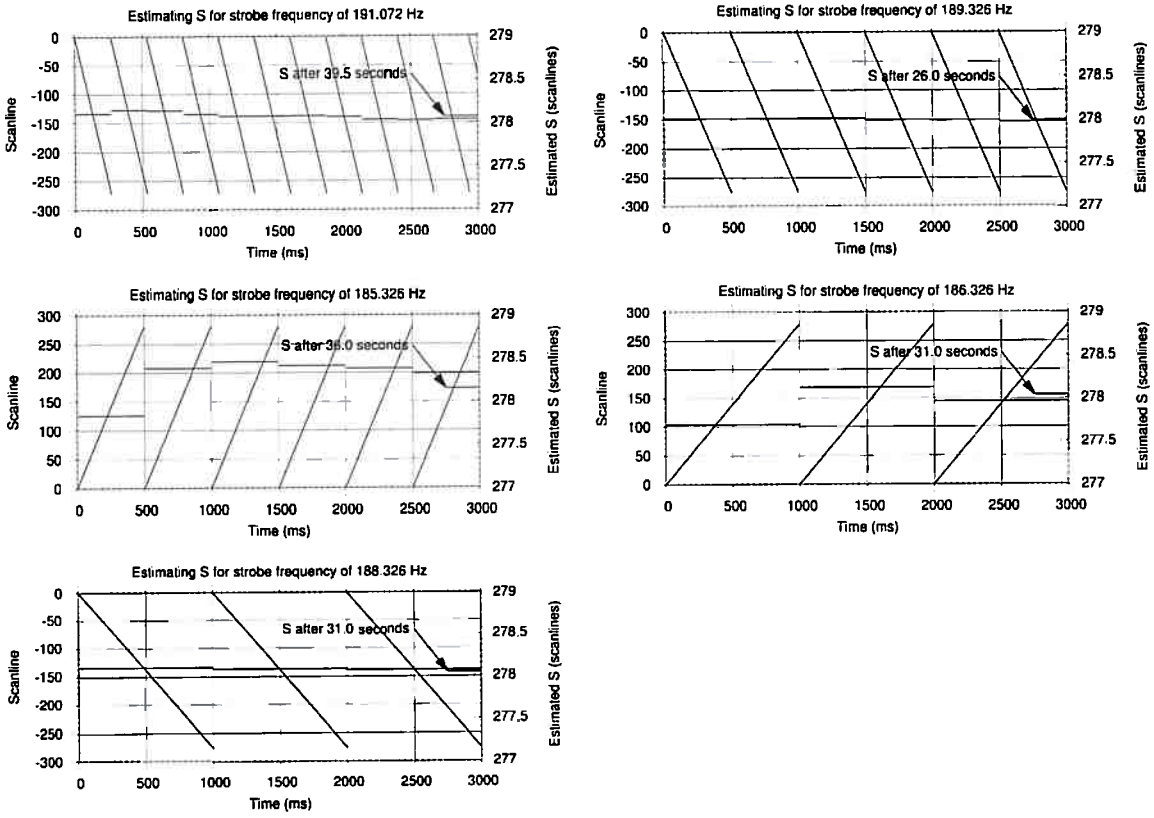


Figure 3.10: *Total scanlines (S) estimated using the procedure described in Section 3.4.1. Each graph shows a time excerpt of three seconds at different strobe frequencies with the first order regressors found.*

We have used a fixed proportion instead of a based on a ROI given by the eye tracking algorithm because the later were run offline, with the data already collected.

Figure 3.11a shows the filtering result for the five strobe triggering frequencies. For all cases, the gain drops steadily after some initial fluctuations (red points). The graphs also show the change in the estimated stripe velocity with respect to the calculated when the total scanlines were estimated (violet points). Note the velocity at the end of more than 50 seconds do not differ from more than 0.01% the initial estimation. The graphs also show the difference between the predicted stripe position with respect to the estimated from the frame images (blue points). A *simple moving average* (SMA) using a time window of one second where done to highlight deviations from zero (dark blue points).

Figure 3.11b shows the time taken to estimate the stripe using the frame image (brown points) and the Kalman filter (green points). The points can be grouped into three distinct regions, two green at the bottom and a brown in the middle. The closest to zero correspond to the prediction step of the filter, and is done for every frame. The next group correspond to the update step, which is done only when the stripe is estimated from the image (the last region). A density plot of the total processing time is presented in Figure 3.12.

3.8 Discussion

A new technique to estimate the total scanlines of a camera was proposed. It exploits artifacts generated in the frame images taken with stroboscopic light. In an experiment with a PS3 Eye

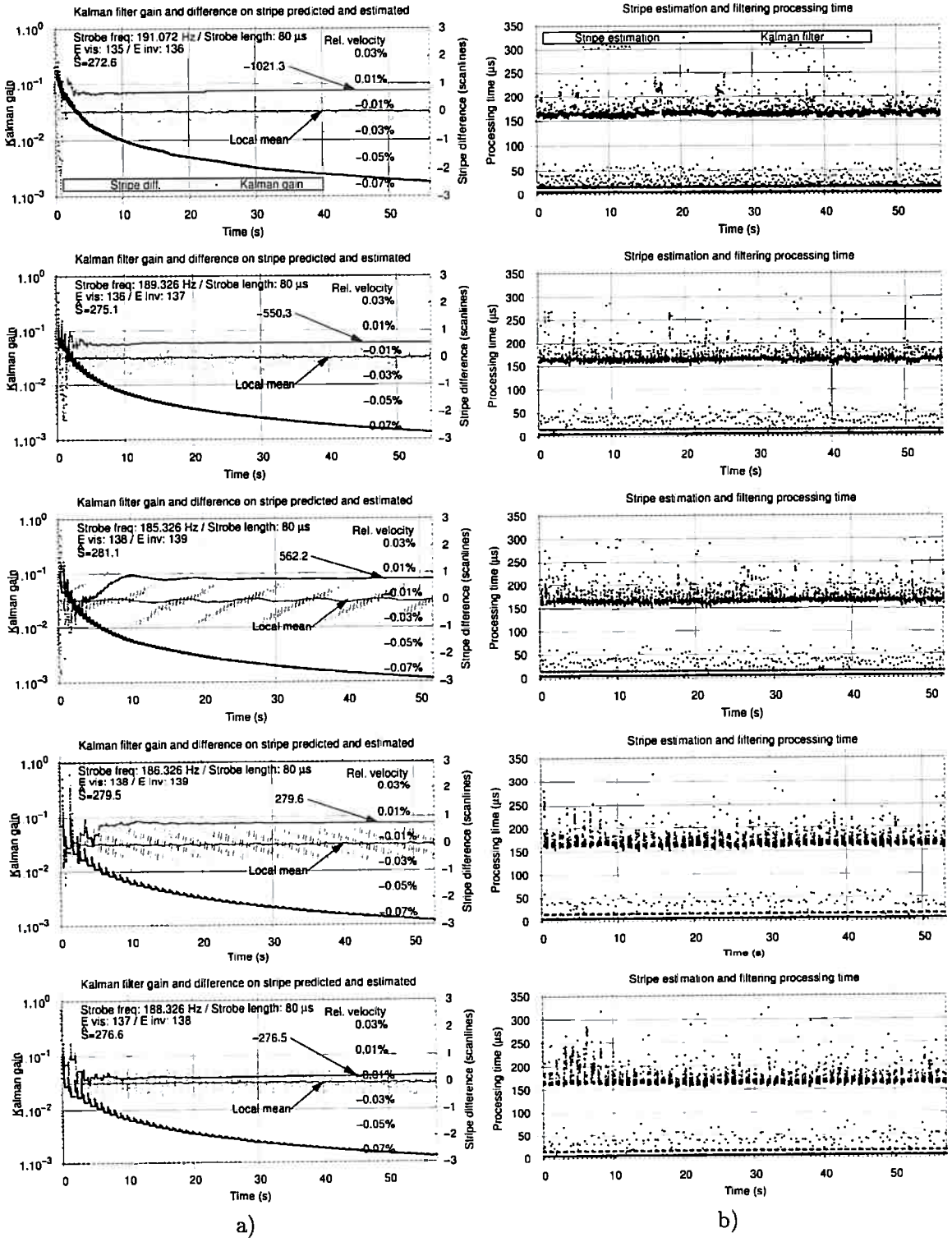


Figure 3.11: *a) the Kalman filter gain, stripe speed estimation (internal filter state) and difference between the stripe position as estimated by the filter and the calculated from the image; b) the processing time needed to track the stripe position.*

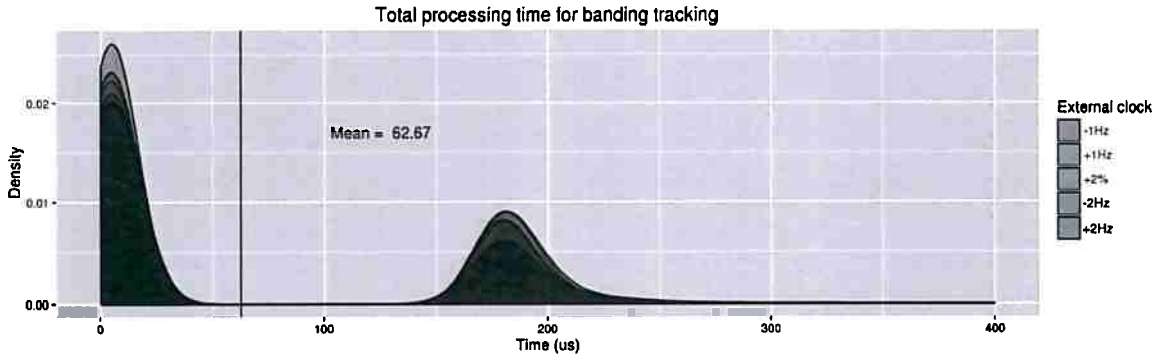


Figure 3.12: *Consolidated processing time needed to track the stripe based on 51146 frames processed.*

Camera at $187Hz$, the method performed well, diverging from the ground truth by at most 0.052%. The mean value found was 278.0575 ($SD = 0.0565$), against the ground truth of 278 scanlines.

The Kalman filter proposed to track the stripe position also performed well. The internal gain of the filter (K) decreased down to a mean value of $1.36 \cdot 10^{-3}$ ($SD = 2.67 \cdot 10^{-4}$). With such values, a stripe wrongly detected on the frame image by half the total scanlines (139) would make the filter deviates in just 0.189 scanlines. The internal velocity did not changed on the last five seconds for 3 out of the 5 tests run. For the filters which had changes, it were on the order of $3.57 \cdot 10^{-6}$ and $7.23 \cdot 10^{-6}$, equivalent to 0.001 and 0.002 scanlines per second. With those results, we believe the filter entered a steady state, despite the long intervals without updates (65.5% of the frames processed).

The filter as proposed is efficient on tracking the stripe. The processing time distribution is bimodal, reflecting the image augmented detection for a fraction of the frames. The overall processing time average was just $62.67\mu s$. When considering only the frames where image operations were done, the average rises to $186.56\mu s$.

The highest sample rate of the camera was chosen for the experiments. While it is challenging as the processing window is reduced, the initial estimations are done much faster. For example, a 2% frequency difference is equivalent to $3.7Hz$ for a camera at $187Hz$, and only $0.6Hz$ for a camera at $30Hz$. This implies in a stripe much slower at a lower camera speed, increasing the estimation time of the total scanlines, which must wait for the stripe to traverse all scanlines a couple of times.

3.8.1 A note on clock drifts

Although crystal oscillators are very stable over time, they suffer from aging, a gradual change in frequency over time on the order of ±3ppm per year. They are also sensitive to variations in temperature, on the order of ±20ppm over the range $(-20°C \sim 70°C)$[3]. As those deviations are very small, minimal phase drifts are expected from crystal-based oscillators prescaled by about 10^5.

The system model employed on the Kalman filter of Section 3.4.2 assumes a constant velocity to track the stripe position over time. The velocity is given by the difference on the clock signal of the camera and external clock (Δf). While the camera employs a quartz oscillator and clock dividers to set the frame rate, we proposed a simple analog circuit to be our external clock source (see Section D.1). We also proposed the use of a microcontroller, which was evaluated and provided

[3]Representative data. The exact values are from surface mount crystal SPK-5032 of 16.000MHz. Specification online, accessed on 07/24/2016.

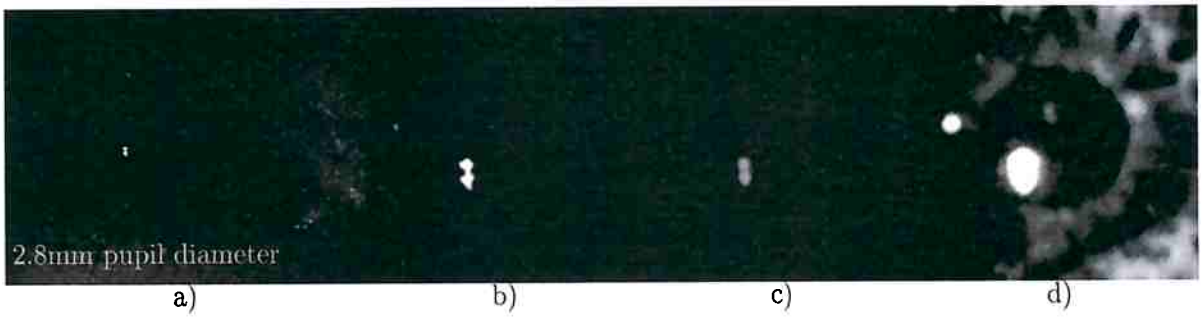


Figure 3.13: *Sample image from the eye of a volunteer subject to* $2000lux$ *measured at the face. a) Image as captured by our prototype; b) Pupil region detail (*$\times 3.2$*); c) Pupil after filtering with a gaussian kernel of size three; d) Contrast enhanced by ten times.*

good stability (see Section D.2).

The analog circuit employs a RC oscillator network, which is influenced by deviations of the two components. For capacitors and resistors, the long term stability is named drift. The capacitance drift for a *polypropylene film* (pp) capacitor is on the order of 0.5% (Vishay Roederstein, 2012). The resistance drift of a thin film resistor is about than 0.3% after 225 thousand hours of operation (Vishay Intertechnology Inc., 2013). With respect to the temperature, both are more sensitive. A pp capacitor have deviations of $-200, \pm 100\text{ppm}/^{\circ}C$ over the range $(-20^{\circ}C \sim 60^{\circ}C)$ (Vishay Roederstein, 2012). The change in resistance due to variations in temperature is given by the so called *temperature coefficient of resistance* (TCR), which typically varies from 5 to $200\text{ppm}/^{\circ}C$.

Considering the data presented, a RC oscillator can easily be a thousandfold less stable to temperature changes than a crystal-based one. However, considering the results found after a 18 day period using crystal oscillators, three orders of magnitude are expected to not pose problems to the filter.

3.8.2 Limitations

Extreme pupil sizes might result in tracking problems, depending on the resolution of the setup (relative to the field of view in pixel/mm). For small pupils, the problem is twofold. When a constricted pupil is imaged, the bright pupil response is reduced (see Figure 3.13). This results in a lack of contrast between the pupil and iris, making the border harder to detect. A second problem is associated to the stripe, particularly when it is over the pupil. For a given pulse length, the stripe size might be such as to result in no usable rays, impairing the pupil detection. For large pupils, the bright pupil effect might be such that a corneal reflection becomes as bright as the pupil. Such a condition might bias the CR estimation, despite our weighted center of mass calculation (4.2).

Figure 3.13 shows an eye with a constricted pupil, subject to $2000lux$ illumination by a white *light-emitting diode* (LED) source. The field of view of our prototype at the working distance of $130mm$ is $26 \times 19mm$, allowing us to estimate the pupil size to be about $2.8mm$. With the smaller aperture, the bright pupil effect is reduced and the bright pupil lacks contrast. Note in Figure 3.13d, that the pupil is not brighter than the iris and a dark border in the pupil aperture is as dark as other iris regions. In Appendix A, Figure A.2, we show the result of our tracking algorithm when the stripe pass over such a small pupil.

With the asynchronous illumination, frames will be composed by two snapshots taken Δ_{clke} from each other most of the time. This might render the pupil to appear broken during a saccade

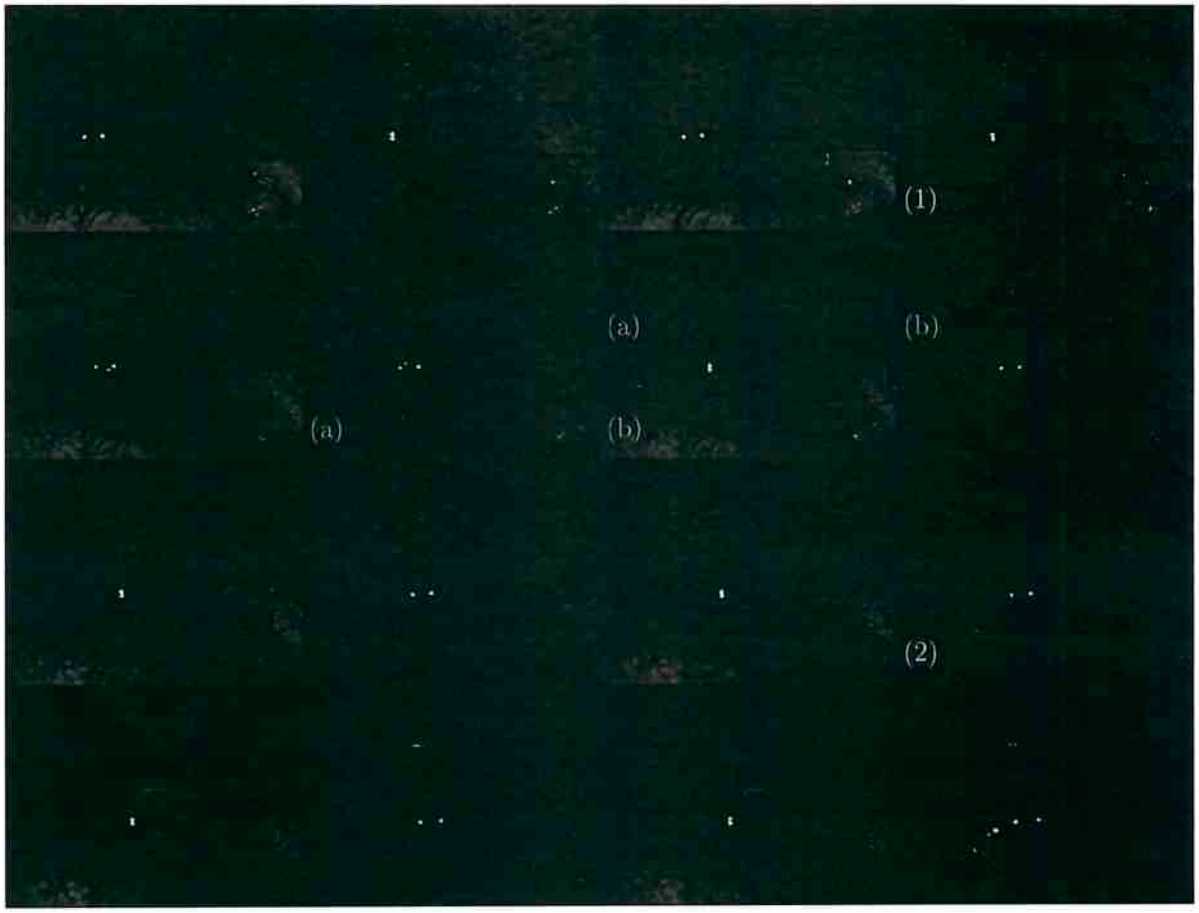


Figure 3.14: *Images captured at* $187.326Hz$ *with an ongoing horizontal saccade. The samples represent an interval of* $85ms$. *The external clock was at* $191.072Hz$.

(see Figure 3.14). It also increases the standard deviation of the response time of the whole system. On the *differential stroboscopic lighting* (DSL), the light pulse was triggered at the very beginning of a frame, but due to the rolling shutter mechanism, this was equivalent to half the integration interval. The gaze coordinates were output only after the integration finished and the processing done. Now the pulses are triggered in two different instants during the integration, and depending on the pupil position on the frame and the lag between the clocks, might render a snapshot of the pupil considerably close or far of the integration finishing time.

Figure 3.14 shows sixteen frames captured during a saccade. Assuming we have a constant ξ which translates the stripe position to the lag between the clocks, the pupil image in Figure 3.14(1), is $\Delta_{clke} + \xi \cdot D_{stripe}$ from the end of the integration time, while for Figure 3.14(2), the pupil is a snapshot from time $\xi \cdot D_{stripe}$. The pupil appears broken when the stripe pass over it as the images are Δ_{clke} apart from each other.

To employ the asynchronous technique for trajectory analysis, the stripe position must be used to place the pupil estimation on the correct time frame. Assuming the images are equally spaced in time does not hold. For real time applications, two options are available: 1. deliver the pupil estimation at the end of processing, rendering a variable response time (with respect to the true pupil position); 2. deliver the pupil estimation at variable times with respect to the end of integration time, rendering a constant, but even more delayed response time. The delay must be at most $2\Delta t$ and can be estimated using the stripe position. This last approach is equivalent to output pupil

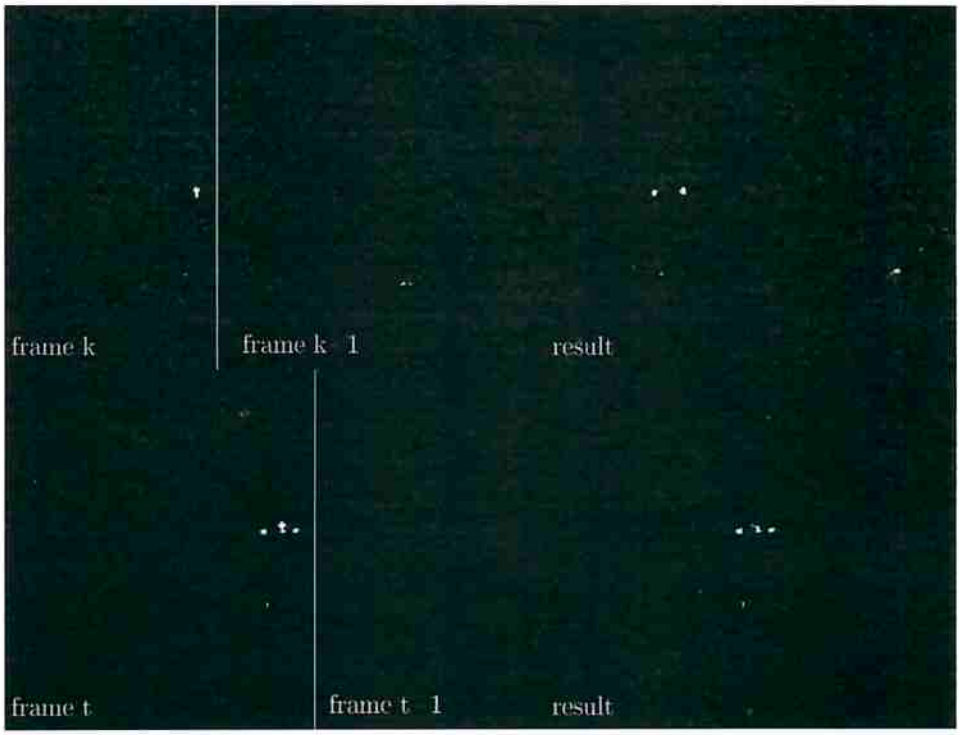


Figure 3.15: *Joining consecutive frames. Ongoing saccade on both samples. On top, $k = 486$ from trial T1, and on bottom, $t = 5378$ from trial T4.*

data synchronized with the external clock, instead of with the frames.

By looking at Figure 3.14, one might think to join the bottom portion of one frame with the top portion of the next. For example, the half frame identified with (a) was lit by the same strobe firing as the top (a) portion of the next frame. The same occurring to the (b) portions and so on. As they were illuminated by the same pulse, they will not suffer from a broken pupil during a saccade, except for the scanlines affected by the stripe translation (particularly when a higher external clock is employed).

Figure 3.15 shows the result of joining the bottom portion of a frame (up to the stripe) with the top portion of the next, for external clocks at a lower frequency (top), and higher frequency (bottom). In the first case, it is possible to join the images without artifacts by increasing the clock difference. Shown is a difference of about 1%. For the case of a higher frequency, the greater the difference, bigger the gap between the frames, which in this case is filled with the next strobe pulse. Despite the possible advantages, the response time of the system will be limited to at least Δt, varying with the clock drift from Δt to $2\Delta t$. Our approach, on the other hand, can have responses down to $\Delta t/2$, depending on the pupil height on frame coordinates[4].

[4]Frame data transport and processing not included in any latency considerations on the whole section.

Chapter 4

Pupil refinement and corneal reflection estimation

Chapters 2 and 3 show how the *differential lighting* (DL) can be extended to detect the pupil using digital consumer-grade cameras. The use of stroboscopic light improves camera image quality, reducing noise, motion blur, and deformations due to the rolling shutter mechanism.

The detect pupil however, corresponds to the overlap between adjacent frames, and accordingly, is a roughly estimation of the pupil. The problem is aggravated during high speed eye movements, as the overlap is reduced. To improve system accuracy and precision the pupil must be further refined.

Another problem with the estimated pupil arise when high sample rates are employed. Low-cost consumer-grade cameras typically trade spatial resolution for temporal resolution. Thus, we need to accurately detect the pupil and *corneal reflections* (CRs), in order to increase the resolution of the pupil-CR vector used for calibration purposes.

4.1 Corneal reflection detection and localization

We use an adaptive thresholding technique in each frame to detect the corneal reflections. It is an optimized version of the method employed on the *Starburst* (Li and Parkhurst, 2005). Before starting, an inverted cumulative histogram of the eye boundary is found ($\mathbb{H}$) using (2.10). It helps on increasing the detection speed by allowing variable stepping through intensity levels. Two parameters are used to skip small specular reflections and large ones. One is the minimum corneal reflection allowed (g_{min}), which is used to bootstrap the algorithm with the initial threshold using (2.11) with $A = \pi \cdot g_{min}^2$. The other is the maximum corneal reflection (g_{max}), which is part of the stopping criterion.

Ideally, by binarizing the image with such threshold would result in the true corneal reflection (i.e. a filled circle which center coincides with the true reflex). However, other small specular reflections tend to be as bright as the found threshold. Additionally, g_{min} can be a really gross estimation in practice (e.g. can be zero as well).

The next step is to take all the disjoint contours on the binary image, do sanity checks (such as removing areas on the image border) and fit a circle to each. If the enclosing circle radius is smaller than g_{min}, the contour is discarded in the current iteration. By summing the area of all remaining

circles and replacing A in (2.11), the threshold of the next iteration is found. If the area is zero or *light-emitting diode* (LED) to the same threshold, the next value is the current decremented by one.

Each corneal reflex candidate g is modeled as a 5-tuple $(l_g, c_g, r_g, d_g, \hat{r}_g)$, where l_g is the number of iterations in which g is present; c_g is the coordinate of the reflex center, r_g is the latest radius of the enclosing circle; d_g is the euclidean distance to the last pupil center found; and $\hat{r}_g$ is the ratio between the sum of the pixels at the circumference centered at c_g with radius r_g and radius $r_g + 1$. At each iteration, an ordered list of the corneal reflex candidates is kept. The candidates are ordered following a quality function defined as

$$\text{quality}(g) = ((l_g + 1) \cdot r_g \cdot (1.0/(d_g + 0.1)) \cdot \hat{r}_i) \tag{4.1}$$

where g is the candidate. The intuition behind the function with each term is: (l_g+1) - long lived areas are prone to belong to the reflexes we look for; (r_g) - the radius of the strongest reflexes on the cornea tend to grow earlier than on other areas (e.g. where the eye lids meet); $(1.0/(d_g+0.1))$ - we give preference for reflexes close to the last estimated pupil location; $(\hat{r}_i)$ - this term is maximum when r_g is covering the whole reflex.

The lists of all iterations are arranged in a tree-like structure in which leaf without parents are allowed at any level. The term l_g is derived from the length of the path from the leave (at the current iteration list) to the head, if present. The parenthood between g and f, nodes on two adjacent lists, is defined if the euclidean distance between c_g and c_f is lower than $min(r_g, r_f)$.

If the stop criterion is met, the first one or two candidates are returned as the areas containing the corneal reflexes, depending if a bright or dark pupil frame is being processed, respectively. There is more than one condition that triggers the end of the detection. The ideal case occurs when the radius of the best candidates do not change from one iteration to another. However, it might occur that the two best candidates are joined together or one of them is joined to an area around it, formerly not present. In any case, there is a degradation due to excess of iterations (i.e. the threshold was reduced more than desirable). When such condition is detected, the candidates returned are brought from the list of the prior iteration.

Once the corneal reflexes are detected, there is an additional step to refine their centers. As in (Li and Parkhurst, 2005), we assume the reflex intensity profile to follow a symmetric bivariate Gaussian distribution. In addition, the image has also an additive component, which is considerably strong in bright pupil frames. The center is refined by calculating the center of mass of a squared *region of interest* (ROI) centered on the initial estimation. However, a shifted estimation results in a biased center of mass. Considering the reflex intensity profile as described, the greatest contribution of the Gaussian occurs closest to its mean. Let (x_i, y_i), $i = 1...N$ denote the image domain coordinates of the pixels inside the ROI. The center (x_c, y_c) of the reflex is obtained by calculating a normalized center of mass function that weights the pixels according to their intensities in a non-linear manner, as follows:

$$(x_c, y_c) = \frac{1}{N} \sum_{i=1}^{N} (e^{\varrho \cdot I(x_i, y_i) - \varrho} \cdot (x_i, y_i)) \tag{4.2}$$

where N is the total number of pixels inside the ROI, the function $I(.)$ returns the pixel intensity normalized in the interval $[0, 1]$ and ϱ is a weighting constant. Higher ϱ render (4.2) to pick the

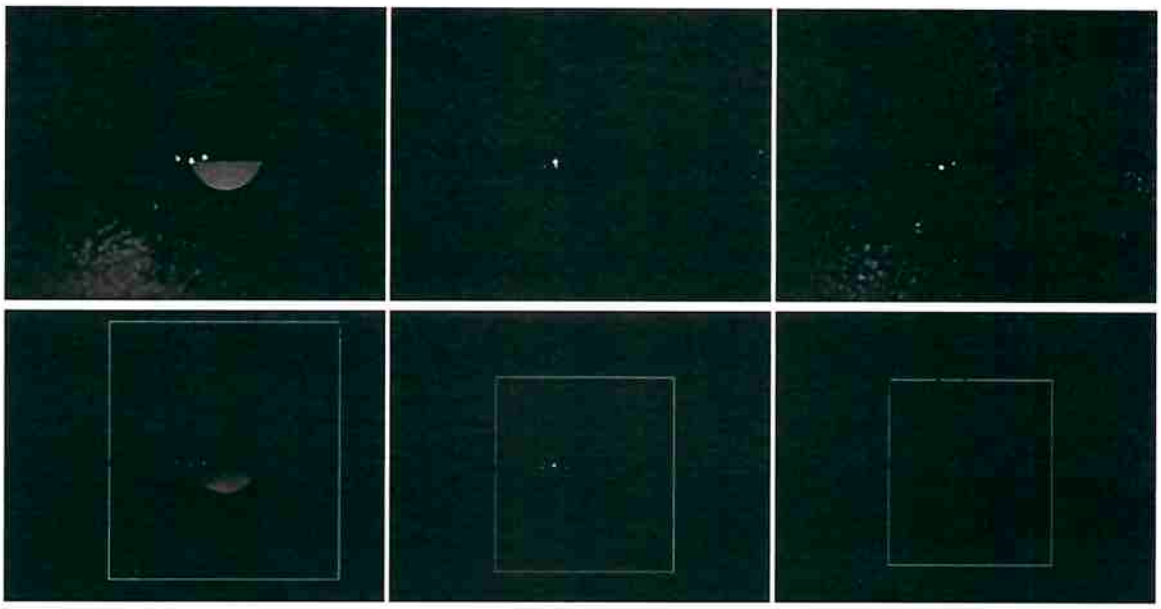


Figure 4.1: *a) Original frame with both bright and dark illuminators reflected (top), and tracking result (bottom). b) and c) Not all CRs found, which makes the eye tracking to assume the last successful found ones as the actual reflexes position. The detected pupil contour is in blue and the pupil-CR vector in red.*

coordinate of the strongest pixel as the center while smaller values makes it to consider all the pixel intensities.

4.1.1 ADSL corneal reflection estimation

With mixed-color pupils and particularly for external clock frequencies higher than the one of the camera, it might happen that both the bright and dark illuminators produce reflexes on the same frame. This can only occur on the scanlines lit by the two strobes, which represents part of the actual stripe. To deal with such cases, we assume three reflections when the stripe intersects with the CRs currently on the first positions of the candidate list. Figure 4.1a shows an example of a frame with three reflexes and the resulting pupil-CR vector on bottom.

When an external clock with a lower frequency with respect to the camera is employed, a dark stripe is present and might result in no CRs at all. Figure 4.1bc show exemplars of corrupted corneal reflections due to the stripe. In this case, the algorithm search for three reflexes as well, returning the last valid CR midpoint (from a previous frame) in case less reflexes are found. This is done because the extent of corruption is hard to predict beforehand, as it depends on the stripe profile and the reflex position with respect to the stripe.

Figure 4.1 shows exemplars of frames where the stripe coincide with the CRs. The frame image in (a) correspond to a capture with the external clock 2% higher in frequency with respect to the camera (at 187Hz). In such a case, the exposure can be adjusted such that the stripe almost vanishes. The images in (b) and (c) on the other hand, are from a capture with the external clock $2Hz$ lower in frequency, case in which the stripe is present even at maximum exposure. Note the problem only grows with a lengthy strobe (Δ_{strobe}) and lower clock frequency (higher Δ_{clke}).

4.2 Sub-pixel pupil contour estimation

The pupil contour estimation is an important step on retrieving the gaze from video images. Starting with the contour, the center of the pupil can be readily found. With the help of the DL, the initial estimation of the pupil position becomes more reliable and robust (Ebisawa, 1998; Morimoto *et al.*, 2000).

The pupil contour refinement starts with a candidate estimation of the pupil, done by either the method presented in Section 2.4 or Section 3.5. An ellipse is fitted to such region (e_o), and is defined by a 4-tuple $(a, b, c, \hat{\theta})$, where a and b are the major and minor axes, $c = (c_x, c_y)$ are the center coordinates, and $\hat{\theta}$ is the rotation angle. Note that while fixating, the overlap produced will be very close to the current pupil, while during saccades it will resemble the shape of a circle-to-circle intersection.

We use a feature-based method to detect the pupil contour based on the work from Ohno *et al.* (2002). Starting at the candidate region, a number of rays R_k, $k = 1...m$ are projected outwards to detect the pupil edges. The length of each ray is proportional to the size of the overlap. The length of each ray is calculated as follows:

$$|R| = e_o(a) \cdot \upsilon \tag{4.3}$$

where υ controls the proportion. Particularly at low frame rates, υ must be increased with respect to a higher frame rate to guarantee the rays cross the pupil-iris boundary perpendicularly to the pupil moving direction.

The intensities of the pixels along each ray are estimated using bi-linear interpolation. The rays are convolved with an 1-d *derivative of gaussian* (DoG) kernel of size 15. Given the transition in brightness is steepest on the boundary of the pupil with the iris, the convolution results in a discrete shape which approximates a gaussian. When looking for a bright pupil, the border will be around the strongest positive peak. If otherwise, we look for a dark pupil, the border will be around the strongest negative peak.

The number of rays is fixed, in contrast to the *Starburst* approach which projects rays until convergence. The original *differential stroboscopic lighting* (DSL) trace rays in a similar manner, but the rays follow integral pixels and the contour candidate is on the strongest discrete response of a 1-d gradient convolution filter (Borsato *et al.*, 2015).

Instead of selecting the strongest response on the discrete convolution as the border location, we fit a second-order function around the peak. The maximum or minimum of the quadratic estimator is used as the border position with sub-pixel accuracy. The estimator uses only three samples from the convolution, the strongest peak and the direct neighbors.

The quadratic estimator can be expressed as (Dvornychenko, 1983)

$$\Delta_{peak} = \frac{C_r - C_l}{2 \cdot (2 \cdot C_{peak} - C_r - C_l)} \tag{4.4}$$

where C_{peak} is the strongest response of the convolution, C_l and C_r are the responses to the left and right of C_{peak}, and Δ_{peak} is the estimated location of the peak in terms of the sample interval with respect to C_{peak}. A Gaussian fitting could be obtained by applying a second-order polynomial to the logarithm of C_{peak}, C_l and C_r, at the cost of three logarithm calculations per ray. Figure 4.6 shows

the result of the refinement of pupil contours using both the quadratic and gaussian estimators against ground truth. There is no actual gain in performance. The same is true in Figures 4.2 and 4.3 with respect to the curves generated.

The rays can be distributed is several ways. It is desirable that they intersect the pupil-iris boundary as normal as possible. If the rays are projected from the overlap center evenly spaced, the pupil border close to the foci will be proportionally less sampled than that close to the smaller axis. This is particularly true for very eccentric pupils, such as the one from Figure 4.4(a). To consider other possible distribution patterns, let a new ellipse (e_i) be the origin of the rays. Consider its axes are weighted by the following factors

$$f_a = (1 - \psi) \tag{4.5a}$$

$$f_b = (1 - \psi) \cdot (e_o(b)/e_o(a) + (1 - e_o(b)/e_o(a))/2), \tag{4.5b}$$

such that $e_i(a) = e_o(a) \cdot f_a$ and $e_i(b) = e_o(b) \cdot f_b$. The constant ψ range from 0 to 1, and defines how smaller the new ellipse will be.

Consider a function $Q(e_i, \hat{\tau})$, which accepts an ellipse and an eccentric anomaly angle $\hat{\tau}$ and returns a point at the ellipse (the angle is the independent parameter). If one chooses to distribute the points following directions defined by $[Q(e_i, \hat{\tau}) - e_i(c)]$, for $\hat{\tau} = 0...2\pi$, the areas close to the foci will be more sampled than before. Figure 4.4(b) shows an example. However, the rays close to the smaller axis are skewed with respect to the expected gradient produced by the pupil-iris border.

Another way is to trace the rays normal to ellipse e_i. This produces rays closer to normal with respect to the overlap and possibly to the true pupil. However, the regions close to the foci are again less sampled (see Figure 4.4(c)).

We adopt an in-between approach. The rays are distributed following directions defined by $[Q(e_e, \hat{\tau}) - Q(e_i, \hat{\tau})]$, for $\hat{\tau} = 0...2\pi$, where e_e is an ellipse centered in $e_o(c)$ with axes given by $e_e(a) = e_o(a)/f_a$ and $e_e(b) = e_o(b)/f_b$. Figure 4.4(d) depicts such distribution.

As the pupil refinement is done after the CRs are estimated, rays intersecting them can be prematurely discarded, avoiding extra processing demands for later filtering. The angular displacement of the CRs with respect to the pupil center are estimated considering their size and distance to the overlap center. The rays inside such range are not traced, but new rays are inserted on the surrounding areas. The intuition behind this processes is to lessen the contribution of the remaining rays which might bias the pupil estimated. The idea is to distribute the rays evenly in a local mean distribution sense.

Each ray generates a convolution array with a peak hopefully corresponding to the location of the transition between the iris and pupil. Greater the contrast between the two regions, higher the strength of the peak response. When the region sampled under the ray have noisy data, i.e. eyelashes, eyelid, reflexes, and so on, multiple equally strong peaks may arise. A normalized weight is used to sort the rays, and is defined as follows:

$$W_k = \max_{j=1...|R|} \left(\frac{R_k^*(j)}{\sum_{i=0}^{|R|} R_k^*(i)} \right) \tag{4.6}$$

where $R_k^*(j)$ denotes the convolution of ray R_k at position j, and W_k the weight attributed to ray R_k.

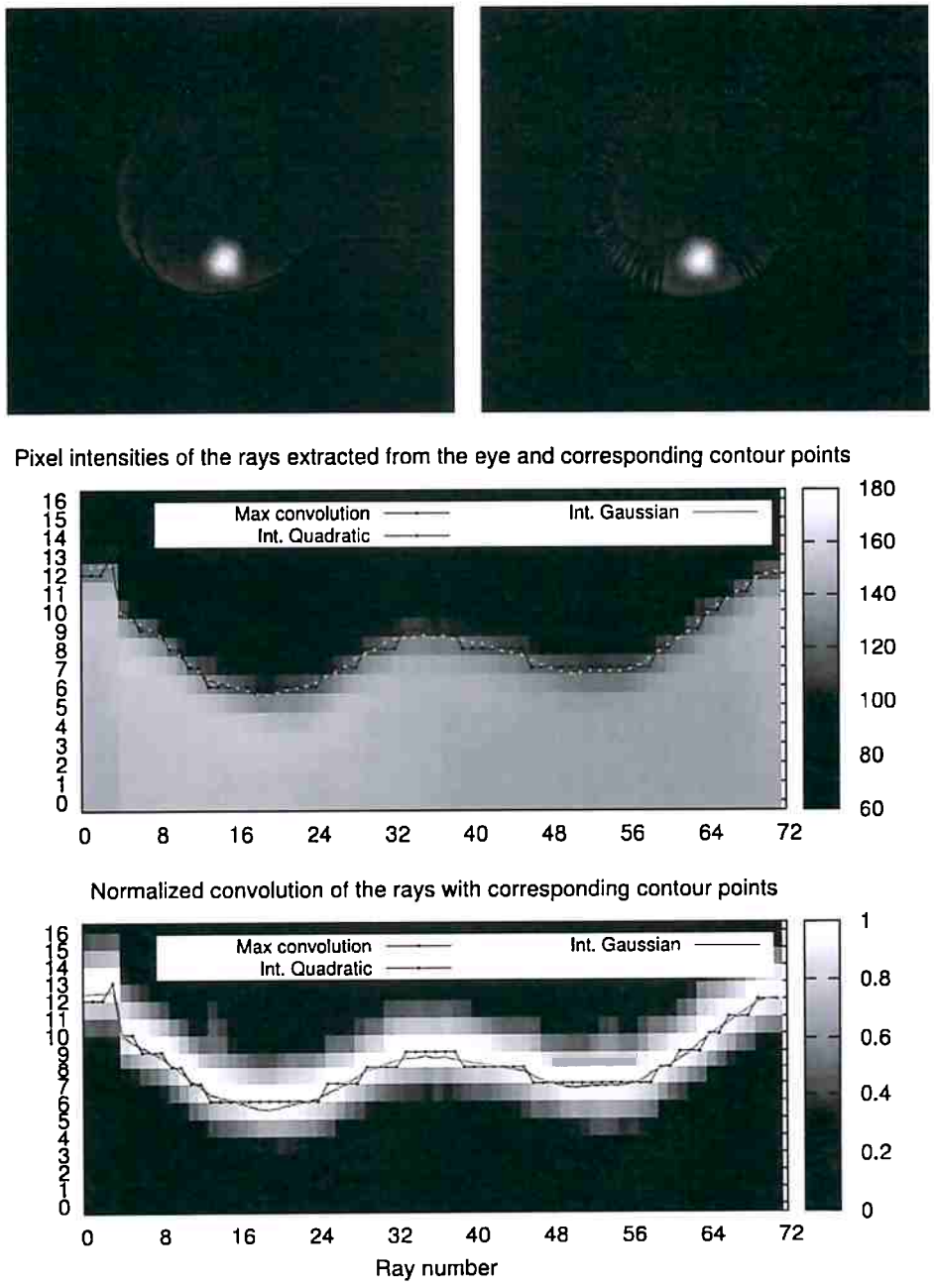


Figure 4.2: *The intensity profiles of each ray extracted from a bright pupil. A total of 72 rays were obtained, in which the red line represents the strongest filter response, while the green and blue represent the result of interpolation. (Middle) Rays resulting from bi-linear interpolation along the ray trajectory. (Bottom) Result of the convolution for each ray.*

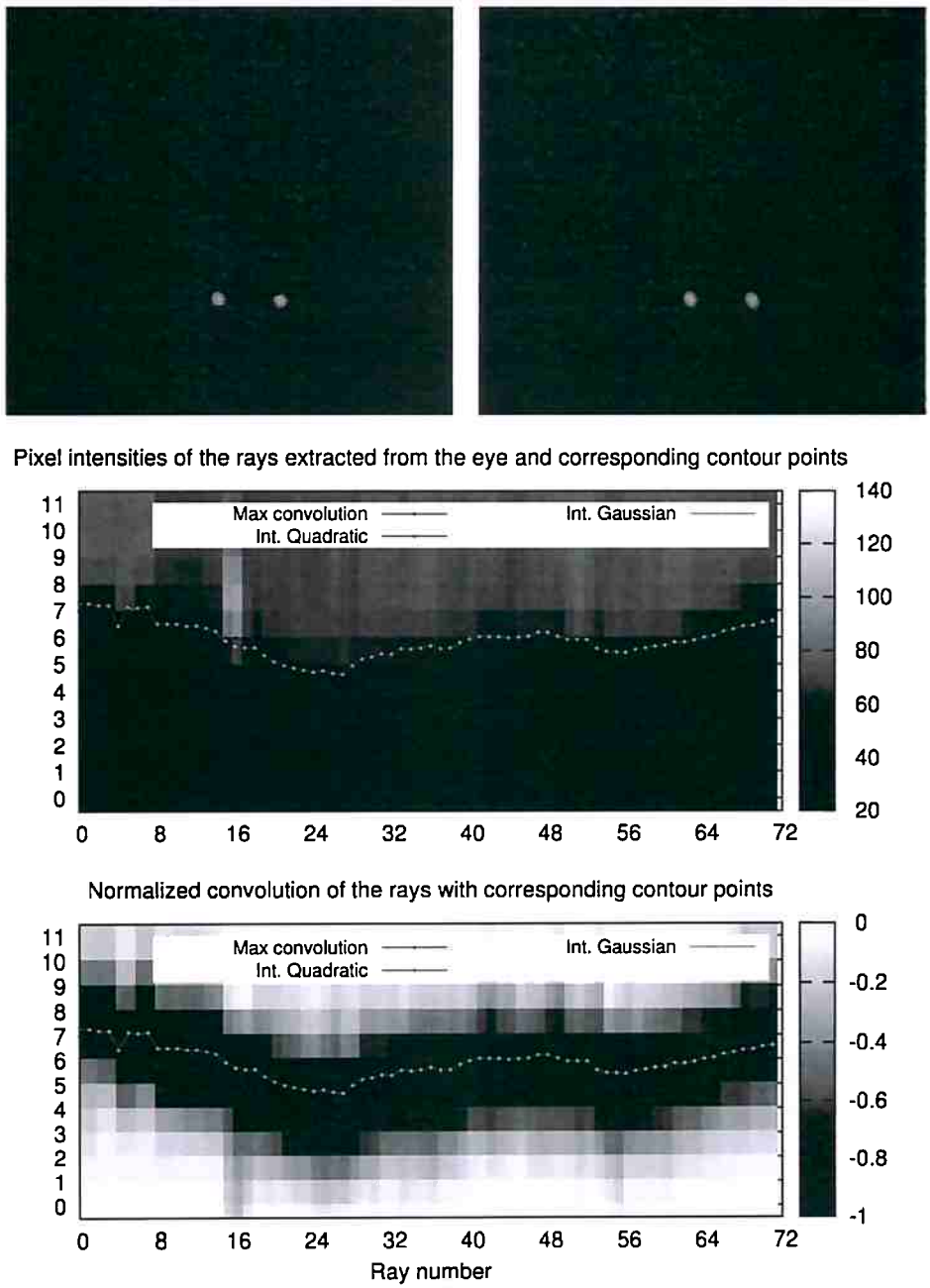


Figure 4.3: *The intensity profiles of each ray extracted from a dark pupil. A total of 72 rays were obtained, in which the red line represents the strongest filter response, while the green and blue represent the result of interpolation. (Middle) Rays resulting from bi-linear interpolation along the ray trajectory. (Bottom) Result of the convolution for each ray.*

The rays with the lowest weight are discarded. In the experiments reported, unless otherwise stated, 10% of the rays are discarded. There is a tradeoff in choosing this number. The rays associated with low weight are generally the ones in difficult positions, close to eyelashes, CRs, and so on, chances are that they are indeed wrong, but they also may be right and could, if not removed, contribute significantly to the correct placement of the pupil ellipse.

The sub-pixel location of the strongest response along each ray (using Eq. 4.4) is used as a boundary candidate. While some works deal with possible outliers using *RANdom SAmple Consensus* (RANSAC) (Fischler and Bolles, 1981) and augmented variations (Kassner *et al.*, 2014b;

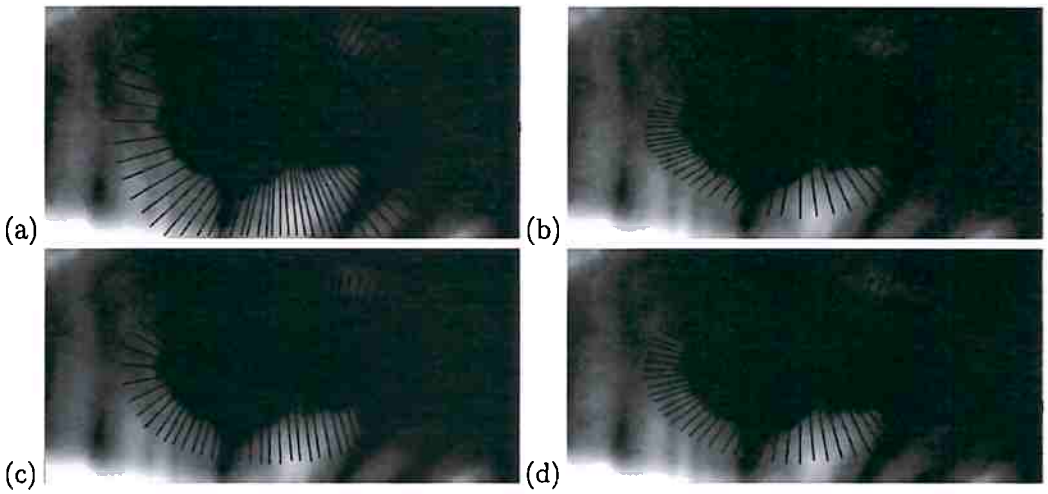


Figure 4.4: *Rays projected from an artificially introduced overlap region on a highly off-axis eye image taken from (Świrski et al., 2012). (a) Rays with a common origin at the overlap center and distributed evenly by angle; b) Rays with origin at the overlap center and passing through points sampled at evenly spaced eccentric anomaly angles c) Rays normal to ellipse e_i; d) Our approach.*

Li and Parkhurst, 2005; Świrski *et al.*, 2012), we tackle the problem using local statistics. The choice for RANSAC is justified when a considerable amount of outliers is known to be present (Starburst produce 17% outliers (Li and Parkhurst, 2005)). We take advantage of the fact that the pupil will be always an ellipse (small deviations considered), as well as the overlap (we enforce that by modeling the overlap as such[1]). The rays are traced from an ellipse and are expected to find an ellipse in their way out, as they cross the pupil-iris boundary (see Figure 4.2 and 4.3, the green line which join the peak responses is smooth). Therefore, the rate of change in the position of the peak among rays is expected to have only small variations. By removing rays responsible for steep changes, we aim at removing points not located on the true pupil border.

Assume we have an array C of the sub-pixel location of the peak in each ray and the associated angle of projection, the angle at which the ray was traced. Each position in this array is a candidate for the pupil border. The candidates are sorted by projection angle and a local standard deviation (LSD) is used to detect possibly wrong candidates. Candidates with high LSD are removed as outliers. The LSD can be calculated using convolution by box filters by noting the following equality

$$\sigma_L = \sqrt{E[(C - \mu_C)^2]} = \sqrt{E[C^2] - (E[C])^2} \tag{4.7}$$

where $E[.]$ is the expected value, i.e. the local mean, and is calculated by convolution $E[C] = C * K$, with K a box kernel of size 7. Note σ_L is an array with the same size as C. High local SD are detected by thresholding, as the candidates with values higher than the mean plus the SD over all LSD, i.e. $threshold = \mu_{\sigma_L} + \sigma_{\sigma_L}$. An ellipse is fitted to the remaining candidates based on the direct least squares method (Fitzgibbon *et al.*, 1999). Note we eliminate outliers without explicitly trying to fit an ellipse to the data.

Figure 4.5 shows a pupil partially covered by eyelashes. The same figure presents the candidates filtered by the threshold presented. In Figure 4.6 it is possible to note how filtering the candidates

[1]In fact, the overlap has the shape of an ellipse-to-ellipse intersection. The smaller the change in position from one frame to the next, closer to an ellipse the intersection will be. At high frame rates, modeling the overlap as an ellipse is a good compromise between performance and approximation error.

improve the performance, notably when the pupil border becomes harder to detect.

The performance of the refinement was evaluated using the dataset published by Świrski *et al.* (2012). It is composed by images taken from the left and right eye of two people, collected using a head-mounted system which produced highly off-axis images. The dataset includes a hand labeled ground-truth, i.e. pupil ellipse parameters, for 600 randomly selected images and was already used on the evaluation of other studies, such as Pupil (Kassner *et al.*, 2014b) and ExCuSe (Fuhl *et al.*, 2015b).

We compared the result of the refinement with the hand labeled images. The Hausdorff distance between the two ellipses was used. The Hausdorff distance finds the maximum Euclidean distance of any point in one ellipse to the closest point in the other. We implemented the distance metric with arbitrary precision using the algorithm in (Maisonobe, 2006), which finds the minimum distance between a point and an ellipse, and the Hooke-Jeeves direct search algorithm (Hooke and Jeeves, 1961) to maximize the distance (minimize −distance) by finding the optimal said point. The metric in the context of pupil ellipse fitting was introduced by Świrski *et al.* (2012) and, despite the pupil center being the relevant information for *point of gaze* (PoG) tracking (thus, the Euclidean distance between the centers would suffice), we think it more completely explain the quality of the refinement method.

Figure 4.6 shows a plot of the contour refinement rate compared against the ground truth data using the Hausdorff distance as error metric. Shown are the results without convolution peak interpolation, with quadratic and gaussian interpolation and with the presented LSD filtering. Just peaking the strongest response from convolution results in a markedly worse performance with easy to refine pupils. The quadratic and gaussian interpolations did not present any appreciable difference. The LSD presented the best results, particularly when the pupil border becomes harder to refine. Results from Starburst (Li and Parkhurst, 2005), ITU Gaze Tracker (San Agustin *et al.*, 2010), Pupil (without p1-right, the most challenging sequence) (Kassner *et al.*, 2014b) and the algorithm from Świrski(Świrski *et al.*, 2012) added for reference only, as they are doing the detection also.

Note the database only have dark pupil images, but our pupil contour refinement requires a pupil overlap from a previous, missing step. We simulate such overlap by perturbing the ground truth ellipse $e_{gt} = (a, b, c, \hat{\theta})$ as follows:

$$a' = a - a \cdot \tau \cdot |\mathcal{N}'| \tag{4.8a}$$

$$b' = b - b \cdot \tau \cdot |\mathcal{N}'| \tag{4.8b}$$

$$c' = (c_x + \mathcal{N}' \cdot (a - a')/4, c_y + \mathcal{N}' \cdot (b - b')/4) \tag{4.8c}$$

$$\hat{\theta}' = \hat{\theta} + \psi \cdot \mathcal{N}' \tag{4.8d}$$

such that $e_o = (a', b', c', \hat{\theta}')$, τ bound the maximum perturbation to the axes, ψ is the maximum perturbation to the ellipse orientation, and $\mathcal{N}'$ denotes a truncated normal distribution as follows

$$\mathcal{N}' = X \sim \mathcal{N}(0, 1) \quad s.t. \quad X \in [-1, 1] \tag{4.9}$$

About 68% of the values sampled will be inside the open interval $(-1, 1)$ and the other 32% will be either -1 or 1. In the experiments of this section we used $\tau = 10\%$ and $\psi = 5°$. Notice

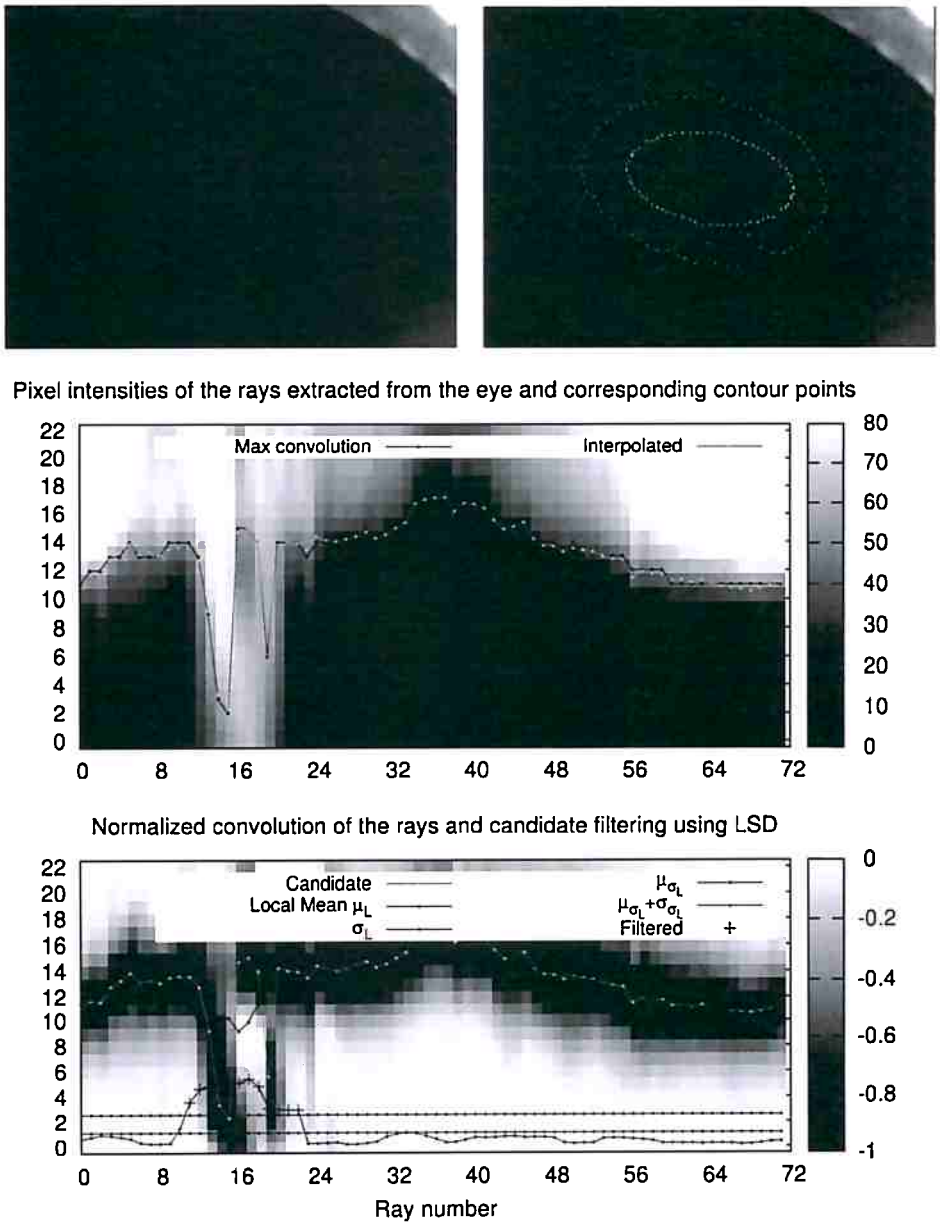


Figure 4.5: *The intensity profiles of each ray extracted from a dark pupil and corresponding convolutions. In bottom the result of filtering using local statistics.*

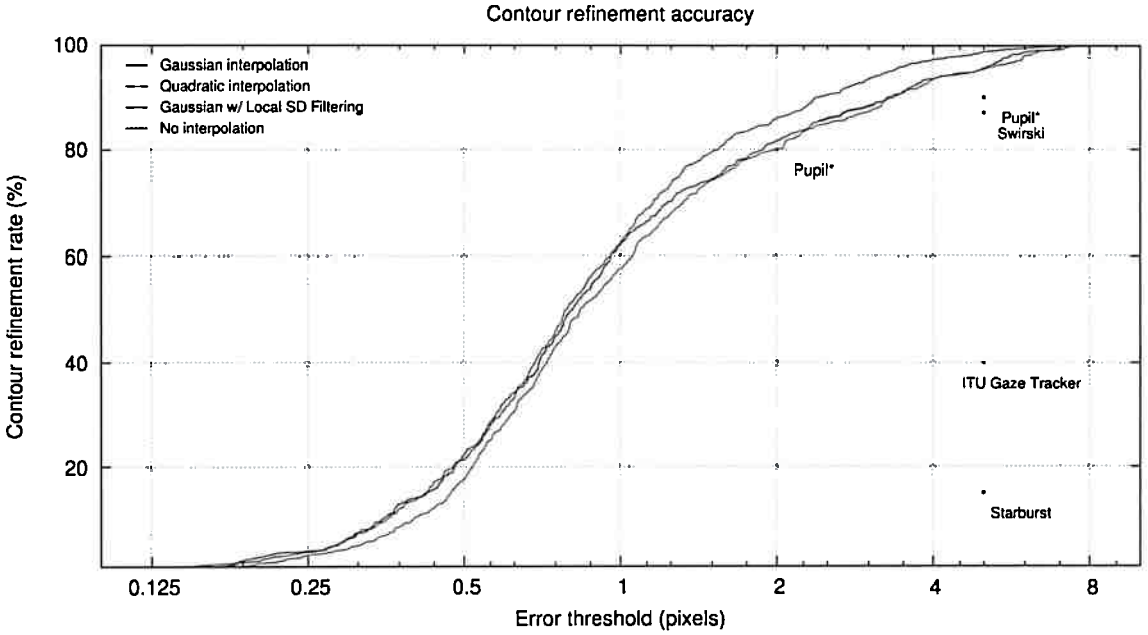


Figure 4.6: *Contour refinement rate using the discrete peak value, quadratic and gaussian interpolation and our filtering based on the local standard deviation.*

those parameters are plausible, as the overlap in real conditions will practically correspond to the pupil most of the time. In Section 4.3, we describe the Extended Temporal Support (ETS), which contributes significantly in approximating the overlap to the true pupil location.

We also tested the effect of varying the number of rays projected. Figure 4.7 shows the accuracy rate against the number of rays. It can be noted that increasing the number of rays from 6 to 20 causes a visible change in performance which reduces as the number of rays are increased.

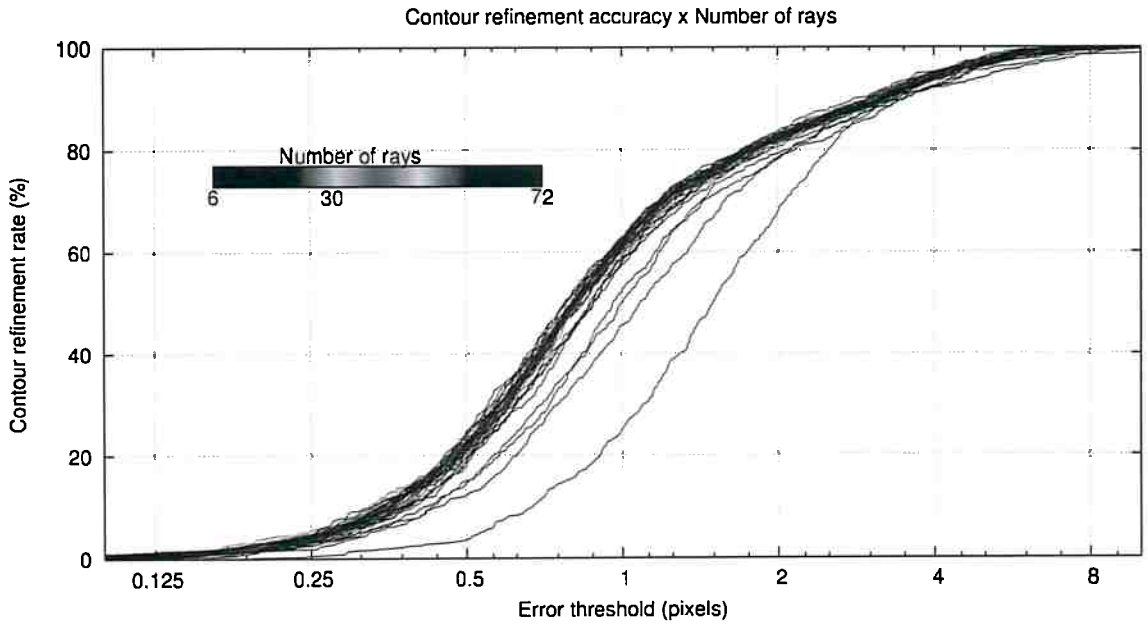


Figure 4.7: *Contour refinement accuracy rate versus ray number.*

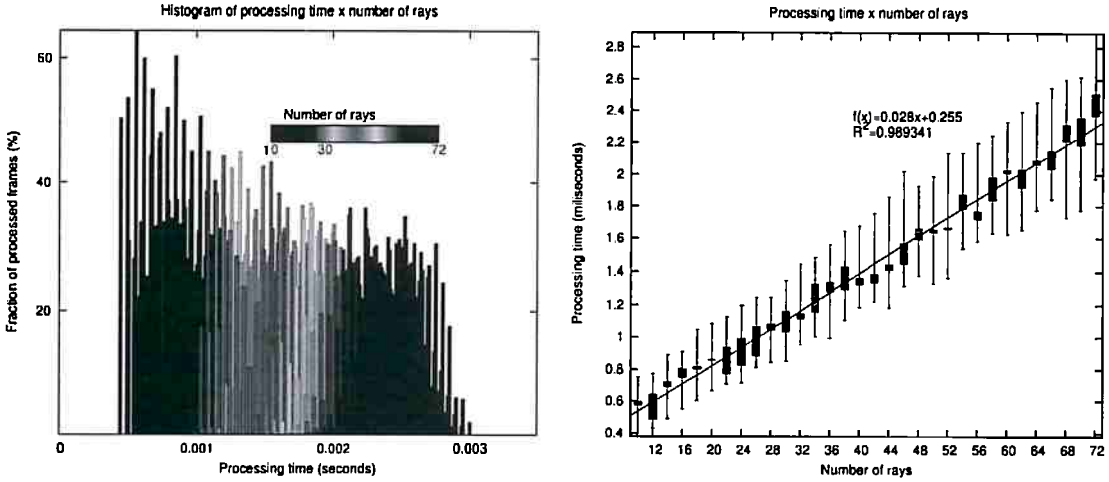


Figure 4.8: *Contour refinement processing time versus ray number on the data from Świrski et al. (2012). Processing time histograms for varying number of rays, on left, with the one of 30 rays highlighted. Box-and-whisker plot of the timings, right, with the best fitting line.*

4.2.1 ADSL pupil estimation

The strongest responses of an 1-d DoG filter are used to detect the actual pupil border. For bright pupils, the border is found around positive peaks, while for dark pupils on negative peaks. Now it might happen that the pupil present two colors at the same frame, and thus, keeping track of the current pupil color alone is not sufficient. The direction the stripe is moving and the position of the ray with respect to the stripe are important to determine the local pupil color. In addition to that, when the stripe pass over the pupil the color sequence is inverted, as can be observed in Figure 4.9, frame 102, which expects a bright pupil but it is mostly dark.

Using a relaxed logical notation, lets call the current pupil color as s and the opposite color as $\neg s$. Lets call $b \downarrow (.)$ and $b \uparrow (.)$ operators which receive the current pupil color and return the expected pupil color below and above the current stripe position[2], such that $b \downarrow (x) = \neg\hat{b}x$ and $b \uparrow (x) = \hat{b}x$, where $\hat{b}$ can be the logical negation or logical identity. Assuming we have a working tracker and the stripe is moving downwards, but it is still in a position higher than the pupil, the rays can assume the pupil color as s, thus $b \uparrow (s) = \neg s$ with $\hat{b} = \neg$. This condition can be maintained until the stripe crosses all the pupil. As soon as the stripe leaves the pupil, we can change the expected pupil color, such that $s = \neg s$, and accordingly, $\hat{b} = \neg\hat{b}$. This must be done because when the stripe leaves the eye ROI, the tracker resumes as no stripe was ever present (and then, the pupil color is s). When the stripe overflows, we must again do $\hat{b} = \neg\hat{b}$. The operation is similar for the case of a stripe moving upwards.

In the 2×2 design, managing the pupil color is similar except for a few points. The algorithm must keep track of the single colored pupils, which are used on the pupil candidate estimation. The refinement can proceed as usual for those frames and follow the same procedure just described for dual-colored pupils. The 2×2 design is particularly useful if reducing the strobe length (Δ_{strobe}) is a problem, as one colored pupils are guaranteed at half the frame rate.

Figure 4.9 shows the rays traced around the candidate pupil detected between consecutive frames. The images were taken at $187.326Hz$ with the strobe clock exactly 2% higher, producing a

[2]We assume a Cartesian coordinate system, and thus, the image origin is at a positive ordinate.

stripe that moves at more than five scanlines per frame.

4.3 Extended temporal support (ETS)

The pupil candidate is found from the intersection between the pupil on the current frame and the last frame processed. The search space, the region in which the rays are traced, also comes from the overlap and is modeled as a function of its size. However, the overlap is just a gross estimation of the pupil location, and not necessarily a good estimator for the pupil size, particularly during saccades.

In addition to that, experimental evidence shows that the mean fixation length during oral reading is 275ms, with a mean saccade length of 1.5°. During a scene perception task, the mean fixation length rises to 330ms, with a mean saccade size of 4° (Rayner, 1998). In terms of our experimental settings, this is equivalent to 51 and 61 frames, respectively, with minimal eye movements (nystagmus, drifts and microsaccades are present).

Considering the evidences about the eye movements, we concluded that the pupil overlap detection (Section 2.4) is not needed for all frames. Moreover, it is also possible to infer when it is needed from the accurate sub-pixel pupil estimation using a simple translation threshold. Therefore, we can save processing time by keeping track of the last pupil movements with respect to the camera (translation in image coordinates). For such, the *extended temporal support* (ETS) module holds information about four instants, the last bright and dark frames processed and, the bright and dark key frames. The data include the frame itself and the tracking data, such as the pupil and CR parameters.

The ETS comprises two steps which resembles a Kalman filter, being invoked in two different instants during the gaze estimation. As soon as a new frame arrives, called the *estimate* step, and after the gaze estimation is done, the *update* step. The former returns the candidate pupil estimate to the subsequent refinement. The later is responsible for updating the internal state of the ETS to better reflect the new data.

With the extended temporal support, the pupil candidate estimation based on image operations can be replaced by a simple arithmetic operation most of the time. During larger eye movements, the processing time for the candidate estimation is the same, but as the pupil position and size are better estimated, the pupil refinement can be done in a more constrained way (e.g. the size of the rays can be smaller).

4.3.1 ETS Update

The update begins just after the refined pupil position is found. The current frame image and tracking information (pupil and CR data) are saved for the next iteration. The last two frames are kept, one being the bright and one being the dark pupil representative. If one of the key frames is unpopulated, the corresponding last frame is used to fill up the position.

The keys can be replaced as needed. Two conditions trigger the change: a bad key was detected; and the overlap of the pupil from the current frame with the one of the key became too small. As the key frame is farther in time with respect to the last frame, it is expected that the distance between the pupil on current frame and the key frame be greater than the distance to the last

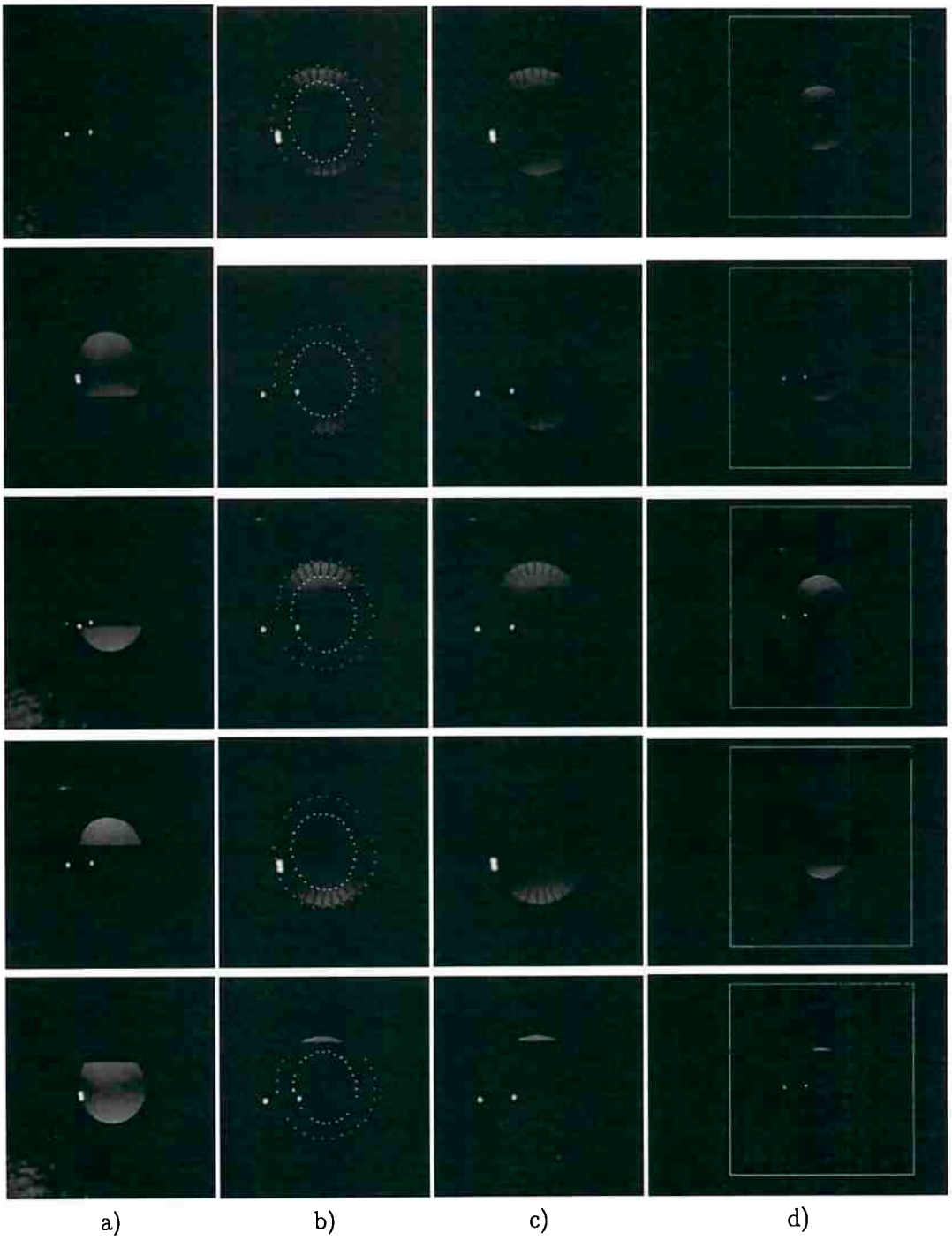


Figure 4.9: *Region of interest excerpts showing the pupil refinement for mixed color pupils. a) the previous frame; b) rays traced until intersecting the stripe; c) rays actually employed on the pupil border refinement; d) tracking result.*

frame. When the opposite occurs, we assume the key to be a bad representative of the pupil state and the last frame is used to replace it.

The other condition that causes the key to be replaced is a small overlap. If the intersection between the pupil of the current frame and the key becomes too small, the intersection might result in noisy data, as the influence of corneal reflections will become more evident at the overlap borders. We also try to avoid a condition in which the intersection becomes empty. A small overlap is assumed when the distance between the center of the current estimated pupil and the center of the pupil at the key become greater than half the current pupil minor axis.

During the update process, the keys might also be erased. This is done when the eye tracker is not tracking, for example, during the blink period.

4.3.2 ETS Estimate

If the key frames are not populated, the tracking must resume as usual, finding the overlap between the last and the current image frames. If otherwise, the key frames are present, the current pupil position is estimated in one of two ways, depending on the amount of change between the position of the pupil on the last frame to the key.

If the difference is lower than a threshold, η, the ETS reports the new candidate pupil center and size as the last one. Note the information used is not up to date, and no prediction about the real pupil position is made. Moreover, it also may happen that a new saccade has just begun, leading to a relative bad estimation result. It is normally assumed that saccadic movement is ballistic in nature (Robinson, 1964), i.e. characterized by high acceleration and deceleration. Our constrain here is that, even if the eye reached full speed during the frame interval, the pupil translation in image coordinates is inside the range covered by the rays (see the parameter selection Section 4.5 for the choice of v).

If the difference is higher than η, the same process used in Section 2.4 is employed, but instead of using the difference between the last frame and the current one, the former is replaced by the stored key. The result is the overlap between the pupil present on key and the pupil of the current frame. As the pupil center of the key is known, the current pupil center can be estimated by noting that the distance between the current pupil and the overlap is the same of the overlap to the key. The new pupil size is not calculated, as the last pupil size is returned as the estimation.

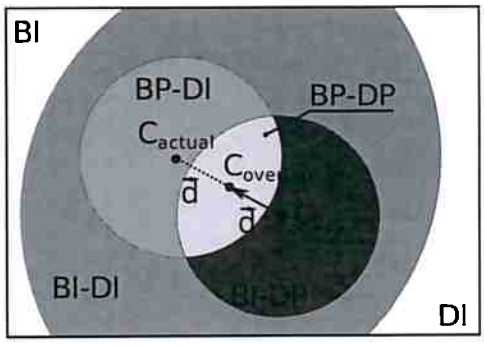


Figure 4.10: *How the ETS estimates the current pupil position. BI, BP, DI, DP stand for bright iris and pupil, and dark iris and pupil, respectively. Particular case for a bright pupil frame.*

Figure 4.10 shows how the new expected pupil position (C_{actual}) is found using the key information and the current frame. In particular, the current frame is a bright pupil image, and

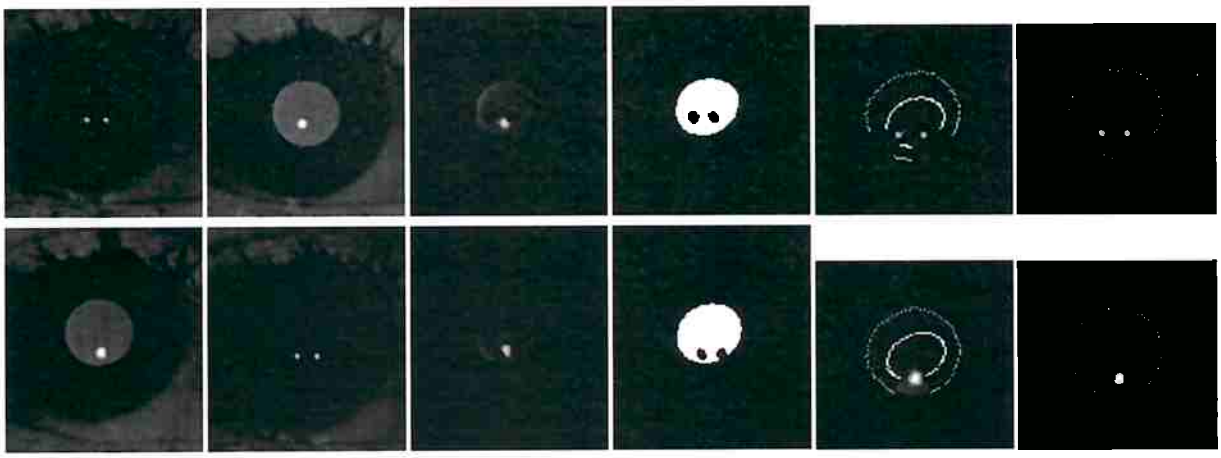


Figure 4.11: *Eye tracker intermediate products without ETS. a) Current frame; b) Last processed frame; c) Difference between current and last frames; d) Threshold of difference; e) Ray tracing, with the peak in convolution in green; f) Pupil candidate estimation, the overlap in this case (red) and refined pupil (blue). Data from USER1 (Section 5.3.1), where the current frame is 103 on top and 106 on bottom (a saccade is ongoing).*

therefore, the key used corresponds to a dark pupil image. The C_{key} is the dark pupil center from the key and is known, $C_{overlap}$ is calculated by finding the center of the overlap bounding box. Then, $\vec{d} = C_{overlap} - C_{key}$ can be found and used to estimate the current pupil center $C_{actual} = C_{overlap} + \vec{d}$.

When the eye tracker detects a condition of tracking failure, the ETS performs a reset, which includes all the key and last pupil and CR data. Therefore, it becomes non functional for a short period of time, in which the pupil candidate estimation is done as described in Section 2.4.

4.3.3 ETS estimated pupil quality

The purpose of the ETS module is twofold. It effectively reduces the mean processing time necessary for the pupil candidate estimation, and it provides a better pupil candidate estimation.

Figures 4.12 and 4.11 show the intermediate products of the eye tracker with and without ETS in the course of a saccade of USER1 (Section 5.3.1). To ease the comparison, the *eye tracker* (ET) was configured to process one frame each three. Without ETS, the overlap (Figure 4.11f, in red) is smaller than the true pupil and the projected rays barely reach the true pupil border. With ETS, the pupil candidate estimation almost matches the true pupil (the red and blue ellipses almost coincide in Figure 4.12).

The distance between the estimated pupil candidate and the refined pupil were also compared with both the ETS disabled and enabled. The Hausdorff distance was used as metric. Both Figure 4.13 and 4.14 show the first ten seconds of the tracking result of USER1 (Section 5.3.1). The pupil distance is given in pixels and the processing time in milliseconds. A paired t-test indicated that with the ETS enabled, the pupil distance ($M_{ETS} = 0.46, SD_{ETS} = 0.32$) was significantly smaller than the ones obtained with the ETS disabled ($M = 1.43, SD = 0.45$) for the frames analyzed ($t(1874) = 84.8, p < 0.001, 99\%CI[0.94, 0.99]$). With respect to the processing time, Section 6.1.1 provides the complete results for all subjects.

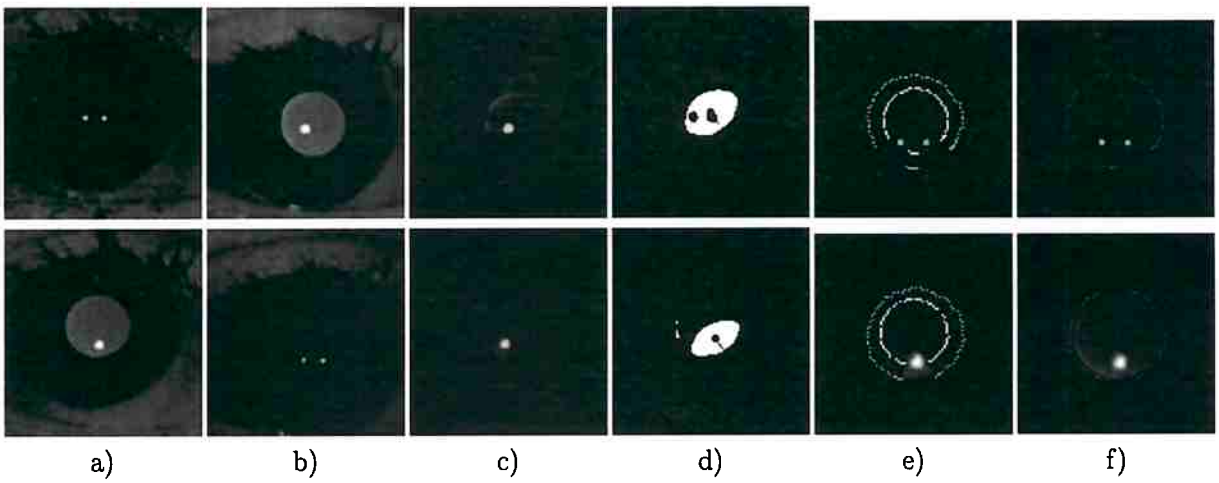


Figure 4.12: *Eye tracker intermediate products with ETS enabled. a) Current frame; b) Key frame; c) Difference between current and key frames; d) Threshold of difference and $\vec{d}$ in violet; e) Ray tracing, with the peak in convolution in green; f) Pupil candidate estimation (red) and refined pupil (blue). Data from USER1 (Section 5.3.1), where current frame is 103 on top and 106 on bottom (a saccade is ongoing). Key frame is 16 on top and 19 on bottom.*

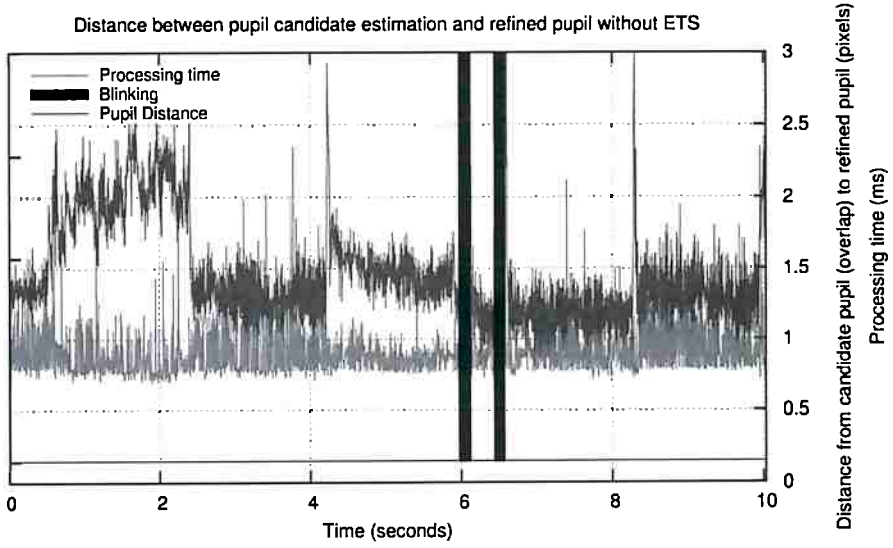


Figure 4.13: *Difference between the pupil candidate estimation and the refined pupil for the first 10 seconds of USER1 (Section 5.3.1).*

4.4 DSL-based eye tracker overview

The DSL-based ET incorporates the techniques presented to perform the pupil and CR position estimation in image coordinates to be further converted into PoG using calibration. Figure 4.15 depicts the flow employed by the ET. When a new frame is received, if the ET is already tracking, the ETS module is invoked to estimate the pupil position (the actual frame can, or not, be used in this step). If otherwise, the ET is not tracking, or in case the ETS could not perform the estimation, the pupil candidate comes from the difference between the frame just received and the last one (Section 2.4).

Next, the CRs are estimated (Section 4.1) and the pupil contour is refined using the peak in convolution along the trajectory of rays projecting outwards the candidate pupil (Section 4.2). If

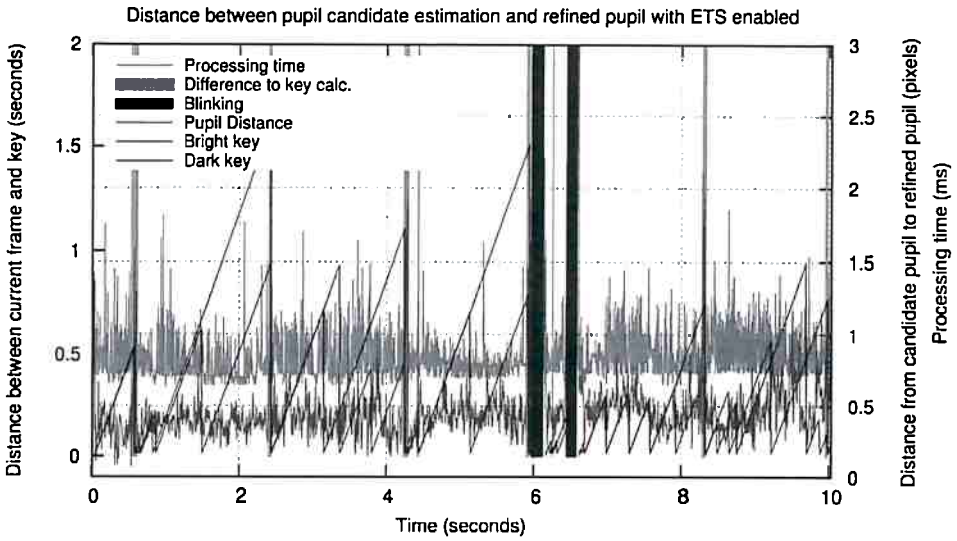


Figure 4.14: *Difference between the pupil candidate estimation and the refined pupil for the first 10 seconds of USER1 (Section 5.3.1) with the ETS enabled. The green areas represent the frames where the difference between the current frame and the key were done to calculate $\vec{d}$. For all other frames, the ETS just returned the last pupil as estimate.*

the refinement results in a good pupil, the ETS is updated with the new CR and pupil parameters (Section 4.3.1). If not, the ETS is reset, i.e., keys and last frames are emptied to give room for new tracking data.

4.5 Parameter selection

There are several parameters on the DSL technique. While most can be configured only once, fitting equally well many subjects, other might be more sensitive. In what follows we discuss the parameters and the problems associated with a wrong definition.

In Section 4.1, two parameters related to the CR estimation were defined, g_{min} and g_{max}. The higher g_{min}, faster the procedure to locate the CRs, as the threshold will start smaller. It might also be set to zero, with no expected loss in detection. The parameter g_{max} is not sensitive either, as it is not used as a success stop criterion. In the same section, a weighting constant ϱ is also defined. Its function is to lessen the weight of the found ROI on the CR position estimation. It is particularly important on the bright pupil frames, where the CR to background difference might be low. Note however, it might be responsible for at most a pixel error on the estimation, not impairing the detection independent of the chosen value.

In Section 2.4, the size of the ROI used on pupil detection is controlled by the $\aleph$ parameter. It is not sensitive, and can be set high such that the ROI matches the frame size, at the cost of more processing resources. With a value equal one, the resulting ROI will match the last pupil found, and is not adequate as it impairs the detection during saccades. We have used a fixed value of 3 in all experiments, which gives enough room for saccades at the experimental conditions.

In the same section, p_{min} and p_{max} were defined, with similar functions to the CR counterparts. The first, related to speeding up the pupil detection by bootstrapping the detector with a lower threshold. The second related to stopping the detection before reaching a zero threshold, in case of a wrong pupil color assumption. Those variables were designed to have a high margin of error,

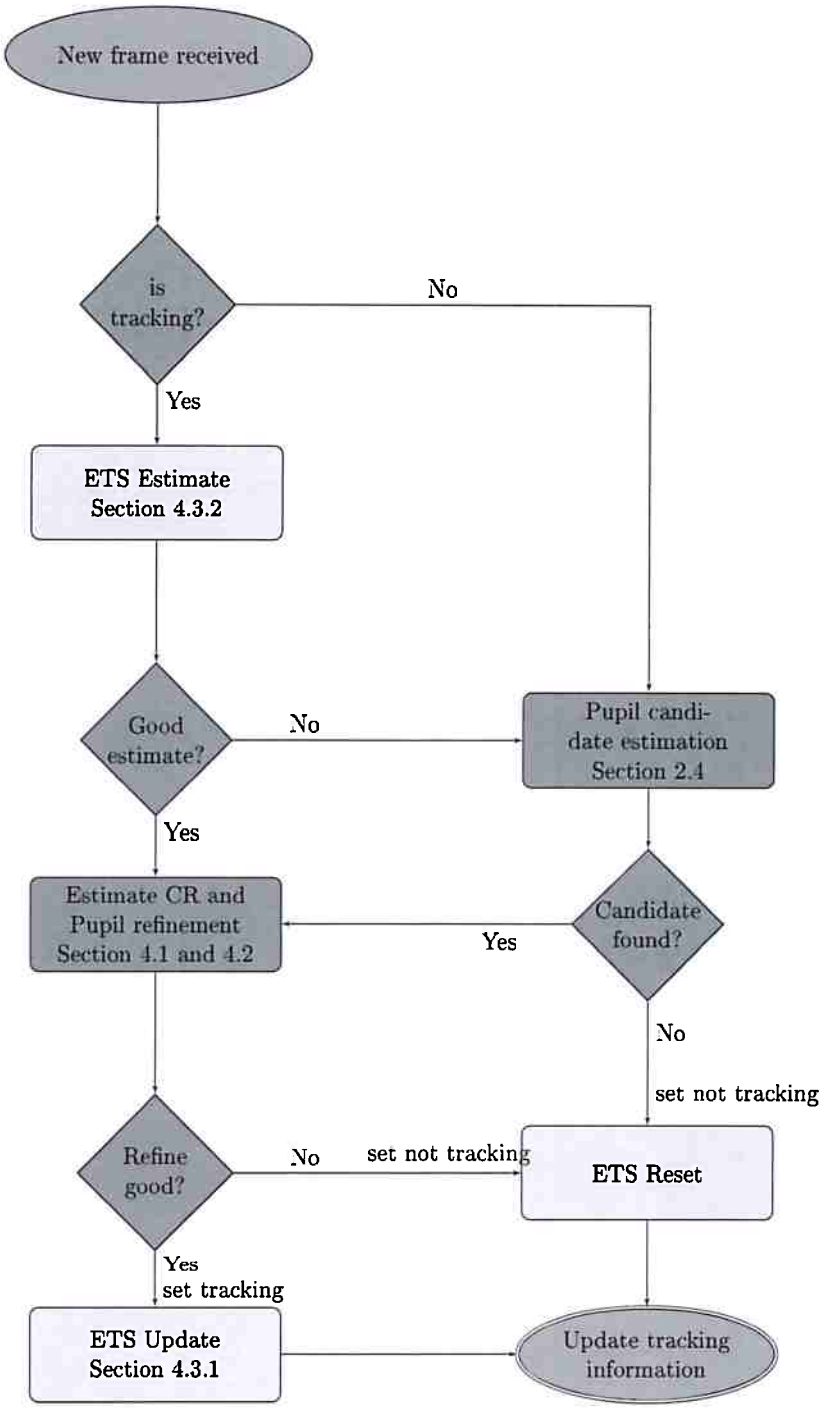


Figure 4.15: *Flowchart of the processing stages of the DSL eye tracker for a single frame.*

but particularly for our new dataset, they turned out to be very sensitive. They had to be set on a per subject basis, indeed. This happened because the contrast in the newly acquired images is low, for both the bright and dark pupil. Additionally, for some subjects, the bright pupil presented a strong intensity gradient, resulting in a poor pupil candidate estimation. Setting p_{min} to better reflect each subject data amended the problem.

The length of the projected rays during the refining stage of the pupil is defined by υ (described in Section 4.2). At first, it depends only on the frame rate. The higher the sampling, the lower υ can be as the overlap better represents the actual pupil position and size. The introduction of the ETS

allowed it to be even smaller, as the pupil candidate estimation is improved. In our experiments, unless otherwise stated, we used a value of $\upsilon = 0.30$.

In Section 4.3.2, there is a parameter η, representing the threshold in image pixels to elicit a difference calculation between the current frame and key. It is based on the translation of the last pupil estimated with respect to the key. Note that there is an interaction between the values of η and υ. If η is chosen big, υ must be chosen big enough to allow the rays to reach the actual pupil border starting from a small overlap. We have set a very conservative value of $\eta = 1.0$ pixel for all sections.

Chapter 5

Eye tracking experiment

In this chapter we describe the eye tracking experiments performed using both the *differential stroboscopic lighting* (DSL) and the *asynchronous differential stroboscopic lighting* (ADSL). The performance is compared against the Starburst (Li and Parkhurst, 2005) and three pupil detectors, namely the ExCuSe (Fuhl *et al.*, 2015a), ElSe (W. Fuhl, 2016) and the algorithm from Świrski (Świrski *et al.*, 2012).

5.1 Common experimental protocol

Each method was evaluated using a typical eye tracking experiment (Morimoto and Mimica, 2005a) that consists of showing a set of targets on the computer screen at known locations and measure the error distribution. A total of 35 targets were displayed, one at a time, approximately positioned on a 7×5 grid covering the whole surface of the screen. Each target was shown by about 2 seconds. target a small white cross hair presented on a black background.

Each volunteer was asked to sit comfortably at 70 cm from a 22" monitor, covering a field of view of approximately $37° \times 24°$. The volunteers were asked to keep a button pressed while blinking, and the two second interval was restarted after the release. To calibrate, 9 out of the 35 points were used to fit the parameters of a second order polynomial used to map the pupil-corneal reflection vector to a point on the computer screen.

The computer employed is equipped with an AMD Turion(tm) II P560 Dual-Core Processor with 2.5 Ghz and 6 GB of RAM. The operating system installed is a Debian Wheezy, 64-bit kernel version 3.2.65-1. The techniques were run in a per-frame round-robin fashion.

5.2 Performance evaluation

Next we present a brief description of the methods we have used to compare the performance.

The Starburst first detects features of the pupil boundaries, abrupt changes in intensity along projected rays, and then fits an ellipse to the feature locations using *RANdom SAmple Consensus* (RANSAC). Following is an image-aware model-based optimization used to improve the ellipse fitting. The algorithm also finds and remove the corneal reflection using interpolation (Li and Parkhurst, 2005).

ExCuSe is the short for Exclusive Curve Selector, and comprehend a technique based on edge filtering and oriented histograms calculated via the Angular Integral Projection Function (Fuhl *et al.*,

2015a). The pupil center is refined by an ellipse estimation similar to the one employed by Starburst (Li and Parkhurst, 2005).

ElSe is the short for Ellipse Selector, and is a method based on edge filtering followed by ellipse evaluation and pupil validation. As in ExCuSe, a Canny edge filter is applied to the image after normalization, followed by an evaluation of the connected edges in which its elliptic properties are analyzed. As the method always find an ellipse, a validation is performed as a last step to avoid returning a pupil position for a closed eye (W. Fuhl, 2016).

The last method used on the evaluation is an algorithm proposed by Świrski, which finds the coarse pupil position using a Haar-like filter. The intensity histogram of the coarse position is clustered using k-means followed by an image-augmented RANSAC ellipse fit (Świrski *et al.*, 2012).

All methods have a C or C++ source code available to download, thus easing performance comparisons. The Starburst implementation used is available online (Li and Parkhurst, 2005) in both C and Matlab versions. The C version lacks the optimization step which was added for completeness. The algorithms Starburst, ElSe and, ExCuSe, were run using default parameters, as provided on source. The algorithm from Świrski were run with default parameters also, except for the pupil minimum and maximum radii, for which the technique showed quite sensitive. This parameter was adjusted on a per user fashion to provide the best results (close radius to the observed were used).

The accuracy of each method is given by the mean error, and was calculated as the mean offset in pixels, and thereafter gaze angle, between the estimated and target points, based on the distance between the eye and the monitor plane. The precision is given by the standard deviation of the data of each user, and is equivalent to using the *root mean square* (RMS) of successive data points normalized by the mean. The time each frame took to be processed was also recorded, allowing the average processing time calculation of each method.

5.3 Differential stroboscopic lighting

5.3.1 Data collected at 187 Hz

Five people (2 female) from 30 to 57, average 37.4 years old, volunteered for the data collection (USER1-USER5). Each volunteer was asked to sit comfortably at about 60 cm away from a 22" monitor. Data was collected at 187 Hz using the head-mounted PS3 prototype described in Section E.1. The strobe length (Δ_{strobe}) was set to 800 μs. The subjects were asked to maintain their head as stable as possible while looking at the targets.

The Starburst and the DSL with and without *extended temporal support* (ETS) were compared to the ExCuSe (Fuhl *et al.*, 2015a), ElSe (W. Fuhl, 2016) and the algorithm from Świrski (Świrski *et al.*, 2012). The last three are pupil detection algorithms only, and thus, the corneal reflection position was gathered from the DSL. The techniques were run in a per-frame round-robin fashion, the DSL on all frames, and the others only on the dark pupil ones. The Starburst algorithm had to be changed to accommodate two corneal reflections, instead of one.

5.3.2 Data collected at 60 Hz with dual camera prototype

In Section E.2 we describe a prototype capable of capturing distinct images from the same perspective point, but using different light schemes. Thus, we can employ both stroboscopic and

Figure 5.1: *Example data captured using prototype from Section E.1, USER4, frames 50 to 54 (top).*

Figure 5.2: *Example capture of subject S3, frames 6102 to 6106, in which a saccade occurs. In the middle, capture using stroboscopic light at 850nm. On bottom, continuous 940nm illumination.*

continuous illuminators, being able to assess the performance of our approach against traditional dark-pupil counterparts without the need to change implementations to fit our data.

Ten people (4 female) from 30 to 59, average 36.2 years old, volunteered for the data collection. The head movement was restrained with the help of a head-rest, as shown in Figure E.5b.

The continuous illumination was provided by a single 940nm illuminator, and the images taken used to feed the Starburst, ExCuSe, ElSe, and the algorithm from Świrski. The pulsed illumination was provided by 850nm illuminators, and the images used to evaluate our technique. The strobe length (Δ_{strobe}) was set to 800 μs.

The synchrony between the cameras and the strobe illumination were assured by hardware. The cameras were setup using the same parameters, including the exposure and gain. The images were taken at $60Hz$ and QVGA format. A multi-threaded software was built to capture the images simultaneously, assigning a timestamp to each frame based on a global clock source. The instant of each target presentation and vanish were also saved to allow the fixation interval recovery.

The fixation sample was defined as the 300ms interval, 500ms before a target vanishing. Due to the size of our prototype, it had to be positioned considerably off-axis. For some volunteers, looking to a target at the top right corner of the monitor lead one of the dark corneal reflections to be on the limbal area, joining other reflections and resulting in a completely wrong *corneal reflection* (CR) midpoint. As we were not testing the prototype, but the method, such targets were removed from comparison on all methods for the given subject.

5.4 Asynchronous differential stroboscopic lighting

The data collections using the ADSL were done using the prototype described in Section E.3. It employs short strobes of just Δ_{strobe} =80 μs, which results in very small stripes.

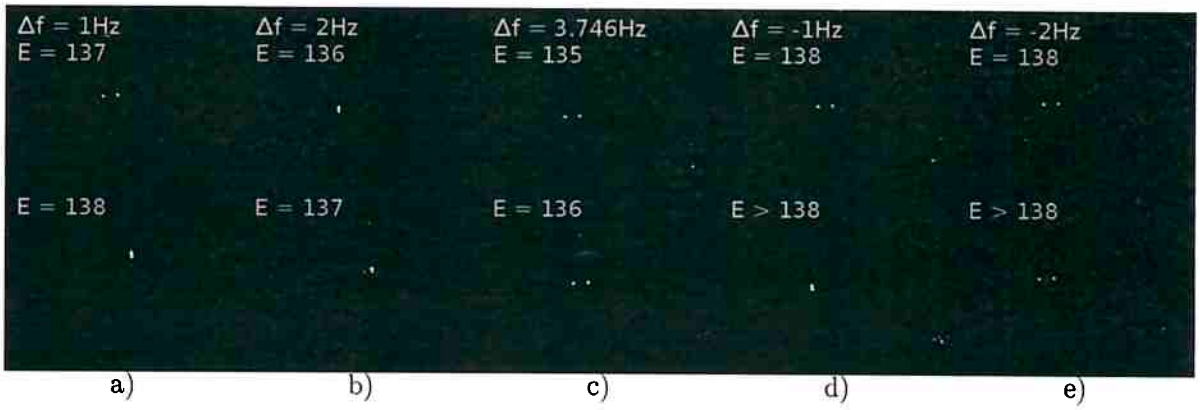


Figure 5.3: *Sample images captured at different external clock frequencies and corresponding exposures to produce a minimal stripe (top) and no stripe (bottom), when possible.*

5.4.1 Pilot experiment

A pilot experiment was done for five external clock frequencies and employing both designs (namely the 1×1 and 2×2), totaling ten trials (T1 to T10). Only one volunteer took part on the data collection. The head movement was restrained with the help of a chin-rest.

Data was collected at two external clock frequencies lower than the camera and three higher. The highest value employed was 2%, which at a camera sample rate of $187.3Hz$ gives approximately $191.0Hz$ ($+3.7Hz$). We bounded the tests to a small clock range because such differences are feasible even with analog circuits, as described in Section D.1.

For all data collection, we assumed $\mathbb{E}_{vis}$ and $\mathbb{E}_{inv}$ were provided. The values were found by visual inspection, following the procedure described in Section 3.4.1. In Figure 5.3, the E values shown correspond to $\mathbb{E}_{vis}$ on top, and $\mathbb{E}_{inv}$ on bottom images.

Figure 5.3 shows sample images for each external clock employed on the pilot experiment and the corresponding camera exposure. Note the exposure is a raw number for the PS3 Eye camera. For cases a), b) and c), increasing the exposure beyond the values presented on the bottom images would lead to a bright stripe, as discussed in Section 3.3 and depicted in Figure 3.4. For cases d) and e), the dark stripe cannot be canceled by an increase in exposure, as it is present even at exposures taking the whole frame interval.

5.4.2 Clock asynchrony of 2%

The pilot test suggests that using frequency differences on the order of 2% do not imply in a worse accuracy when compared to smaller differences (refer to Section 6.2.1). However, as the experiment was done with only one subject, the hypothesis of worse performance can not be statistically rejected. Nonetheless, assuming the best results are achieved in the synchronous scenario, even an analog circuit can be attuned to reduce the clock differences arbitrarily, by simply inserting a potentiometer on the external clock timing resistor network. Therefore, we consider 2% a reasonable worst case. As already discussed, the stripe can be minimized if the external clock is at a higher frequency. As it is easily controllable by the same means, we decided to perform an eye tracking experiment considering a fixed value of 2%.

The same protocol of the previous section (5.4.1) were performed with six subjects (3 male, average 37.5 years old, SD=10.7) which volunteered for the experiment. The same camera sample

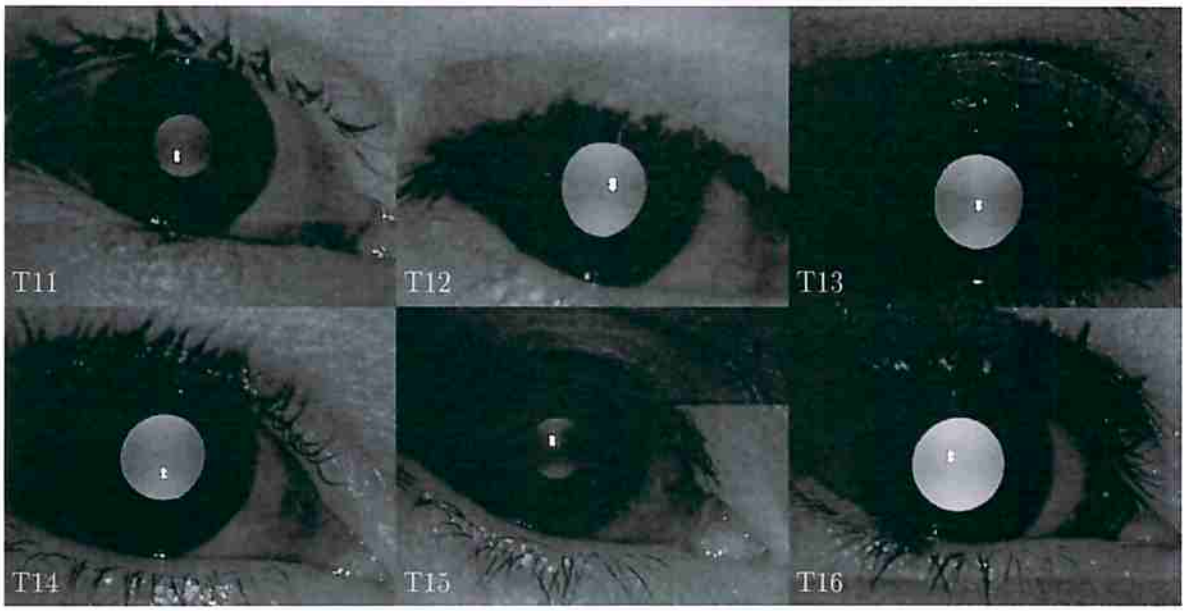


Figure 5.4: *Sample images from different subjects which volunteered for the eye tracking experiment. Over each image the corresponding trial they belong.*

rate of $187.326Hz$ were employed as well.

Figure 5.4 shows a bright pupil sample from each subject, to whom a trial number was associated (T11-T16).

The results are presented in the next chapter.

Chapter 6

Eye tracking experimental results

6.1 Differential stroboscopic lighting

6.1.1 Data collected at 187 Hz

The Starburst and the *differential stroboscopic lighting* (DSL) with and without *extended temporal support* (ETS) were compared to the ExCuSe (Fuhl *et al.*, 2015a), ElSe (W. Fuhl, 2016) and the algorithm from Świrski (Świrski *et al.*, 2012). The last three are pupil detection algorithms only, and thus, the corneal reflection position was gathered from the DSL. Table 6.1 summarizes the overall error of each algorithm.

An one-way between subjects *Analysis of Variance* (ANOVA) was conducted to compare the methods (Starburst, DSL, DSL with ETS, Excuse, ElSe and Świrski) on the gaze estimation error for all targets. The analysis was conducted on the log-transformed distance, to make it better fit the assumptions of normality and homoscedasticity. There was a significant difference on the gaze estimated for the six methods $[F(5, 877.3) = 8.26, p < 0.01]$. Post hoc comparisons using Tukey Contrasts indicated that the mean error were significantly different between all algorithms, except between the DSL with ETS ($M = 1.220, SD = 1.28$) and the method from Świrski ($M = 1.212, SD = 1.49$), between the ExCuSe ($M = 2.312, SD = 3.458$) and ElSe ($M = 2.442, SD = 4.132$), between Starburst ($M = 2.330, SD = 2.897$) and both ExCuSe and ElSe. The DSL ($M = 1.215, SD = 1.21$) did not significantly differ from the DSL augmented with the ETS as well.

The mean processing time per frame for each algorithm was also calculated. Two variations of the DSL with ETS were tested, with a change in the relative ray size. The DSL+ETS† was using

Volunteer	Starburst		DSL		DSL+ETS		ExCuSe		ElSe		Świrski	
	mean	SD	mean	SD	mean	SD	mean	SD	mean	SD	mean	SD
USER1	2.691	3.109	0.769	0.468	0.783	0.457	1.213	1.450	2.589	6.889	0.774	0.433
USER2	1.307	1.262	0.992	0.657	1.029	0.655	3.411	12.842	2.464	6.959	0.934	0.674
USER3	2.172	2.060	1.238	0.943	1.245	0.947	1.441	1.617	2.799	4.548	1.273	0.887
USER4	2.263	3.086	1.680	1.958	1.766	2.262	2.976	5.591	3.556	11.450	1.621	3.635
USER5	3.071	3.024	1.326	1.153	1.293	1.105	3.034	3.851	1.553	1.211	1.508	1.249

Table 6.1: *Mean error and standard deviation in degrees over the 35 targets. The proposed DSL and DSL+ETS are compared against the Starburst (Li and Parkhurst, 2005), ExCuSe (Fuhl et al., 2015a), ElSe (W. Fuhl, 2016) and the algorithm from Świrski (Świrski et al., 2012).*

Processing time			
Algorithm	Mean	SD	Realtime
DSL	1.864	0.392	100%
DSL+ETS†	1.933	0.560	99.7%
DSL+ETS‡	1.630	0.388	100%
Starburst	2.906	1.034	98.9%
ExCuSe	11.4969	2.485	0%
ElSe	34.023	4.853	0%
Świrski	112.6828	87.146	0%

Table 6.2: *Mean processing time and standard deviation in milliseconds per frame.* † $\upsilon = 0.30$. ‡ $\upsilon = 0.25$.

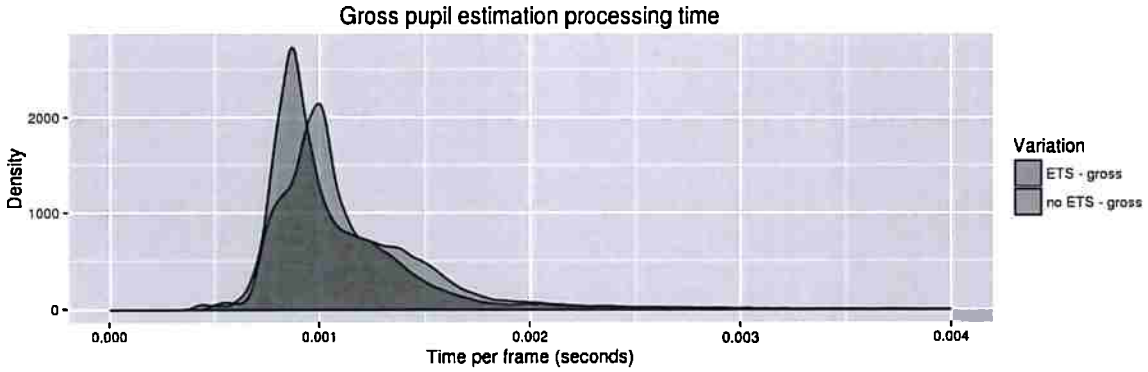


Figure 6.1: *Processing time of the pupil candidate estimation with and without ETS.*

$\upsilon = 0.30$ while DSL+ETS‡ was using $\upsilon = 0.25$. The time taken by the algorithms ExCuSe, ElSe and the one from Świrski do not include the *corneal reflection* (CR) processing time and they are also not processing the whole frame, as a *region of interest* (ROI) is given from the DSL. Table 6.2 shows the results and include a column which indicates the fraction of frames processed inside the $1/187s$ time window.

We further investigated the role of the ETS on the eye tracking performance. A paired Wilcoxon signed rank test was done and indicated that with the ETS enabled, the pupil candidate estimation time ($M_{ETS} = 1.014, SD_{ETS} = 0.295$) was significantly smaller than the ones obtained with the ETS disabled ($M = 1.109, SD = 0.296$) considering the data from all users ($p < 0.001$). It corresponds to a speedup of 9% when using the ETS with respect to the DSL algorithm without it. Figure 6.1 presents a density plot of the timings. However, as the candidate estimation now better represents the pupil (in size), the rays projected on the refinement phase are longer, resulting in an overall increase in processing time (compare the DSL with the DSL+ETS† on Table 6.2). Testing smaller rays ($\upsilon = 0.25$ instead of 0.30), resulted in an increase in performance (DSL+ETS‡ on Table 6.2) with a small impact on accuracy (0.041°).

6.1.2 Data collected at 60 Hz with dual camera prototype

Table 6.3 summarizes the overall error of each algorithm for each volunteer. The CR coordinates for the ExCuSe, ElSe and Świrski, come from the Starburst algorithm. We also added the overall error if instead, the CR as calculated from the DSL was used. In addition, the table also presents the DSL result for the half bright, and half dark pupil frames. The results were obtained using

Volunteer	Starburst		ExCuSe		ElSe		Świrski		DSL		DSL*		DSL**	
	mean	SD	mean	SD	mean	SD	mean	SD	mean	SD	mean	SD	mean	SD
S1	1.40	1.23	1.30	1.10	2.82	1.47	0.82	0.57	0.63	0.35	0.54	0.33	0.58	0.33
S2	1.15	0.93	4.31	1.24	4.61	0.86	0.69	0.52	0.83	0.67	0.74	0.64	0.80	0.60
S3	1.02	0.77	3.60	1.44	3.06	1.69	0.65	0.48	0.49	0.49	0.42	0.43	0.43	0.46
S4	0.94	0.56	2.59	1.43	4.20	1.29	0.74	0.38	0.60	0.34	0.52	0.29	0.54	0.28
S5	0.87	0.62	3.04	1.39	2.93	1.49	0.77	0.79	0.63	0.36	0.55	0.31	0.47	0.31
S6	1.54	1.17	2.66	1.51	3.57	1.60	1.56	1.31	0.77	0.47	0.76	0.52	0.59	0.38
S7	1.65	1.29	4.37	1.20	4.53	1.03	0.97	0.65	1.10	1.08	1.01	1.11	1.19	1.07
S8	1.41	1.02	1.28	1.06	2.15	1.47	4.38	1.15	0.67	0.49	0.54	0.33	0.74	0.67
S9	1.09	0.89	3.19	1.46	2.04	1.56	0.67	0.33	0.59	0.41	0.52	0.35	0.57	0.49
S10	1.50	1.01	1.43	1.10	3.80	1.34	0.92	0.53	0.83	0.68	0.71	0.64	0.67	0.71
Overall†	1.25	1.01	2.77	1.70	3.37	1.65	1.19	1.29						
Overall‡			2.64	1.76	3.31	1.67	1.11	1.34	0.71	0.59	0.63	0.56	0.65	0.60

Table 6.3: *Mean error and standard deviation in degrees over the 35 targets for all subjects. † CR from Starburst. ‡ CR from DSL. * bright pupil only frames. ** dark pupil only frames.*

	Processing time			
Algorithm	Mean	SD	Realtime†	Realtime‡
Starburst	5.891	3.537	100%	43.1%
ExCuSe	16.186	4.503	39.8%	0%
ElSe	45.177	4.974	0%	0%
Świrski	96.486	66.828	0%	0%
DSL	2.340	0.875	100%	99.5%

Table 6.4: *Mean processing time and standard deviation in milliseconds per frame. † Frames processed in 1/60s. ‡ Frames processed in 1/187s.*

different polynomial coefficients calculated for each pupil color.

In addition to the tabular data, we also added the error density plot for each method (Figure 6.2), as the distribution deviates considerably from normality, being positively skewed and with a right long tail. Figure 6.3 shows the error distribution for each technique considering data from all subjects.

An one-way between subjects ANOVA was conducted to compare the methods on the gaze estimation error for all targets. The analysis was conducted on the log-transformed distance, to make it better fit the assumptions of normality and homogeneity of variances. There was a significant difference on the gaze estimated for the five methods $[F(4, 16984) = 21.8, p < 0.000]$. Post hoc comparisons using Tukey Contrasts indicated that the mean error were significantly different between all algorithms, except between the pairs: (DSL, Starburst, $p = 0.15$), (DSL, Świrski, $p = 0.46$), (Starburst, Świrski, $p = 0.975$) and, (ExCuSe, ElSe, $p = 0.60$).

The time each method took to process each frame was also recorded. Table 6.4 summarizes the overall processing time.

From Figure 6.4 we can note the processing time is not normally distributed. The times from ElSe and ExCuSe present a bimodality which avoids any simple data transformation to allow parametric statistical tests. The number of data points do not vary between methods, but vary between subjects (as some targets were removed due to non-characteristic CR), resulting in an

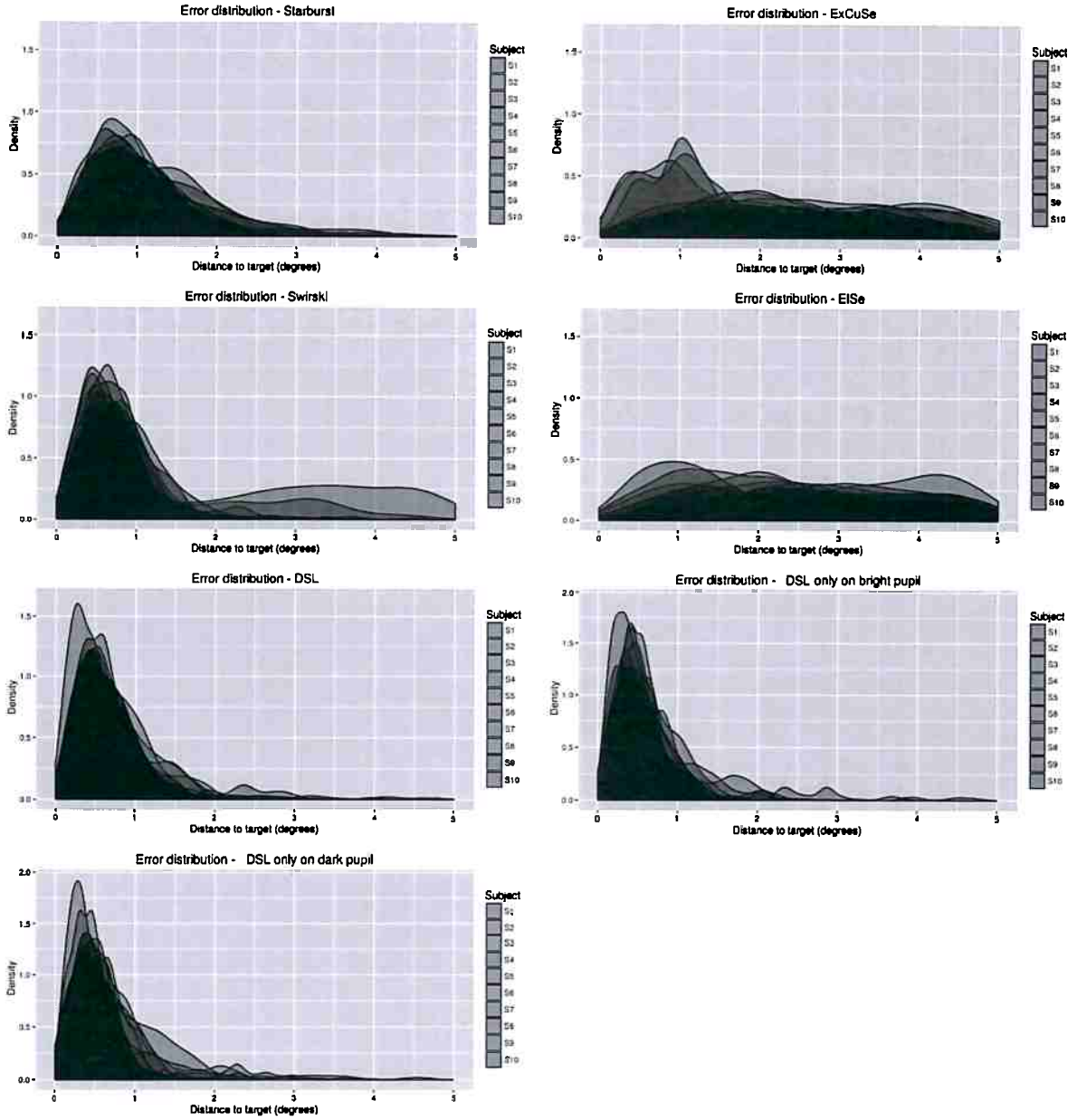


Figure 6.2: *Error distribution per subject.*

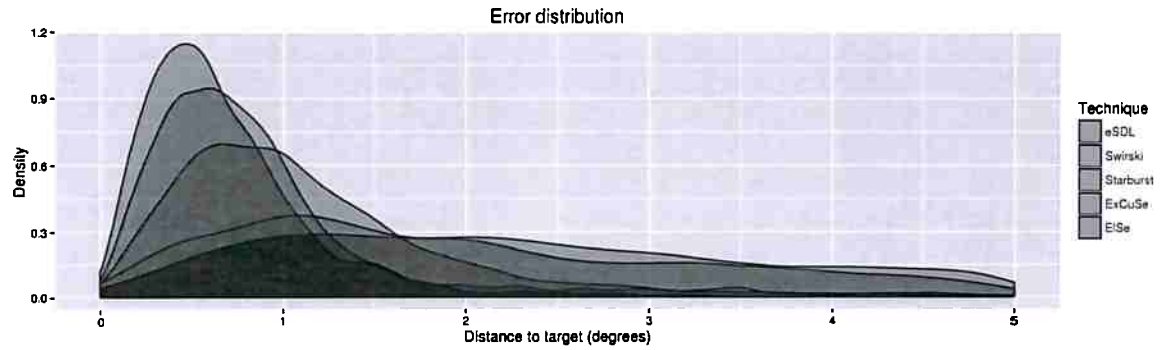


Figure 6.3: *Error distribution per technique for all subjects.*

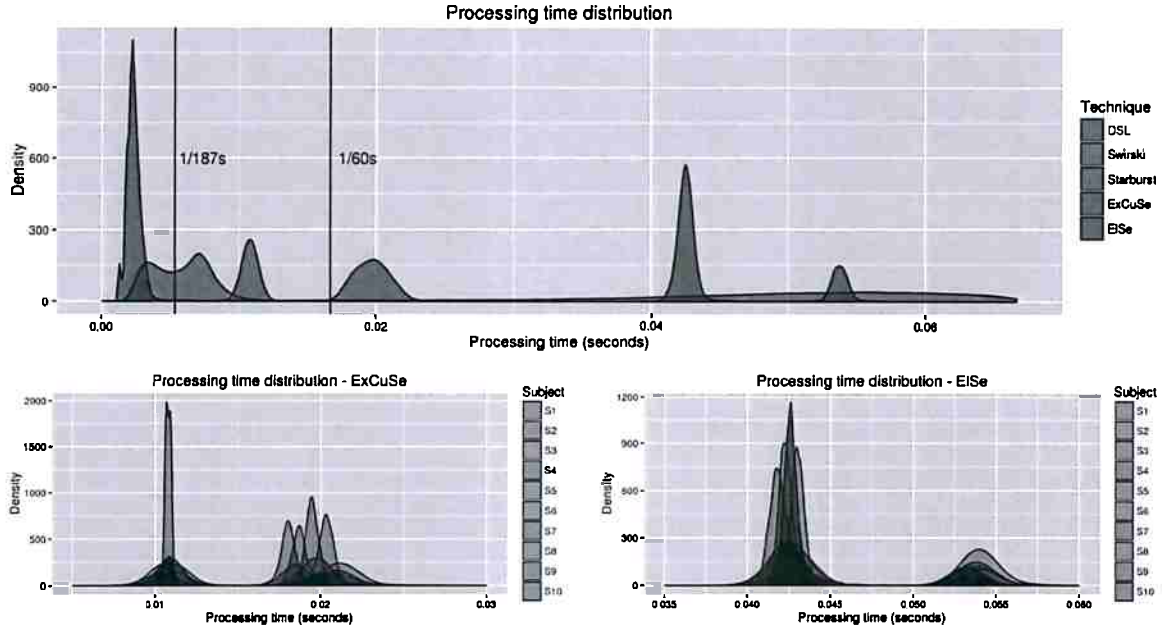


Figure 6.4: *Frame processing time distribution per technique for all subjects (top). Frame processing time per subject for ExCuSe and ElSe (bottom).*

incomplete block design. Thus, non-parametric randomized block analysis of variance are also non-applicable.

Figure 6.4(bottom) reveals that the source of bimodality did not come from joining different performing subjects. Particularly for ExCuSe, the distinct processing times come from a decision point after image normalization and histogram analysis. Depending if a peak is detected or not, the method follows different processing steps which take distinct processing times (Fuhl *et al.*, 2015a). The case is similar for ElSe, if after calculating and filtering edges an ellipse is not found, a completely different processing flow is followed (W. Fuhl, 2016).

6.1.3 Discussion

The mean error of the Starburst is greater than found on literature (Li and Parkhurst, 2005). We attribute such in part to the difference in image size, as we employ a QVGA frame against a VGA format of the original work. Moreover, to detect the pupil border, the algorithm utilizes a step and a threshold parameter. The default values were used as they showed best results in the original work. However, no studies were made to assess the adequacy to our setup. The number of rays used was ten, as starting from five no appreciable differences were demonstrated (Li and Parkhurst, 2005).

The techniques from Świrski, ExCuSe and ElSe are pupil detectors, and do not include CR processing. The CR was initially gathered from the Starburst. As both the calibration procedure and the subsequent error estimation use pupil-CR vectors, we investigated if the different CR estimation played a role on results. The calibration polynomials and the error estimation were redone using the CR as calculated by the DSL, presenting similar results (see the overall results on Table 6.3).

The illuminator responsible for the bright pupil was arranged at the middle point, between the two dark pupil illuminators. This arrangement was chosen to allow the use of the dark pupil CR

midpoint interchangeably with the bright corneal reflection, assuming the eye lens to be a perfect spherical cap. However, this assumption did not hold. When the corneal reflection approximates the limbal area, the lens curvature change, and hence the reflex relative position. As a result, despite the stable pupil estimation, the calibration is negatively affected by diverging corneal reflection positions on adjacent frames for some of the targets. Table 6.3 presents the results of running the calibration and error estimation for the bright and dark pupil frames separated. Note that both result in a lower error as compared to using all frames, at the cost of half the frame rate.

A prototype was constructed to allow an objective comparison to other techniques. The prototype is described in Section E.2. On the first data collection (Section 6.1.1), our technique were run on all frames, while the competing techniques were run on half of them (the dark pupil ones). However, it had been shown that *root mean square* (RMS)-based precision is affected by the sampling rate (Blignaut and Beelders, 2012). Additionally, running the other techniques on the dark pupil frames exposed for a short period, benefit them from a better image quality they generally do not count on.

Compared to the data from the first data collection (Section 6.1.1), the dual-camera prototype produces eye images about 32% bigger, i.e., the field of view is smaller (cover a rectangle $28 \times 21mm$ at the focal distance). However, the images produced have lower contrast, partly due to the beam splitter and filters, and partly due to the bright pupil illuminator, which causes some shadowing on the frame. This effect can be seem on Figure 5.2.

With respect to the processing time, the DSL was consistently faster than the other methods, while presenting a competing performance. Only 0.5% of the frames were not processed in the $1/187s$ time window. From those, the great majority correspond to blink intervals, which controversially, are the most time consuming. A similar behavior is reported by Hennessey *et al.* (2006). This happens because we bootstrap the pupil differencing iterative procedure with a value taken from the last refined pupil. If no pupil candidate is found, the bright and dark pupil frames are inverted and the processing of the corresponding frame restarted. This is necessary because in case of a processing or input-output demand surge, frames might get dropped, inverting the actual sequence.

We were not able to analyze the timings statistically, but the density plots of Figure 6.4 are quite clear. The closest in performance to the DSL is the Starburst, which was able to process frames at competing times. However, the variance on the error to targets is higher, despite not statistically significant. The other techniques took much longer to process each frame.

6.1.4 Limitations

A very important assumption of the method is that the sampling must be high enough to guarantee the overlap between the pupil on adjacent frames, even at peak velocities during saccades.

The presented method is partially robust to outliers when refining the pupil border. While it might perform well for rejecting features generated by eyelashes, the same does not occur for occlusions caused, for example, by the eyelid. This behavior is easily explained by the way outliers on the pupil border are detected. Our approach is based on local statistics, and assumes the distance of the overlap to the true pupil border varies at a slow rate. As it is much like an outlier detector, it easily finds points not belonging to the pupil border if they are few, isolated along the border. Otherwise, the variance increases reducing the power of the method, rendering a situation where no point might be filtered at all.

Mean error on all 35 targets

Trial	Δ_f	Mean	SD	Median	Total frames
T1	-2Hz	0.71	0.60	0.58	1995
T2	-1Hz	1.01	0.70	0.92	1995
T3	1Hz	0.75	0.87	0.60	1995
T4	2Hz	1.02	0.71	0.88	2004
T5	2%	0.73	1.38	0.48	1995
1×1		0.84	0.91	0.66	9984
T6	-2Hz	0.56	0.78	0.48	1995
T7	-1Hz	0.75	1.11	0.59	1995
T8	1Hz	0.93	0.63	0.91	1995
T9	2Hz	0.97	1.27	0.63	1995
T10	2%	0.65	0.37	0.61	1995
2×2		0.77	0.91	0.62	9975

Table 6.5: *Mean error and standard deviation in degrees over the 35 targets for all trials. The median and total frames considered are also presented.*

A simple solution is to simply drop feature points on rays pointing towards the upper and bottom portions of the eye (Ryan *et al.*, 2008b). While it might be acceptable when filtering points at the limbus, for the pupil, it throws good candidate points away most of the time. Another approach, which we let as a future improvement, is to model the eyelids using active contours (Kass *et al.*, 1988), as done by Ryan *et al.* (2008b).

The ETS was designed to give a better temporal support, but it was only partially achieved. An error in pupil estimation might be propagated (for particular v values), and no restrictions or evaluations are applied to identify a poor pupil based on previous data. With respect to the evaluation of the technique, the ETS is still to be evaluated appropriately.

As any IR-based method, ours suffer from outdoors light. Despite we have not measured how the performance is affected, a great impact is expected. Our system was built on top of the assumption of a rolling shutter mechanism, which for being able to capture a pulse of light all over the frame, relies on the exposure being set close to maximum. An external IR-rich illumination might not only degrade the image quality (bring back motion blur and skew), but can also saturate portions of the image, which can result in tracking impairment. It is common sense to employ bandpass filters on such systems, but their effects were not assessed on this work.

Other solutions to such cases do exist. One might simply rely on a hybrid system which switches to another algorithm, turning off the bright pupil illuminator and reducing the exposure, as suggested by Ryan *et al.* (2008a).

6.2 Asynchronous differential stroboscopic lighting

6.2.1 Pilot experiment

Table 6.5 summarizes the overall error of each trial. The accuracy is given by the mean error and the precision by the standard deviation.

In addition to the tabular data, we also added the error density plot for each external clock frequency (Figures 6.5 and 6.6), as the distribution deviates considerably from normality, being

positively skewed and with a right long tail. Figure 6.7 shows the error distribution for each design considering data from all frequencies.

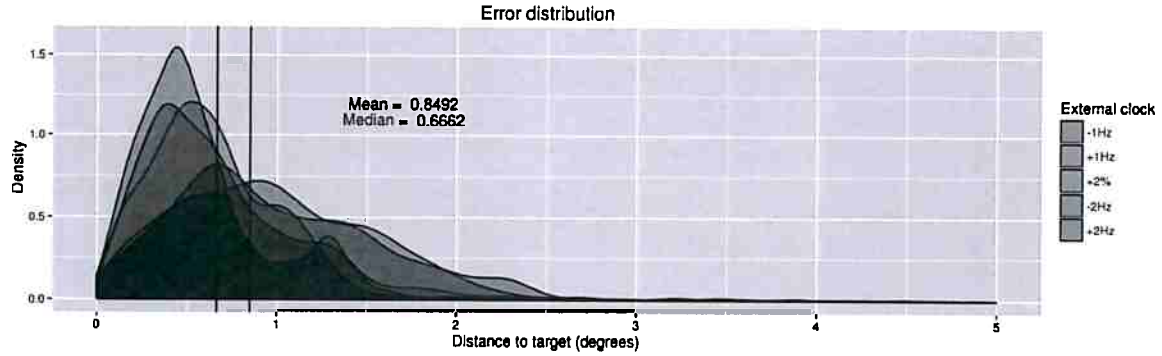


Figure 6.5: *Error density distribution in degrees for all targets for the* 1×1 *design.*

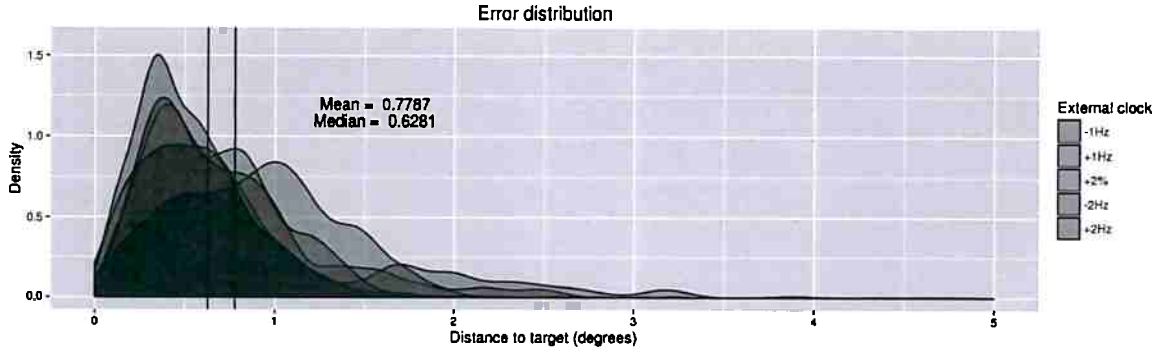


Figure 6.6: *Error density distribution in degrees for all targets for the* 2×2 *design.*

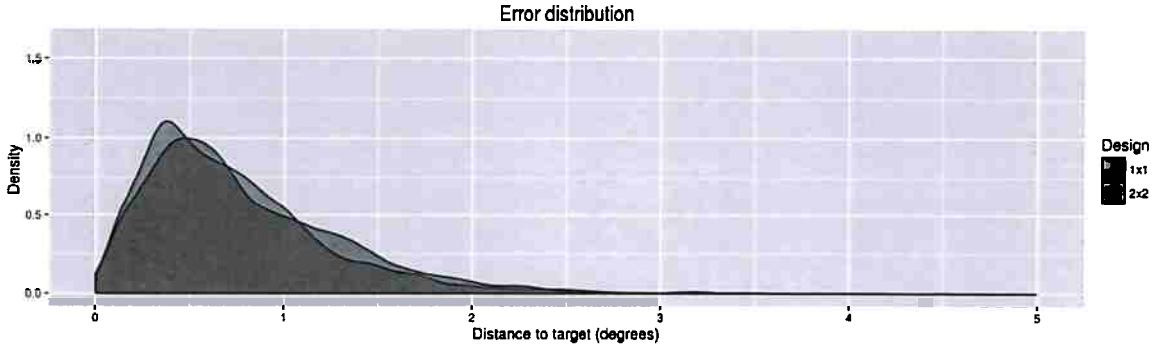


Figure 6.7: *Consolidated error density distribution in degrees for* 1×1 *and* 2×2 *designs.*

The time to process each frame for each trial was also recorded. Table 6.6 summarizes the overall processing time. Considering that in the 2×2 design the pupil candidate estimation is not done for some frames (particularly when the stripe is passing over the pupil, for mixed color pupils), it is expected the mean processing time to be slightly lower. We also recorded the pupil diameter, as the *region of interest* used to trace the rays to refine the pupil and the search space for CRs is based on such (see Figure 4.1 for a comparison of the ROI). Note that even with a mean pupil size slightly bigger, the 2×2 design performed faster.

		Processing time				
Trial	Δ_f	Mean	SD	Pupil dia.	Total frames	Realtime†
T1	-2Hz	2.131	0.489	46.5	15207	15177
T2	-1Hz	2.222	0.541	48.7	15306	15263
T3	1Hz	2.226	0.557	48.0	15061	15003
T4	2Hz	2.080	0.417	48.8	15261	15240
T5	2%	2.293	0.358	62.7	15650	15630
1×1		2.191	0.483	51.0	76485	76313
T6	-2Hz	2.057	0.462	53.0	15714	15698
T7	-1Hz	2.016	0.440	56.4	15083	15071
T8	1Hz	2.286	0.438	70.4	15296	15271
T9	2Hz	2.003	0.453	52.9	15383	15365
T10	2%	2.047	0.425	53.1	15121	15109
2×2		2.082	0.459	57.1	76597	76514

Table 6.6: *Mean processing time and standard deviation in milliseconds per frame. Mean pupil diameter and total frames processed also shown. † Frames processed in 1/187s.*

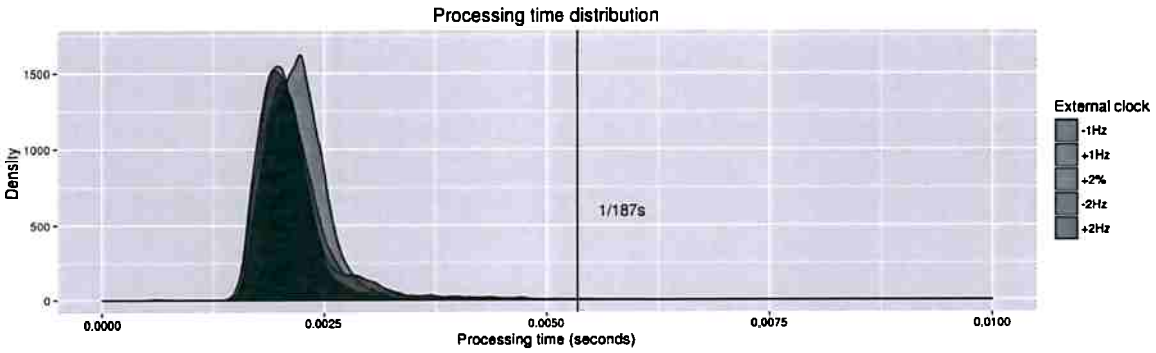


Figure 6.8: *Processing time density distribution for the 1x1 design.*

Figure 6.9: *Processing time density distribution for the 2x2 design.*

6.2.2 Clock asynchrony of 2%

Table 6.7 summarizes the overall error for each subject. The accuracy is given by the mean error and the precision by the standard deviation. The error was estimated three times, considering all frames and each pupil color isolated.

We have also included the total processing time, from the pupil candidate estimation to the pupil refinement. Table 6.8 summarizes the results. Only 88 frames took longer than $1/187s$ to be

Mean error on all 35 targets

	Both CR			Bright CR		Dark CR		
Trial	Mean	SD	Median	Mean	SD	Mean	SD	Total frames
T11	0.84	1.30	0.57	0.83	1.13	0.93	1.37	1938
T12	0.93	0.90	0.92	0.89	0.84	0.97	0.98	1881
T13	0.55	0.54	0.60	0.57	0.59	0.51	0.43	1995
T14	0.71	0.73	0.88	0.70	0.75	0.72	0.68	1938
T15	0.84	1.05	0.48	0.84	1.02	0.80	1.13	2015
T16	0.68	1.11	0.48	0.79	1.20	0.66	0.78	1992
Overall	0.76	0.98	0.58	0.77	0.95	0.76	0.96	11759

Table 6.7: *Mean error and standard deviation in degrees over the 35 targets for all trials. The median and total frames considered are also presented. The results of running the calibration and error estimation on a particular pupil color also shown.*

Processing time

Trial	Mean	SD	Pupil dia.	Total frames	Realtime†
T11	2.017	0.455	49.7	15764	15755
T12	2.260	0.390	70.7	15601	15593
T13	2.336	0.363	72.2	15024	15015
T14	2.234	0.379	70.4	15326	15300
T15	1.998	0.406	45.9	14888	14871
T16	2.257	0.388	76.3	15962	15943
Overall	2.186	0.418	64.4	93565	93477

Table 6.8: *Mean processing time and standard deviation in milliseconds per frame. Mean pupil diameter and total frames processed also shown.* † *Frames processed in 1/187s.*

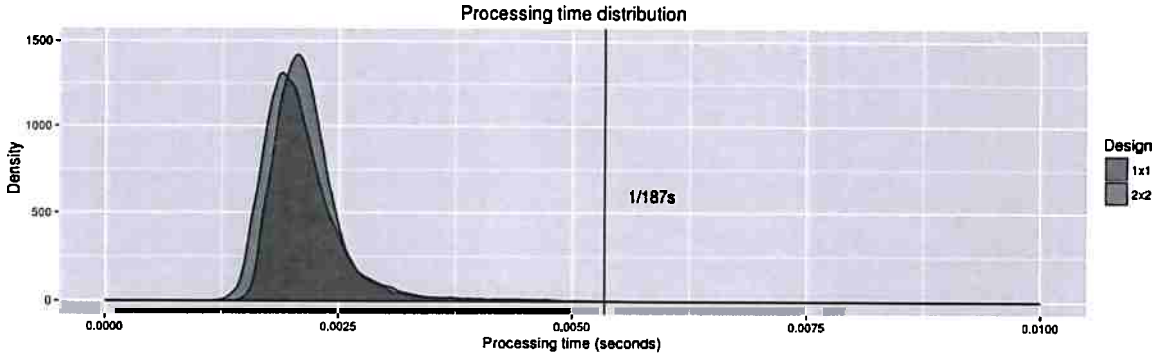


Figure 6.10: *Consolidated processing time density distribution for 1x1 and 2x2 designs.*

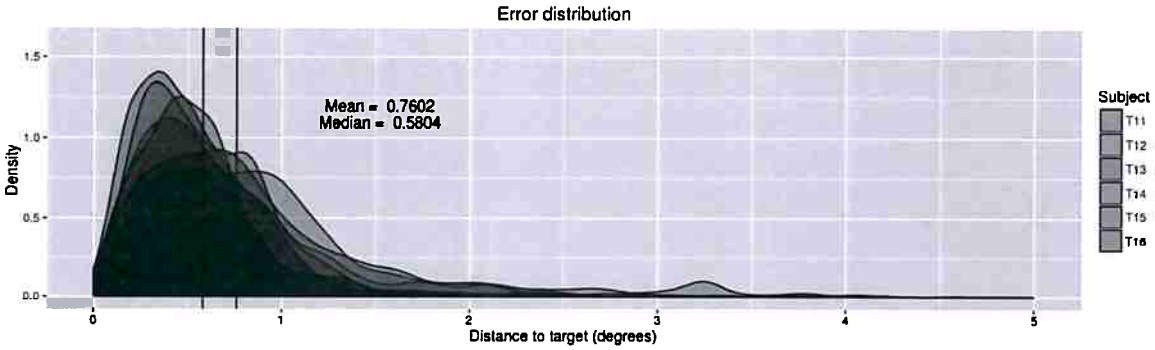


Figure 6.11: *Error density distribution in degrees for all targets for each subject.*

processed, on an universe of $93K$ frames which correspond to nearly 500 seconds. Visual inspection reveled the frames not processed on time are one of two kinds: the pupil is not visible (due to a blink), or; a normal frame, which simply took longer to process. More trial sections are necessary to test the hypothesis of an association between the pupil size (and consequently the size of the ROI used to refine the pupil) and the number of frames not processed on time.

6.2.3 Discussion

The results when taking only the bright and dark pupil frames, using different calibration polynomial coefficients to perform error estimation, do not match the results of Section 6.1.2. This might be caused by a slightly different depth of the bright and dark pupil illuminators (with respect to the camera optical axis). Albeit small, as a result a misalignment was visible between the corneal reflections on image frames. Another hypothesis is the violation of the symmetric bivariate Gaussian distribution profile of the bright pupil CR. Due to a particularity of the prototype, it is now a blob result of two very close bright spots, and being unaware make the automatic process to sometimes bounce between the top and bottom portions of the reflex.

Improvements in computational performance can be achieved by manipulating the value $\aleph$, defined in Section 2.4. The eye ROI is defined as a multiplicative factor of the pupil size. The value employed ($\aleph = 3.0$) resulted in almost the whole frame being the ROI for some subjects. However, despite the pupil size must be considered, the factor can have an additive nature, as the pupil overlap is dictated by the maximum translation the pupil can perform between adjacent frames. Thus, it is a function of the lens magnification (covered field of view) and sample rate, plus the

pupil size.

We were not able to compare the results with other algorithms, as a number of frames are corrupted by the stripe (the most important for comparison purposes indeed). The dual-camera prototype developed to assess the DSL technique were unable to provide the short pulses we require. This is attributed mainly due to high optical losses imposed by the beam splitter and filters employed (see Section E.2).

Chapter 7

Conclusions and future directions

Eye trackers (ETs) are devices that can estimate the point of gaze. High temporal and spatial resolution is desirable to assist on the development of responsive gaze-based *human-computer interfaces* (HCIs). Low cost solutions are increasingly available, but are generally limited at sixty samples per second due to high processing requirements. To reduce the bottleneck, we proposed a technique based on the *differential lighting* (DL) technique, which detects pupil candidates from two adjacent images by exploiting the bright pupil effect.

Thus, the gains introduced by the DL technique can be exploited by any camera, independent of the technology employed or price range. As a tradeoff, additional hardware is necessary to control the light triggering. Another overhead introduced is the processing needed to adjust an external clock tied to the lights, by means of image processing algorithms.

The synchrony signal, which used to come from the camera on the original versions of the DL, was replaced by an external clock adjusted by software. To reduce the complexity introduced by this arrangement, we also investigated the feasibility of allowing the clock to run independently. As a result, the software is simplified, as it stops working to sustain the synchronization, to just keep track of the asynchrony as it evolves. The frames asynchronously illuminated are prone to have artifacts such as dark or bright stripes, and are the result of two snapshots taken by different lights at different points in time. As a result, the pupil might appear dual-colored, which poses further difficulties in the pupil and *corneal reflection* (CR) detection.

Three prototypes were constructed and employed in experimental sections. One is composed by two independent acquisition systems, being able to capture images simultaneously using pulsed and continuous illumination, thanks to a particular arrangement of a lens, a beam splitter, and two cameras. This system allowed the assessment of the software-synchronous approach against dark pupil techniques. Due to losses on the light path, this prototype were unable to produce good images with shorter pulses of light, as required by the asynchronous technique. This limitation motivated the construction of another prototype, capable of delivering a high radiant intensity during very short intervals.

A singular attention was given to the circuits needed to implement the techniques. As is very common in academia, there is a natural split of competences, in which the three ET groups - of users, software developers and hardware developers - have only small intersections. Expecting to provide a good starting point for others interested in implementing the techniques presented, we provided sample circuits which we have used in experiments.

The remaining of the chapter is organized as follows. Section 7.1 discusses particular points

of the technique, including power consumption concerns. Section 7.2 revisits the most important limitations, while Section 7.3 presents further possible research opportunities. Section 7.4 closes this chapter.

7.1 Discussion

In Section 4.2 we assessed a sub-pixel pupil contour estimation using highly off-axis images from the work from Świrski *et al.* (2012). The contour refinement accuracy and processing time using variable number of rays (see Figures 4.7 and 4.8) suggest we can trade-off accuracy by computational performance. For the particular data considered, the performance has little accuracy changes from 20 rays on, while each traced ray is responsible for a mean processing time of $0.028ms$ (and is linearly correlated $R^2 = 0.989$).

We have not done statistical tests on the hypotheses of similar processing times between the techniques presented. As we employed different prototypes equipped with different optics, the covered field of view and image quality diverge. However, the coverage is sufficiently similar ($28 \times 21mm$ against $26 \times 19mm$) for a visual inspection (compare Table 6.4 with Table 6.8). Another factor that influence processing times is the relative pupil size. This can be easily observed on Table 6.6, which presents results from the same subject. Albeit the many uncertainty sources which avoid a comparison of the techniques, the processing times are expected to be very similar. They basically differ on what to do with the stripe estimated. The software synchronization analyze frame images to keep the external clock adjusted, while the asynchronous estimate the stripe position from image whenever possible to feed the Kalman filter.

For our particular implementation of the asynchronous technique, a higher external clock frequency is desirable. With a shorter activation period than the camera, the lights become overlapped for some scanlines, which generates a bright stripe effect. Such a stripe is easily removed by adjusting the exposure to match the stroboscopic light period. The opposite is not true, as a dark stripe can not be compensated by adjusting the camera exposure.

In Section 3.8.2 we discussed a key limitation of our technique. Due to the stripe translation over the frame, the pupil as captured by the camera might be the joint result of two snapshots taken Δe apart. Depending on the pupil relative size and the clocks difference, the pupil estimation latency can change considerably in a couple of frames interval. While this will probably not be an issue for *human-computer interaction* (HCI), further uses might be limited.

As a solution, we offered the possibility to join consecutive frame excerpts illuminated by the same strobe light. In such case, a lower clock frequency is desirable, as it allows the stripe to be completely removed. The drawback is an effective lower sample rate, equivalent to the strobe frequency. Another drawback is an increased latency, ranging from Δt to $2\Delta t$. Despite the problems, using a lower frequency and joining frames allow a less strict stripe size, which allows the pulse length to be increased. This might result in easy to build circuits and allow low power *light-emitting diodes* (LEDs) to be employed.

The improvement in image quality result of stroboscopic light might encourage remote initiatives. The availability of both processing and power resources make it more appealing. While it is possible to use a large number of high power LEDs to yield the same irradiance at the eye, at dis-

tances of $50cm$ to $80cm$ typical of remote eye trackers, the exposure at the nominal $20cm$[1] distance will be much higher. This might result in a device classified in one of the non-exempt risk groups defined by in IEC/EN 62471 (IEC, 2006).

The techniques described are not fully integrated. The capture software implements only a fraction of the functionalities, while the remaining are made available as separate software packages. This is in part due to the need for offline data for post-processing. Running more than one algorithm at $187Hz$ would result in many frames dropped and probably in a synchrony problem for the software-sync technique. The separate pieces were tested standalone, and must be integrated into a fully featured eye tracking solution.

Moreover, to be able to work as a mobile device, a means to compensate head movements is necessary. A common approach is to employ a scene camera, as it allows the study of the activity being undertaken as well. This certainly complicates the processing pipeline, as it requires new algorithms to be run in parallel and new calibration procedures.

7.1.1 A note on power requirements

It is important to highlight that the power consumption is little increased by the use of stroboscopic light. As shown in the *pulse length minimization design* section (C.1), the power needed in pulsed mode can be found using a continuous illumination use case, and is a good indicative that the power consumption is only marginally affected. Possible losses occur due to lower LED efficiency at high forward currents (the *relative total radiant flux* curve at pulsed mode have a slope smaller than one), capacitor internal resistance (known as *Equivalent Series Resistance* (ESR)), the *metal-oxide-semiconductor field-effect transistor* (MOSFET) switching and internal on resistance (the MOSFET we employed is expect to have a low on resistance of less than $50m\Omega$ (ON Semiconductor, 2008)). As an example, in our operating conditions, at $187Hz$, the on resistance alone is responsible for a power loss of $37.4mW$, tenfold lower than the losses introduced by the current limiting resistors, which is on the order of $374mW$.

Particularly for the prototype described in Section E.3, a considerable reduction in consumption can be achieved by employing lenses in front of the LEDs. The LED we employed has a half angle of $\pm 45°$, but our experimental conditions (distance of $130mm$ and circular area lit of $35mm$) require less than $\pm 8°$. Considering the emission drops linearly (crossing a relative radiation of 0.5 at 45°)[2], the reduction in power can reach a factor of four if lenses are employed.

A further reduction in consumption can be achieved by keeping the link between the running system and the illuminators, allowing the lights to be turned off during blinks. Particularly, they can be poled from time to time to check if the eyes are already open, resuming normal operation whenever possible.

7.2 Limitations

Two key limitations were already discussed in Section 6.1.4 and are worth being revisited. One is associated to the lack of eyelid modeling, and the other is related to the outcomes under IR-rich

[1] For classifying the risk group, a nominal distance of $20cm$ is considered (IEC, 2006).

[2] This is a very conservative assumption, considering the *radiation characteristics* graph found in OSRAM Opto Semiconductors (2014a).

environments.

The pupil refinement is based on a least-squares method to fit an ellipse to a set of points. As a consequence, the response to outliers is very bad. A statistical filtering is employed based on the distance to the overlap between pupils from adjacent frames to improve the fitting. While the method performs well for filtering isolated points or small sets of points due to the eyelashes, bigger sets such as the ones resulting from eyelid occlusion, make the method ineffective.

Both techniques described rely on camera exposures close to the frame period, which under some circumstances might render an unusable frame. External light sources like incandescent bulbs, indirect and direct sunlight are examples of illumination sources which might pose a problem. In such cases, the ambient light may be strong enough to render a less evident bright pupil, a brighter dark stripe and the reintroduction of undesirable effects of blur and skew.

Section 3.8.2 also presented an important limitation. It is associated to the relative pupil size and the corresponding bright response. For particular pupil sizes, the average pupil intensity might be very similar to the iris, making it difficult to detect the border. The opposite can also occur, with the average pupil intensity very close to the CRs, yielding a difficult and sometimes erroneous CR detection. To attenuate the problem, Ebisawa (1995) proposed to control the LED current based on the pupil size. A similar approach can be adopted using the techniques presented, without expected side-effects.

In addition to the already discussed limitations, the technique is prone to errors when other reflexes are present on the image. Such reflexes might be caused by glasses, for example, as shown in Figure A.3bc. While bigger reflexes might not be confounded with CRs due to size restrictions, they interfere on the difference image used to estimate the pupil position on the frame. As a result, the tracking can be impaired. Wearing glasses however, does not imply in reflexes from the lenses on the image, as shown in Figure A.3a.

It is interesting to add a note about the stripe tracking using the Kalman filter. Assume for a moment that there is no drift between the clocks. Assume also that the filter internal state is the true representative of the system state (stripe position and velocity), at some time t. Thus, it is correct to say that the predicted state at time $t + \varepsilon$, for any ε, will correspond to the true system state[3]. However, the prediction is invoked in the event of a new frame, which might get delayed due to a number of reasons. Thus, one prediction will consider a large interval, while the next will possibly consider a smaller. However, at the hardware level, the camera period is constant. As a result, the stripe estimated might contain errors even in the scenario of an optimal filter. While this might not sound good, a positive consequence is that a frame drop do not impact the next predictions.

7.3 Future work

Further research might explore some kind of eyelid modeling. Tracking the eyelid contour would result in a more robust pupil detection. The additional computational cost can be readily compensated by an earlier blink detection, avoiding the search for an occluded pupil. A possible direction is to model the eyelids using active contours (Kass *et al.*, 1988), as done by Ryan *et al.* (2008b).

[3]Assume for simplicity, that ε is such as to not make the stripe go to the invisible scanlines or become negative, in which cases the estimated $\hat{S}$ interfere with the predicted value.

A future study might also use a lower frequency external clock and join consecutive frames, as discussed in Section 3.8. For a given difference in frequency, this allows to compensate the stripe of a lengthy pulse. A primary advantage is a lower power surge requirement, as the LEDs are allowed to stay on for a longer period. Another advantage is that artifact-free frames can be obtained. The trade-off is a lower effective sample rate and increased latency, as the sample is effectively given by the external clock pulses in this case.

Given the work from Ebisawa (1995), a natural extension of this work is to allow the automatic strobe power adjustment in order to avoid bright pupil saturation, in which case the CR becomes hard to detect. The approach can also help on increasing the contrast of constricted pupils, as the power can be momentarily increased.

Further work needs to be done to establish whether the automatic procedures introduced to adjust the thresholds used on the pupil overlap detection and CR can result in increased robustness to changing lighting conditions, defining a protocol for testing this particular feature.

Both techniques rely on close to maximum exposures, which are prone to problems in an IR-rich ambient. Further investigations in the direction of reducing the exposure are strongly recommended, as this will increase possible use-cases for the technique.

With the results of the external clock adjustment on the order of 1.0 to 1.7 scanlines/s, and considering a scanline is equivalent to $19.23\mu s$ in the studied setup, it would be interesting to investigate if the external clock could be used to simulate a synchronization output from the camera, as it might result in affordable solutions to multiple camera synchronization problems.

Finally, the use of stroboscopic light to illuminate a frame results in high quality images. Thus, the potential gains are more evident when high speed movements are imaged, such as when a saccade occurs. Additionally, using a high sample rate allows the detection of small eye movements which otherwise can occur between two camera samples. Thus, further research should be carried out to establish a protocol for testing saccades.

7.4 Closing remarks

The proposed software-synchronous solution has the potential to transform an ordinary camera into a high-grade one, improving the image quality by allowing a very short virtual exposure and also keeping the lights synchronized with the camera frames. Our solution can be thought and implemented as a middle layer, running on top of a low cost hardware and delivering high quality images for an eye tracker unaware of its existence. The increase in consumption due to the pulsed lights is expected to be minimal, assuming the illumination is already necessary.

In addition to the synchronization method, we also proposed a lightweight pupil detection and tracking which exploits the DL technique. As far as possible, ground truth pupil contour data were used to show the potential of the pupil refinement in terms of accuracy. Performance comparisons with state-of-art pupil detectors were done using a typical eye tracking experiment, using a custom dual-camera prototype capable of imaging the eye using both continuous and stroboscopic light. The results show the potential of our technique to obtain better results at a lower computational load.

A proper asynchronous solution was also proposed. It simplifies the additional hardware necessary to pulse the lights, as it is allowed to run independently of the eye tracking system. Such

solution presents specific demands to deal with dual-colored pupils and corrupted CRs result of two light pulses. Experimental results with an eye tracking task reveled its feasibility to work as good as the synchronous counterpart.

Both techniques are lightweight and can be easily tuned to specific demands. The results suggest they can outperform existing algorithms both in speed and accuracy, with about 0.7° accuracy and a feasible 500 Hz sample rate on a modest platform.

Appendix A

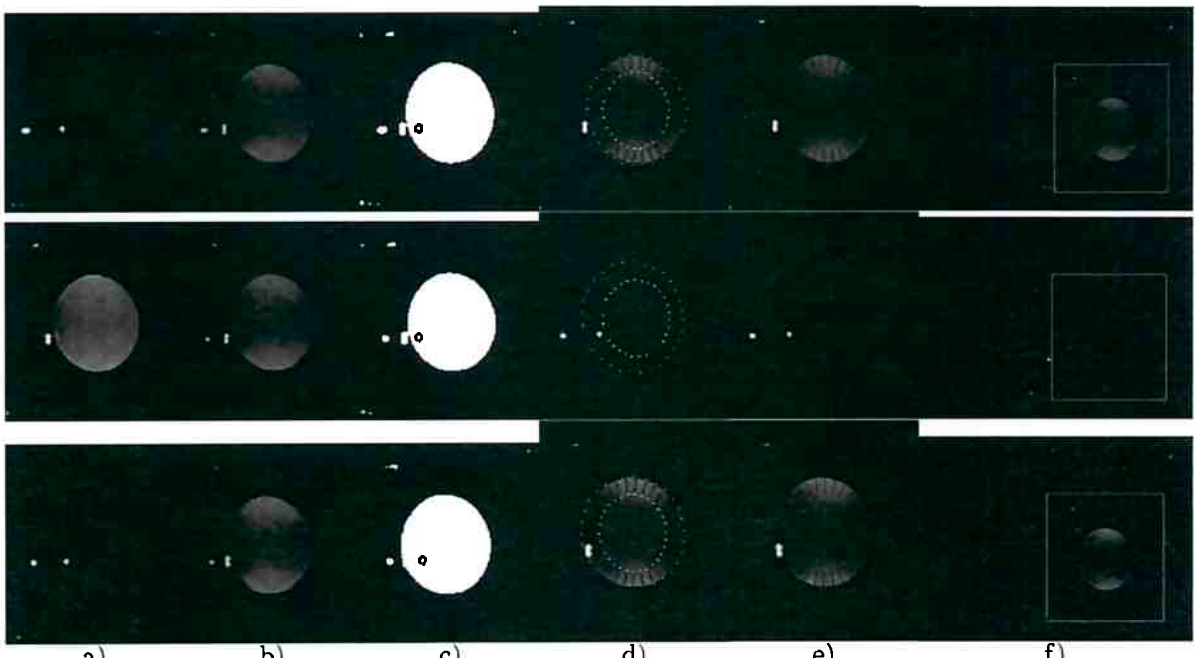
Additional results

This chapter presents some additional results which were suppressed on the corresponding sections to avoid breaking the text flow.

Figure A.1 shows a sequence of 21 frames captured of a subject wearing makeup. A saccade of about $110ms$ takes place while the stripe (not visible, due to a combination of parameters) crosses the pupil. Note in frames 566 to 576 that the estimated pupil appears skewed because the ellipse is fitted to points belonging to pupil positions taken at different moments.

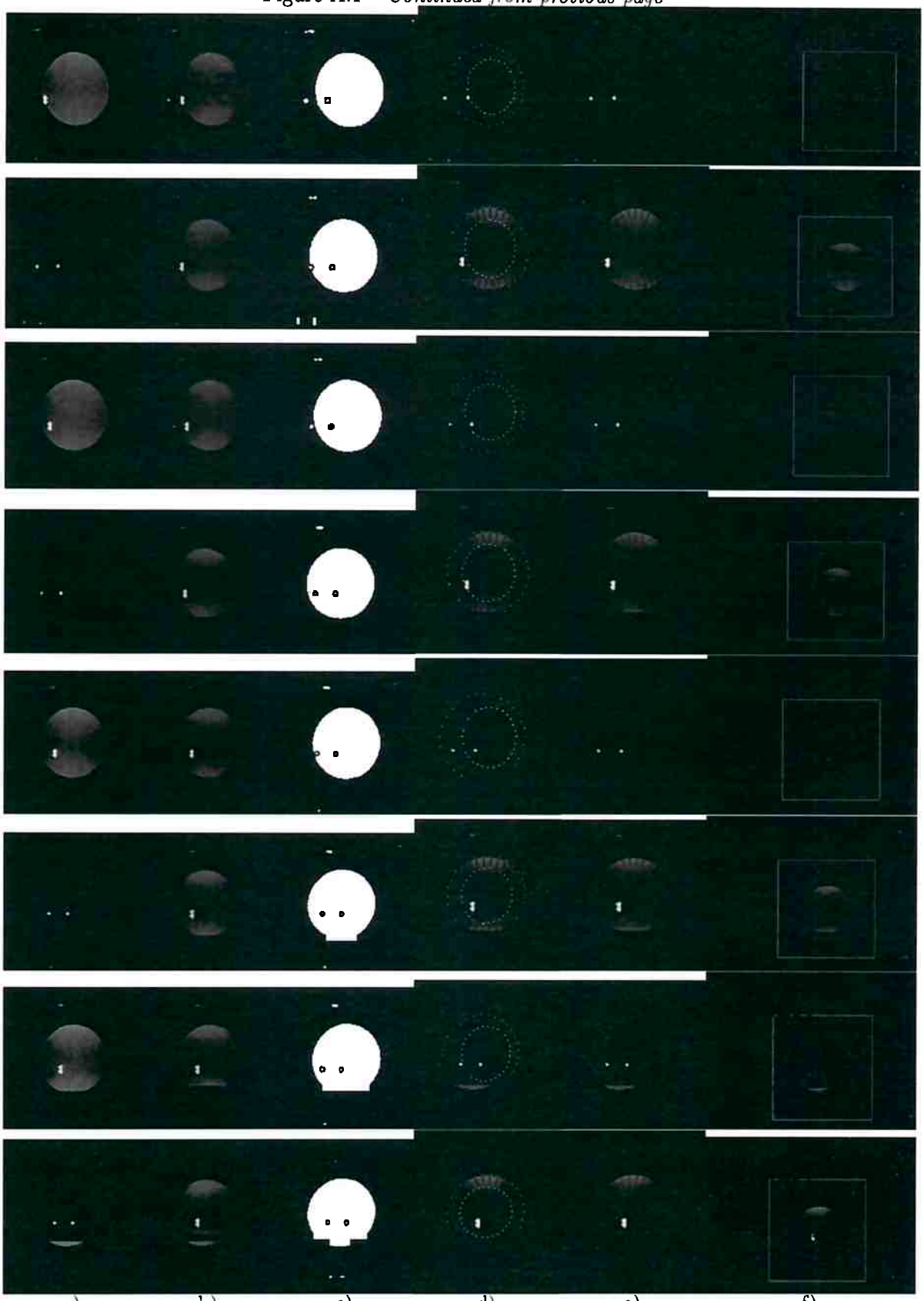
Except for the last column, all images are excerpts corresponding to the calculated ROI based on the pupil radius. The columns contents are as follows. a) Correspond to the previous frame; b) Image difference between the consecutive frames; c) Thresholding result with stripe gap filled with neighbor pixels; d) Rays traced until intersecting the stripe; e) Rays actually employed on the pupil border refinement; f) Tracking result.

Figure A.1: *Pupil tracked using the ADSL during a saccade.*



Continues on next page

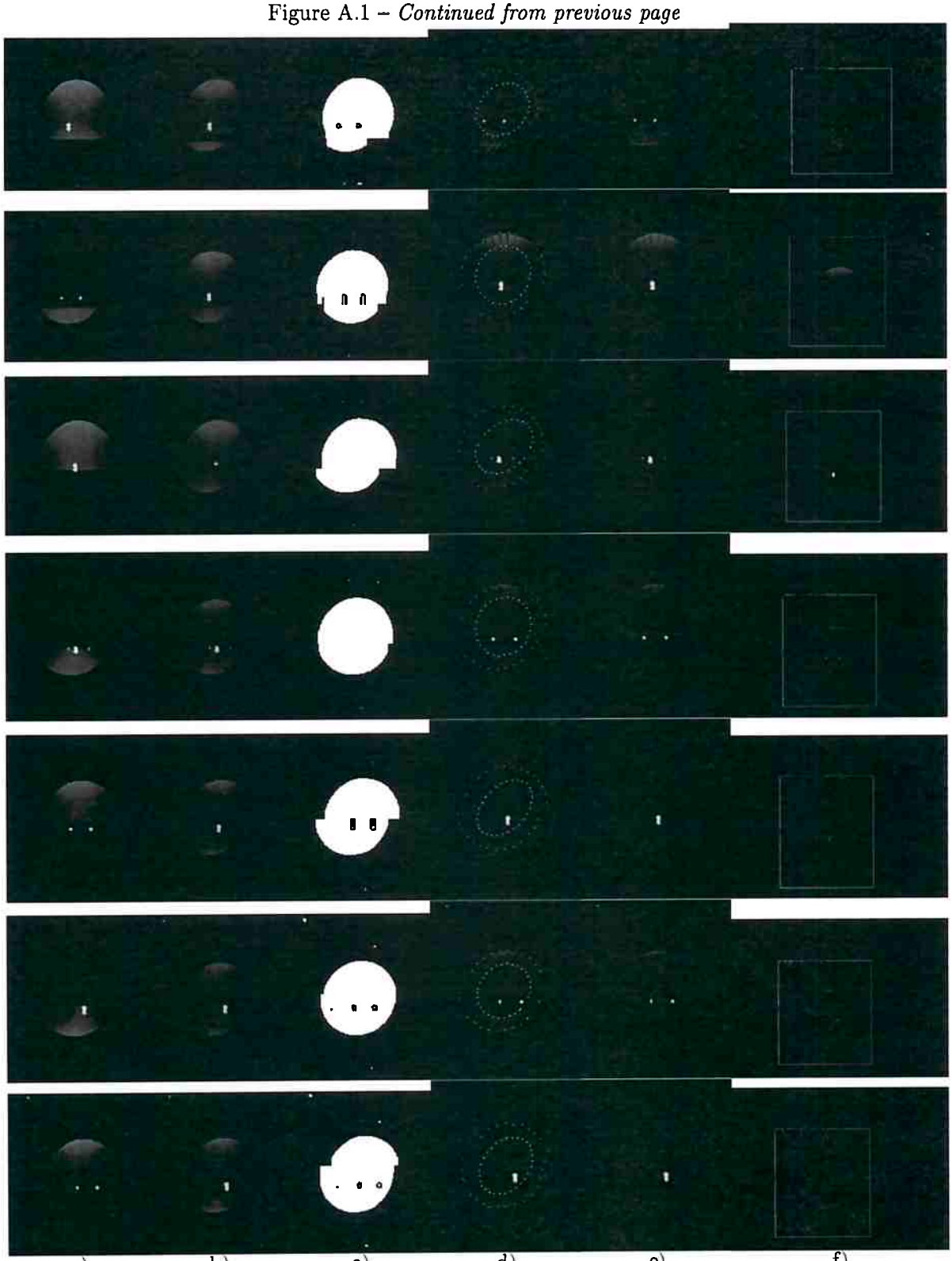
Figure A.1 – *Continued from previous page*



a) b) c) d) e) f)

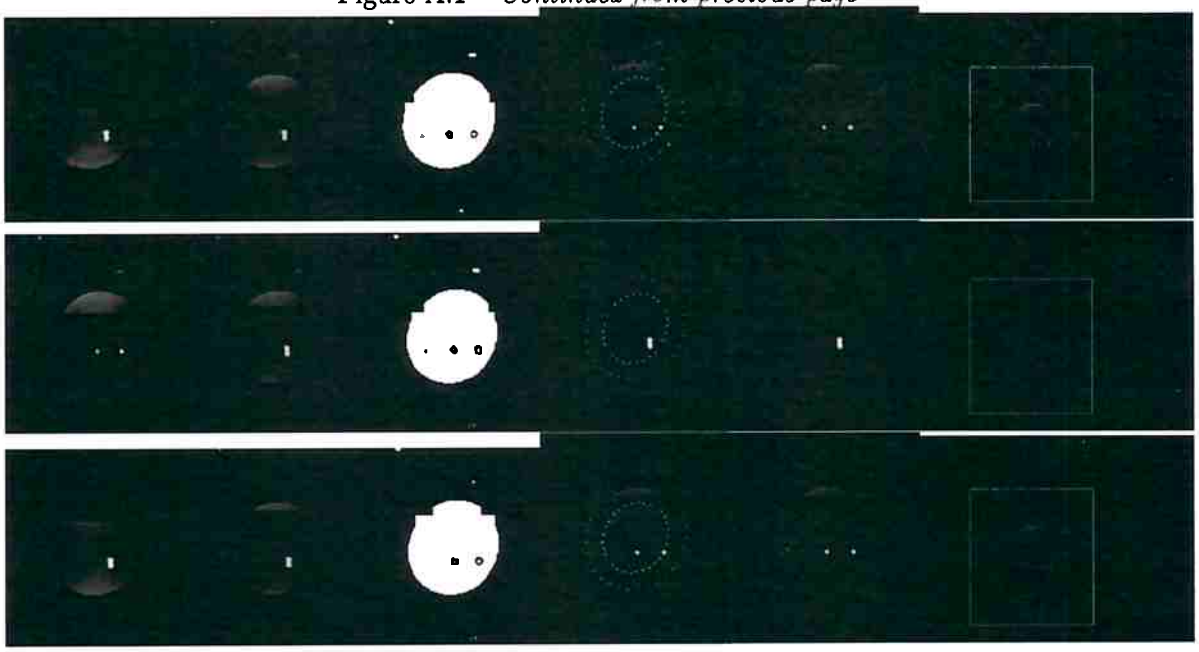
Continues on next page

Figure A.1 – *Continued from previous page*



Continues on next page

Figure A.1 – *Continued from previous page*

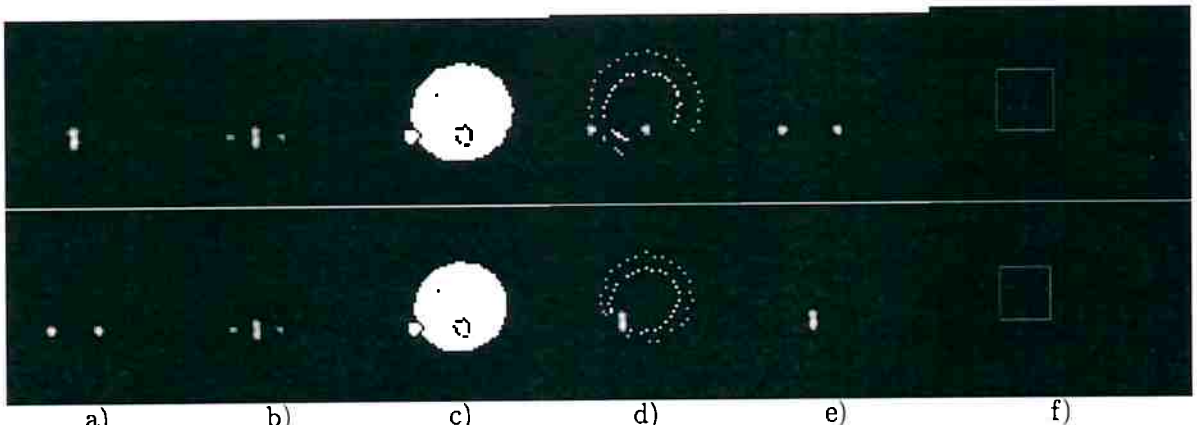


The bright pupil effect is idiosyncratic and varies with age (Nguyen *et al.*, 2002). Additionally, it also varies with the gaze angle (Agustin *et al.*, 2006). A small pupil might render a bright pupil with a mean intensity very close to the iris, making the contour detection more challenging.

Figure A.2 shows a sequence of 13 frames captured of a volunteer subject to $2000lux$ of visible light, provided by white LEDs. The pupil size was estimated to be about $2.8mm$ in diameter. The stripe (not visible), crosses the pupil during the interval.

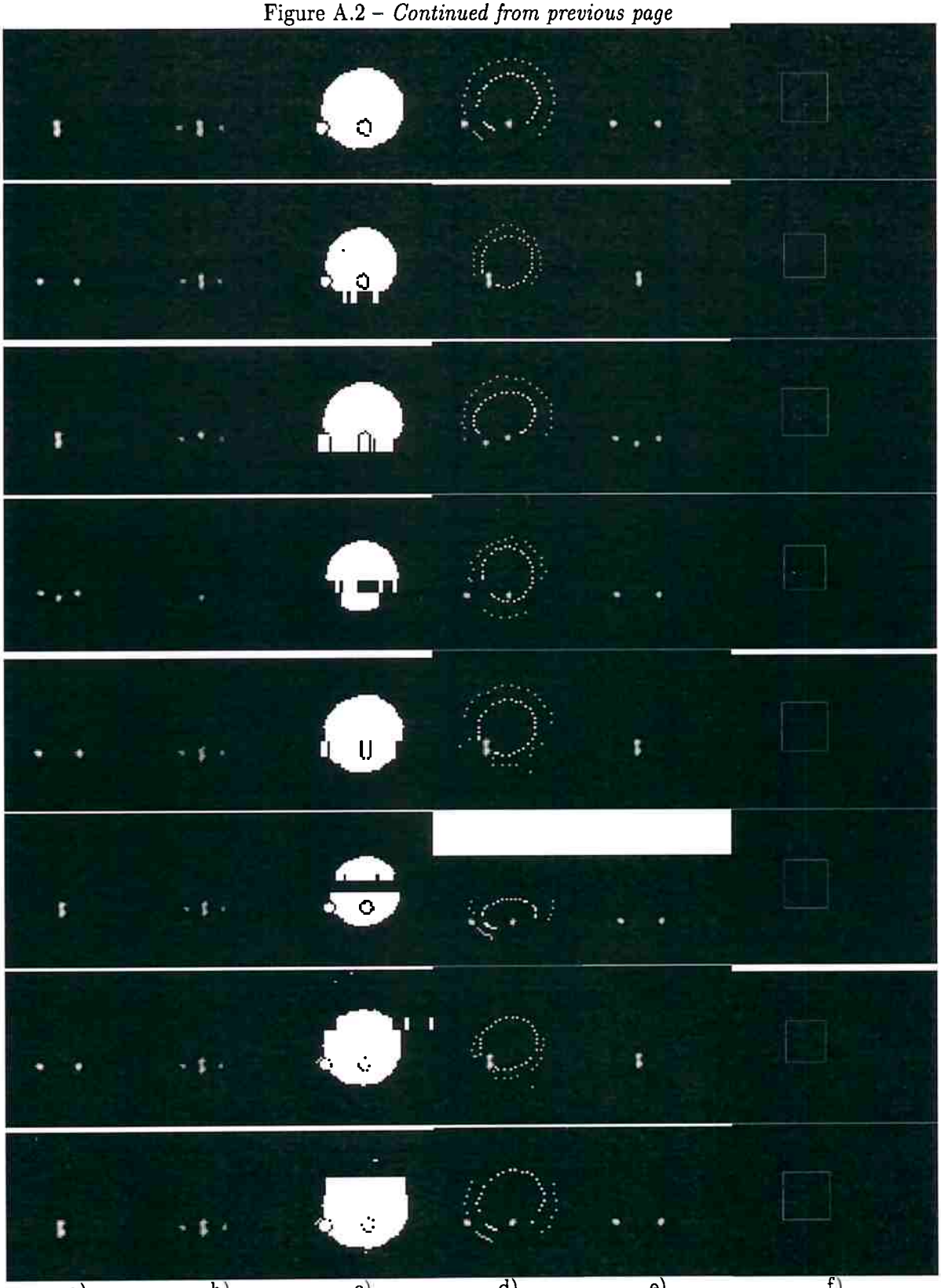
Note on frame 56, that the process of filling the gap of the stripe failed. This probably happened because the stripe position was mistakenly estimated ahead of the true position. The condition which can cause such event is discussed in Section 7.2. Note however, the ellipse fitting performed pretty well using points from roughly half the pupil.

Figure A.2: *Pupil tracked using the ADSL with a measured illumination of 2000lux at face.*



Continues on next page

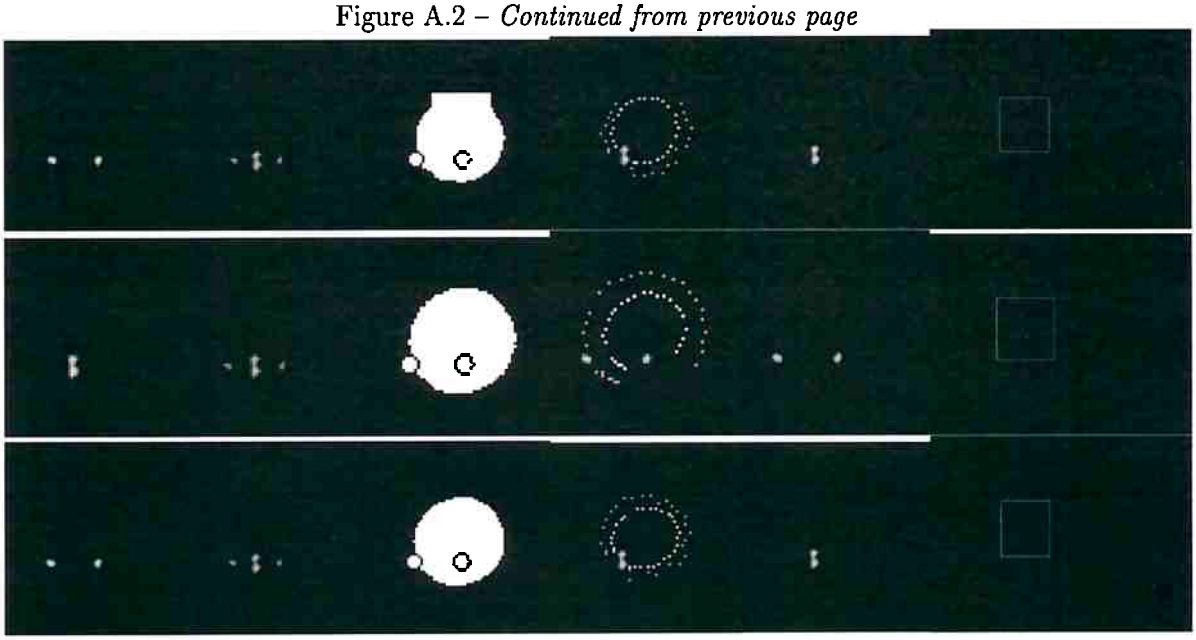
Figure A.2 – *Continued from previous page*



a) b) c) d) e) f)

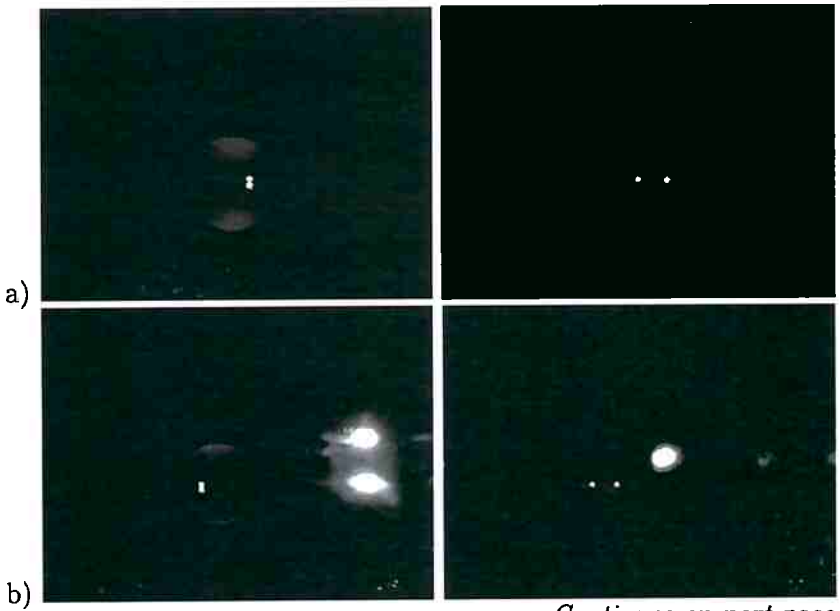
Continues on next page

Figure A.2 – *Continued from previous page*



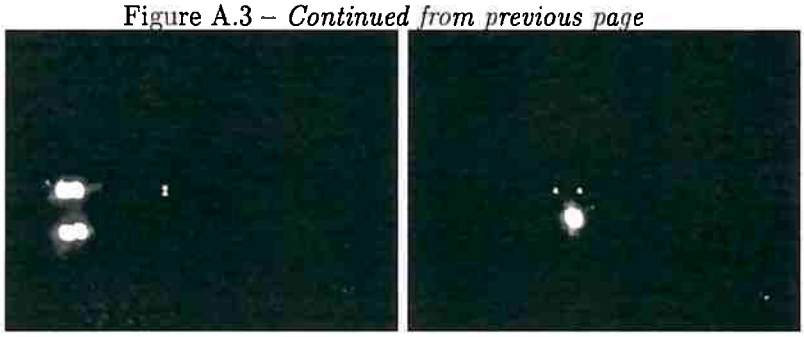
Glasses become a problem when they reflect the illumination back to the camera. On the experimental sections we performed, only one subject wore glasses with no problems, as can be seen in Figure A.3a. In another test with a different subject, the reflex on the lens persisted even after changing the eyeglasses model (Figure A.3bc).

Figure A.3: *Sample images from volunteers wearing glasses.*



Continues on next page

Figure A.3 – *Continued from previous page*

c) 

Appendix B

Eye safety standardization

There is an increasing concern about the safety of light emitted by *light-emitting diodes* (LEDs). The motivation is the raising power output, wavelength coverage and the low-cost of these devices (Altkorn *et al.*, 2005; Behar-Cohen *et al.*, 2011; Castro, 2008; Topal *et al.*, 2014). The risks associated to the light depend upon many factors which will be described along this chapter.

There are several standards and guidelines developed by international organizations associated to the light hazard assessment[1]. Two of greatest relevance will have their limits presented, the *International Electrotechnical Commission* (IEC) and the *International Commission on Non-Ionizing Radiation Protection* (ICNIRP).

The ICNIRP publish a *Guideline on Limits of Exposure to Broadband Incoherent Optical Radiation*, which assist on the development of principles of protection to the eyes and skin against optical radiation hazards, including standards like the IEC/EN 62471, which will be presented next.

Before 2008, most LEDs and products containing visible LEDs, were tested and labeled in accordance with the IEC/EN 60825-1 laser safety (coherent light source) standard (IEC, 2007). However, as LEDs are not sources of coherent light, the standard was not adequate as limits for lasers are more restrictive. As a result, in late 2008 a newer standard, referred to as IEC/EN 62471 was adopted for conventional, or lighting class, LEDs (Cree Inc., 2013; IEC, 2006). The result of such late introduction is that many documents still refer to the former standard and there are recent works which still follow it (such as Topal *et al.* (2014)).

The new standard is named *Photobiological Safety of Lamps and Lamp Systems*, which reflects its scope for both the light and products using it. The IEC/EN 62471 standard specifies four classifications, called risk groups, for lamps and lamp systems (excluding lasers) emitting light in wavelengths from 200 to 3000 nm as set forth in Table B.1.

The IEC/EN 62471 defines different exposures based on time and area of the human body affected (IEC, 2006). Of special interest are the limits associated with infrared radiation hazard for the eye (cornea), and the thermal hazard to the retina. The danger of *near-infrared* (NIR) radiation is caused mainly by tissue overheat, and accordingly, exposition time plays a fundamental role. For details on potential optical radiation hazards and a comprehensive introduction on radiometric measures, please refer to Mulvey *et al.* (2008).

Other aspects necessary to determine the eye safety limits include the apparent source size of the LEDs viewed through the optics and the wavelength. The first is directly related to the exposure

[1]For a list of organizations and related standards and guidelines, see Altkorn *et al.* (2005)

Risk Group	Risk	Definition
Exempt	None	No photobiological hazard
RG-1	Low Risk	No photobiological hazard under normal behavioral limitation
RG-2	Moderate risk	Does not pose a hazard due to aversion response to bright light or thermal discomfort
RG-3	High risk	Hazardous even for momentary exposure

Table B.1: *Risk groups as defined in IEC/EN 62471 (IEC, 2006).*

of the retina, as the smaller the apparent size (the spot size), the most effective is the radial cooling by the neighboring areas on the retina. The larger the apparent size of the source, the larger the image formed on the retina and more likely to make injuries to the tissue.

The second, the wavelength, also plays a fundamental role in defining the risks. There are different adverse effects related to distinct wavelengths. More details in section B.1.

Depending on the task, the subject is supposed to look to the source of radiation in a day long basis (i.e. a traditional CRT monitor), while some may only occur on accidental exposures (like a laser accidentally directed towards the eye). For an eye tracker, it is very likely that the exposures will extend for hours and thus, it is imperative that the final arrangement be in the exempt group, where no restrictions to use apply. The emission limits for this group in the infrared band are defined as (IEC, 2006):

$$\begin{aligned} L_R &\leq 28000/\alpha \quad \text{within 10s - for retina} \\ L_{IR}\ \text{(low visual stimulus)} &\leq 6000/\alpha \quad \text{within 1000s - for retina} \\ E_{IR} &\leq 100 \quad \text{within 1000s - for cornea} \end{aligned} \tag{B.1}$$

where $d = 0.2m$ is the fixed distance to define the risk group and α is given in *radians*. Radiance is given in $W/m^2.sr$ and irradiance is given in W/m^2.

The remainder of this chapter is organized as follows: First we present the biological effects of NIR radiation to the eye structures in section B.1. Next, the exposure limits for the anterior segments are presented in section B.2, followed by exposure limits for the posterior segments (retina) in section B.3.

B.1 Biological effects to the eye

As a general rule, light poses a health hazard only to absorbing tissues, which in turn varies with the wavelength and accordingly, the biological effects on different eye structures. Radiation in the infrared band can cause thermal retinal and corneal hazard and possible delayed effects on the lens (ICNIRP, 2013; IEC, 2006). The radiation absorbed by the anterior segment (the cornea, aqueous humor, and lens) can produce clouding of the cornea and lens when the corresponding thresholds are exceeded. Exposure limits are set to protect both against acute as well as chronic exposure (Aly and Mohamed, 2011).

On the other hand, blue light can photochemically destroy the photopigments (and some other molecules) which then act as free radicals and cause irreversible, oxidative damages to retinal cells. As NIR is used, these adverse effects do not pose a risk for eye trackers in general. Therefore, this reaction will be ignored in the rest of the text.

B.2 Exposure limits for the cornea

The ICNIRP recommend a maximal daily corneal exposure of $100W/m^2$ total irradiance for wavelengths 780-3000nm (E_{IR}) for day-long, continuous exposures (Mulvey *et al.*, 2008). This is exactly the same value adopted by the IEC/EN 62471 (IEC, 2006)

$$E_{IR} = \sum_{\lambda=780}^{3000} E_\lambda \cdot \Delta\lambda \leq 100 \quad W/m^2 \quad for \quad t \geq 1.000s \tag{B.2}$$

where E_λ is the spectral irradiance in $W/m^2/nm$ and $\Delta\lambda$ is given in nm.

As of 2013, ICNIRP reviewed the exposure limit to cornea and lens, as infrared LEDs emit in a small wavelength band (in general, *Full width half maximum* (FWHM)[2] of 45nm). Only a small fraction of the optical radiation that is incident on the cornea in the wavelength range of 780 nm to about 1,000 nm is absorbed in the anterior parts of the eye as the cornea is transparent and the iris is also partially transmitting for infrared radiation. The exposure applicable to the wavelength range between 780 nm and 1000 nm were then weighted with a factor of 0.3, which raised the permitted exposure of the eye for NIR LEDs by a factor of about three (ICNIRP, 2013). This is in line with the studies from Jelinkov *et al.* (2004), which found absorptions of more than 99% in $\lambda = 1.34\mu m$ decreasing to about 50% at $\lambda = 1.08\mu m$, considering all layers before retina (cornea, aqueous humor, lens, vitreous humor). The exposure according to the ICNIRP becomes then

$$E_{IR} = \sum_{780}^{1000} 0.3 \cdot E_\lambda + \sum_{1000}^{3000} E_\lambda \tag{B.3}$$

For a *near-infrared* LED, a good approximation for (B.2), for a device with emission centered about λ is (Jäger, 2010)

$$E_{IR} = E_\lambda \tag{B.4}$$

and accordingly, (B.3) becomes $E_{IR} = 0.3 \cdot E_\lambda$.

For shorter exposures ($t \leq 1000s$), the limit of the ICNIRP and IEC/EN 62471 (ICNIRP, 2013; IEC, 2006) can be approximated by

$$E_\lambda = 1.8 \cdot t^{-0.75} \cdot 10^4 \quad W/m^2 \tag{B.5}$$

where t is given in seconds.

B.2.1 Irradiance from LEDs' datasheet

The irradiance E_e can be calculated from the radiant intensity I_e and distance d (for die size $\ll$ d) using the inverse square law, as follows:

$$E_e = \frac{I_e}{d^2} \tag{B.6}$$

Note that (B.6) is an approximation which assumes the LEDs approximate a point source. The error introduced by such is on the order of 1% when d is five times the die size (largest dimension

[2]Given by the interval between the two wavelengths where the power is equal to half its maximum.

in case of rectangular dies) (Ryer and Light, 1997, Chap.6). If more that one light is activated simultaneously, the emissions must be added, assuming a perfect overlap between them (unless of course, the overlap is physically impossible).

Note also that we assumed the die size, which is the active chip emitting light inside the LED, as the emitter size. However, lens present on the LED encapsulation and others, typically put in front of them, change the apparent size. In such cases, the apparent die size must be considered instead. If it is not five times smaller than the distance to the eye, an extended source must be assumed. We will not cover such case in this chapter, as it is not common in video-based eye tracking.

In both our synchronous and asynchronous techniques the lights are kept on for a fixed period in each frame time. This gives a variable duty cycle based on the frame rate. For those cases where pulsed lights are employed, a weighted radiant exposure is used against the limits (IEC, 2006). The weight is given by the duty cycle, calculated as the ratio between the pulse length and period between two pulses.

B.3 Exposure limits for the retina

Besides the exposition time, the limits for retina are associated with the pupil diameter, the wavelength, and the emitting source size. The wider the pupil, greater the risk as more light is allowed to enter the eye. As the NIR radiation presents low to none visual stimulus, a full open $7mm$ pupil diameter is assumed. As the danger to the retina is basically thermal, the exposition limits are bound by a so-called burn hazard weighting function also, as follows (ICNIRP, 2013; IEC, 2006):

$$R(\lambda) = 10^{\frac{700-\lambda}{500}} \tag{B.7}$$

where λ is the emitter wavelength and in this formulation accepts values ranging from $700nm$ to $1050nm$. The closer to $700nm$, higher the value of $R(.)$.

The emitter is focused by the eye lenses which projects an image into the retina. That small spot receiving light at the retina depends on the size of the emitter and its distance (these two variables define the subtended angle (α), the angle in which the rays enter the eye). Due to the inherent eye movements and the dynamic heat dissipation of the tissue, there is a lower and higher bound for α as shown in Table B.2.

Figure B.1 shows the measure, α, given in radians, used on retinal exposure calculations. The planar angular subtense is obtained by finding the arc tangent $2tan^{-1}(d_s/2r)$, and can be well approximated by dividing the source extent (d_s) by the distance (r), for small angles such as α_{max} of Table B.2.

The near infrared burn hazard radiance L_{IR}^{EL} (for weak visual stimulus) and $t > 10s$ is limited to (IEC, 2006)

$$L_{IR}^{EL} = 0.6 \cdot \alpha^{-1} \cdot 10^4 \quad W/m^2/sr \tag{B.8}$$

where α is the subtended angle. The maximum allowed radiance for short pulses (for $t = 10\mu s$ to $10s$) is defined as follows:

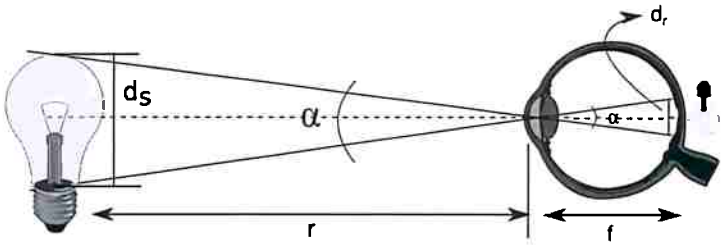


Figure B.1: *The eye and the planar angular subtense, α, of a source and the retinal spot size (adapted from ICNIRP (2013)).*

Exposure duration	α_{min} radians	α_{max} radians	Source
$t \geq 10s$	0.011	0.1	(IEC, 2006)
$0.25s \leq t < 10s$	$0.0017 \cdot (4t)^{0.5}$	0.1	(IEC, 2006)
$t \geq 0.25s$	0.0015	0.1	(ICNIRP, 2013)
$t \leq 0.25s$	0.0017	0.1	(IEC, 2006)
$625\mu s \leq t < 0.25s$	0.0015	$0.2 \cdot t^{0.5}$	(ICNIRP, 2013)
$t < 625\mu s$	0.0015	0.005	(ICNIRP, 2013)

Table B.2: *Limits of the angular subtense α.*

$$L_R^{EL} = 5 \cdot \alpha^{-1} \cdot t^{-0.25} \cdot 10^4 \quad W/m^2/sr \tag{B.9}$$

where α is the subtended angle and t is the exposure length in seconds. The ICNIRP defines a more restrictive limit for a different time range, from $t = 1\mu s$ to $100s$, as follows (ICNIRP, 2013)

$$L_R^{EL} = 2 \cdot \alpha^{-1} \cdot t^{-0.25} \cdot 10^4 \quad W/m^2/sr \tag{B.10}$$

The actual radiance is calculated as follows:

$$L_{IR} = \sum_{\lambda=780}^{1400} L_\lambda \cdot \Delta\lambda \cdot R(\lambda) \tag{B.11}$$

where L_λ is the spectral radiance in $W/m^2/sr/nm$, $\Delta\lambda$ is given in nm and $R(\lambda)$ is the burn hazard function.

B.3.1 Radiance from LEDs' datasheet

For an infrared LED, a good approximation for (B.11), for a device with emission centered about λ is (Jäger, 2010)

$$L_{IR} = L.R(\lambda) \tag{B.12}$$

which can be further approximated to (Bass *et al.*, 1996, Chap.24):

$$L_{IR} = \frac{I_e \cdot R(\lambda)}{A} \tag{B.13}$$

where A is the emitter area.

Note again, lens present on the LED encapsulation and others, typically put in front of them,

change the apparent emitter area. In such cases, the apparent die area must be considered instead. However, if it is not five times smaller than the distance to the eye, an extended source must be assumed. We will not cover such case in this chapter, as it is not common in video-based eye tracking. Note also, that if more than one LED is used and contrary to what happens to the eye lens, the emissions do not need to be summed, as different areas will be affected on the retina.

B.3.2 Radiance of pulsed sources

When a pulsed source is employed, as with both our synchronous and asynchronous techniques, the emission must be assessed using two time scales: considering the radiance of a single pulse; and considering a long term averaged radiance. The radiance of a single pulse must be at most (B.9) and is calculated using (B.13).

The averaged radiance can be calculated as follows

$$E_{avg} = E_{pulse} \cdot D \tag{B.14}$$

where D is the duty cycle, calculated as the ratio between the pulse length and period between two pulses.

E_{pulse} is calculated using (B.13) and E_{avg} must be compared against (B.8). Note that the effective subtended angle can be different for each time scale (see Table B.2).

Appendix C

Asynchronous design goals

As shown in previous sections, the technique benefits from the shortest pulses as possible. In general, the total radiant flux in function of the forward current can be well linear approximated with a unitary slope. Thus, as a rule of thumb, reducing the on period by half can be compensated by doubling the current. High power *light-emitting diode*s (LEDs) have specific current curves depending on duty cycle. Different maximum currents can be handled depending on the active period. Being particularly close to the permissible pulse handling capability of the device, might result in no room for current compensations.

The OSRAM SFH4715 is a high efficient *near-infrared* (NIR) LED centered at 850nm, qualified for automotive use, can operate in a wide temperature range of $-40°$ up to $125°C$ (OSRAM Opto Semiconductors, 2014a). We will use this device along the section as it presents a high total radiant flux ($630mW@I_f = 1A$) in a small package outline ($3.95 \times 3.95mm$), making it a good choice for eye tracking applications.

As an example of the limitation on the current increase to compensate for pulse length reductions, consider the OSRAM device. The maximum forward current for a pulse length $t_p = 1ms$ and duty cycle $D = 20\%$ is $I_f = 2.37A$. By reducing the pulse length by half, but keeping the same pulse frequency, the current can be increased to only $I_f = 3.55A$ (with $t_p = 500\mu s$ and $D = 10\%$)[1]. The current could not be doubled to compensate the reduction in pulse length.

Another important issue on reducing the pulse length is the safety to the eyes. Different limits apply for continuous ($t > 10s$) and pulsed sources for both corneal and retinal exposures. As the power must be increased when the pulse size is constrained, a specific limit might be reached even with the time-averaged exposure inside the safe levels.

C.1 Pulse length minimization design

The technique benefits from the shortest pulses as possible. The pulses cannot be reduced arbitrarily, as the power must be increased, rising the concerns about the safety to the eyes and the LEDs capabilities. We have defined a procedure to help on designing eye trackers based on the asynchronous technique, which is safety and device centered, that allows one to find the shortest manageable pulses for a given setup. Therefore, the aim is not to provide a device selection guide, as

[1] Approximations taken from the permissible pulse handling capability graph, available from OSRAM Opto Semiconductors (2014a)

important aspects of the eye tracker are assumed to be defined, such as the distance to the subject and the illumination source employed.

The procedure can be divided into sequential steps as follows:

1. **find *maximum permissible exposures* (MPEs)** according to the latest standards which are summarized in Appendix B. Note that the distance to the illuminators and their die size (as viewed though the lens) are mandatory;

2. **determine LED forward current** of the illuminators necessary to reach the MPE. This step is a safety measure for the pilot test, in case it is done with human subjects;

3. **pilot test** with the illumination turned on continuously, while adjusting the power to obtain satisfactory image quality (in terms of brightness and contrast). The exposure must be set to maximum and the gain to a small value to reduce noise. The outcome of this step is the power needed to produce good frames at a specific camera setup;

4. **find radiant intensity at pilot test** from the power drawn, using the illuminator documentation;

5. **find the minimum duty cycle** to reach the same average power, while still being inside the safe limits.

Step 1 - Find MPEs

Considering a long term exposure, the MPE for the cornea is fixed in $E_{IR} = 100W/m^2$ (B.2). Using (B.6) to obtain the radiant intensity I_e from the MPE irradiance we have

$$I_{IR} = d^2 \cdot n^{-1} \cdot 10^{-1} \quad mW/sr \tag{C.1}$$

where the distance (d) is given in millimeters and n denotes the number of LEDs turned on simultaneously.

The limit exposure for the retina is given in (B.8), and depends on the angular subtense, which in turn depends on the distance and area of the illuminator. Assuming an illuminator with a squared die with side w, the maximum radiant intensity from the source is given by

$$I_{IR}^{EL} = 6dw \cdot R(\lambda)^{-1} \quad mW/sr \tag{C.2}$$

with the distance (d) and the die side (w) given in millimeters and $R(\lambda)$ defined as in (B.7). The formulation is for subtended angles within the range $[\alpha_{min}, \alpha_{max}]$. If α is smaller than defined in Table B.2, multiply the result by $w \cdot d^{-1} \cdot \alpha_{min}^{-1}$. If bigger, then multiply by $w \cdot d^{-1} \cdot \alpha_{max}^{-1}$.

Step 2 - Determine LED forward current

The second step is to find the current ($I_{e,safe}$) that drives the LED to emit the lower of the two calculated MPEs. This can be done using the maximum radiant intensity ($I_{e,max}$) and the curve of the relative total radiant flux ($f(I_f) = \Phi_e/\Phi_e(I_{f,max})$). For simplicity, we will write the relative total radiant flux as just Φ_e.

The current necessary to reach the MPE will be on the abscissa of the graph. The coordinate is given by r, the relative reduction (or increase) in emission, and is given by

$$r = \frac{min\left(I_{IR}, I_{IR}^{EL}\right)}{I_{e,max}}. \tag{C.3}$$

Figure C.1 shows a typical curve. The emitted power is in general well approximated by a linear function of the forward current. The value we look for is on the green area (only if $r \leq 1.0$). In case $r > 1.0$, the LED cannot reach the MPEs on the long term, and is safe to be operated at the maximum current allowed for continuous operation.

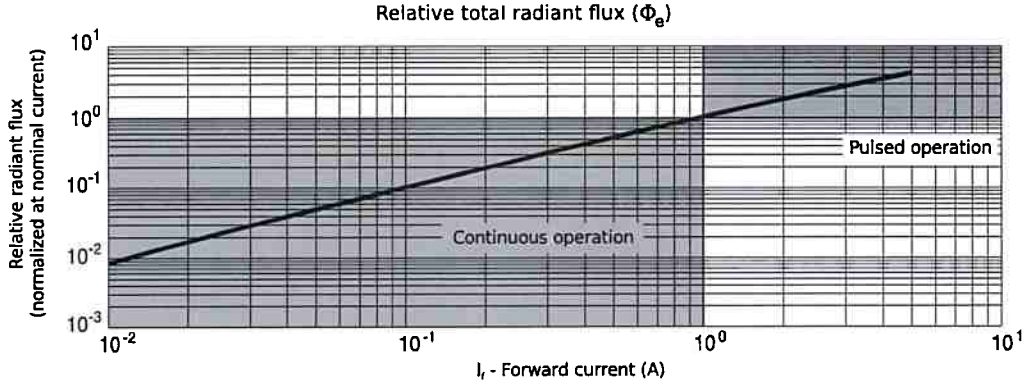


Figure C.1: *Power IR LED typical relative radiant flux (Φ_e) curve according to forward current.*

Step 3 - Pilot test

This work target low cost setups which in general are not well documented, or come from other disassembled devices, such as webcameras modded to remove *infrared* (IR)-blocking filters and to support lenses with higher focal length. In addition, such modded cameras employ RGB sensors which present some sensitivity on the NIR spectrum, but the response in not guaranteed in any way. Thus, we propose a pilot test to determine the amount of radiant illumination necessary to produce high quality images.

The pilot test involves indirectly measuring the illumination necessary using a prototype. This is done by keeping the bright or dark pupil illumination active all the time. Accordingly, the exposure is set to maximum in order to capture light during all the frame period. To improve the signal to noise ratio, the camera gain must be reduced. With the camera ready, start increasing the voltage which feeds the illuminator until the frame present the desired characteristics.

Care must be taken to not exceed the maximum LED ratings or the MPEs, which comes first. This step requires the measurement of the LED current, which can be done by placing an ammeter in series with the LED or with a voltmeter, measuring the potential difference between the current limiting resistor terminals. The second approach is preferred as the current can be measured while the device is in operation, without the possibility of a reduced current reading due to ammeter lead wire resistance and inductance. Figure C.2 shows the placement of the meter for both approaches.

Using the second approach, the LED current ($I_{f,\Delta t}$) can be found using the first Ohm's law, as follows

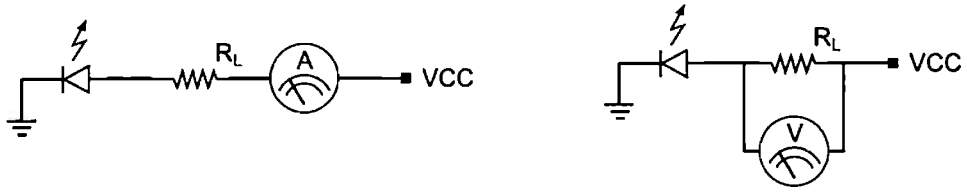


Figure C.2: *Measuring the LED forward current using an ammeter (left) and an voltmeter (right).*

$$I_{f,\Delta t} = \frac{U_{\Delta t}}{R_L \cdot n} \tag{C.4}$$

where $U_{\Delta t}$ is the measured voltage drop at current limiting resistor R_L, and n is the number of LEDs arranged in parallel. If more than one LED is arranged in series, n must be set to one.

Step 4 - Find radiant intensity at pilot test

With the LED current found in the pilot test ($I_{f,\Delta t}$) we will estimate the relative drop in the total radiant flux. This is done by inspection of the graph $f(I_f) = \Phi_e/\Phi_e(I_{f,max})$, replacing I_f by the found $I_{f,\Delta t}$. Assuming the radiant flux is directly related to the radiant intensity by a constant c^2, the same ratio can be applied to the radiant intensity at the nominal current ($I_{e,max}$) to find the setup radiant intensity demands, which we will call $I_{e,\Delta t}$. The green area of the graph on Figure C.1 will be used again.

Step 5 - Find minimum duty cycle

Now we will find the minimum pulse length with the same averaged power inside the safe range. The found $I_{e,\Delta t}$ will be increased inversely proportional to the reduction in pulse length. The radiant intensity for a pulse length Δ_{strobe} can be expressed as

$$I_{e,\Delta strobe} = \frac{I_{e,\Delta t} \cdot \Delta t}{\Delta_{strobe}} \quad mW/sr \tag{C.5}$$

The MPE for the cornea for exposure times $t \leq 1000s$ is given by (B.5). It can be expressed in terms of the source radiant intensity as follows

$$I_{IR} = 18d^2 \cdot n^{-1} \cdot t^{-0.75} \quad mW/sr \tag{C.6}$$

where the distance (d) is given in millimeters, t denotes the time in seconds and n denotes the number of LEDs turned on simultaneously. We can find the minimum pulse length which makes our setup to exceed the MPE by replacing I_{IR} in (C.6) by $I_{e,\Delta strobe}$ from (C.5) and t by Δ_{strobe}, as follows

$$\Delta_{strobe,c} = \sqrt[0.25]{\frac{\Delta t \cdot I_{e,\Delta t} \cdot n}{18d^2}} \tag{C.7}$$

where Δt is given in seconds, $I_{e,\Delta t}$ in mW/sr and d in millimeters.

Accordingly, the MPE for retina given in (B.9) for short exposures (for $t = 10\mu s$ to $10s$) becomes

[2]This is feasible because the radiant flux considers the power emitted in all directions, while the radiant intensity considers the power emitted in a very narrow area.

$$I_{IR}^{EL} = 50dw \cdot t^{-0.25} \cdot R(\lambda)^{-1} \quad mW/sr \tag{C.8}$$

with the distance (d) and the die side (w) given in millimeters, t in seconds and $R(\lambda)$ defined as in (B.7). Doing similar substitutions we find

$$\Delta_{strobe,r} = \sqrt[0.75]{\frac{\Delta t \cdot I_{e,\Delta t} \cdot R(\lambda)}{50dw}} \tag{C.9}$$

where Δt is given in seconds, $I_{e,\Delta t}$ in mW/sr, d and w in millimeters.

The formulations (C.8) and (C.9) are valid for subtended angles inside the range $[\alpha_{min}, \alpha_{max}]$. If not, replace d by $w \cdot \alpha$, with alpha taken from Table B.2.

The minimum pulse length safe to the eyes considering the increase in power due to the reduction in duty cycle is given by $max\,(\Delta_{strobe,c}, \Delta_{strobe,r})$. The reduction in pulse length implies in a considerable increase in radiant intensity ($I_{e,\Delta strobe}$), which can be found using (C.5). Assuming a constant relation between the radiant intensity and the total radiant flux, we can look on the relative total radiant flux graph for the coordinate $I_{e,\Delta strobe}/I_{e,max}$. If the graph is defined for the value, the LED is capable of delivering the power at the indicated forward current. In case the graph is undefined for the value, we must increase the pulse length to the operating area.

To do that, we will find the $I'_{e,\Delta strobe}$ given by the maximum relative radiant flux (taken from the graph) multiplied by the nominal radiant intensity $I_{e,max}$. Then, replace $I_{e,\Delta strobe}$ by $I'_{e,\Delta strobe}$ on (C.5) to find the new Δ_{strobe}. Find the duty cycle by dividing the strobe length by the frame period (Δt) and check on the pulse handling capability plot of the LED if it can handle the maximum current at such conditions.

Figure C.3 shows a sample pulse handling capability curve. The forward current a LED can handle depends primarily on efficiency and heat transfer capacity. Thus, the duty cycle and the on period (t_p) determine the maximum power the LED can handle. A color coded duty cycle curve of a pulse at 187Hz was superimposed to highlight the interaction between the power handling and duty cycle.

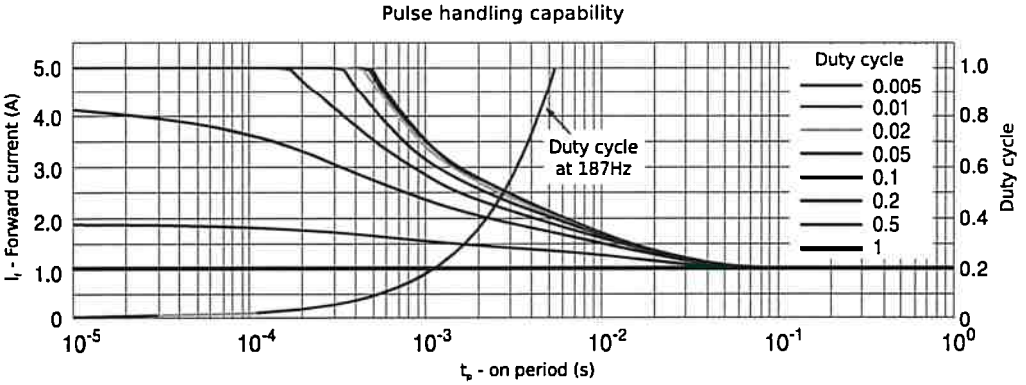


Figure C.3: *Power IR LED sample pulse handling capability curves according to duty cycle and on period. Superimposed is the duty cycle curve at a fixed frequency of 187Hz.*

C.2 Sample design

Our eye tracker hardware is composed by a camera and illuminators for the bright and dark pupil. Each illuminator is composed by two LEDs OSRAM SFH4715 (OSRAM Opto Semiconductors, 2014a). The lens employed on the camera assume a distance to the subject of $13cm$, and the LED die size is $1 \times 1mm$. For details on the prototype refer to Section E.3.

The first step is to calculate the MPE considering those setup parameters. The limit for the cornea is given in (C.1) and results in $I_{IR} = 845mW/sr$. The limit for retina is given in (C.2) and results in $I_{IR}^{EL} = 1090mW/sr$. Note that the subtended angle α of the setup is smaller than the minimum, and thus, the result was multiplied by $130^{-1} \cdot 0.011^{-1}$.

The second step is to calculate the maximum forward current of the LEDs. The maximum radiant intensity of the most efficient batch is $I_{e,max} = 500mW/sr$ at the nominal maximum current of $I_f = 1A$ for continuous operation. In this particular case, the radiant intensity is lower than the two calculated MPE and thus, no additional calculations are necessary ($r = 1.69$ in C.3). The device is safe to be operated continuously using the maximum rating. In case the LED is enable to produce a radiant intensity higher than any of the found MPEs, the maximum current must be reduced to keep the emission in the safe range.

The third step is to perform a pilot test to determine the amount of illumination necessary given the particular setup. The camera was configured to output frames at *quarter video graphics array* (QVGA) resolution and 187Hz sample rate. The gain was adjusted to 5 (a small value for this camera) and the exposure to maximum. Brightness and contrast were configured at default values of 0 and 32, respectively. The input voltage was initially set to $1.2V$ and gradually increased while the image quality and the forward current at the LEDs were monitored. The images produced were considered satisfactory when the current of $120mA$ was reached, resulting in $I_{f,\Delta t} = 60mA$.

The fourth step is to find the drop in radiant flux due to the current employed to infer the drop in radiant intensity. By looking at the relative total radiant flux graph of the LED datasheet, we find a drop equivalent to $6 \cdot 10^{-2}$. Thus, $I_{e,\Delta t} = 30mW/sr$ per LED.

Fifty step. Using (C.7) with $\Delta t = 1/187$, $n = 2$ and $d = 130$, we find $\Delta_{strobe,c} = 1.23 \cdot 10^{-24}s$. Using (C.9) we find $\Delta_{strobe,r} = 285ns$. Using the limit established by the ICNIRP (replace 50 by 20 in (C.9) and use the α restrictions as defined in Table B.2), we find $\Delta_{strobe,r} = 2.52\mu s$. As this last one is more restrictive, we will keep it as the target minimum pulse length.

Using (C.5) with $2.52\mu s$, $I_{e,\Delta t} = 30mW/sr$ and $\Delta t = 1/187s$, we find $I_{e,\Delta strobe} = 6.36 \cdot 10^4 mW/sr$, which is 127 times the maximum radiant intensity at the nominal forward current. Visual inspection of the *relative total radiant flux* graph reveals that the radiant flux can be increased by about 4.3 times by mean of an increase in current. We will find the maximum radiant intensity that can be achieved by increasing the current to maximum by simply multiplying 4.3 by $I_{e,max}$ to find $I'_{e,\Delta strobe}$. Using (C.5) we find a minimum $\Delta_{strobe} = 74.6\mu s$. Such pulse length corresponds to a duty cycle of 0.014 (1.4%). The *pulse handling capability* curve for the closest higher duty cycle (0.02) of the device shows it can work safely at the defined conditions.

Therefore, in our setup at 187Hz, continuous illumination fed with a current of $60mA$ can be replaced by stroboscopic illumination with a pulse length of $74.6\mu s$ with a burst current of $5A$. The images produced are expected to be similar in terms of brightness, but the real integration time of $5347\mu s$ is replaced by a virtual one of just $74.6\mu s$. The application is safe to the eyes by a large margin considering both the IEC standard and the ICNIRP guidelines.

Once the parameters are set, it is time to move to the board configuration. The pulse length must be setup before changing the LED current. In the board of our sample design, the current is controlled by the input voltage and an current limiting resistor. While it is possible to calculate the target input voltage to produce a given current, considering the resistor and the LED forward current curve, we do not advise such approach. There are uncertainties associated to each component the current flows through. For example, the LED current curve considers a single pulse $100\mu s$ wide and is representative of a typical performance. Another example is the *metal-oxide-semiconductor field-effect transistor* (MOSFET) transistor used to switch the LEDs on, which has an on resistance ($R_{DS(ON)}$) dependent on several factors, such as temperature, gate voltage, gate to source voltage. As the current pulse is considerable, any wire can act as an inductor, changing the pulse profile as well.

Considering all the interference sources, a direct measurement is preferred. The LED current is measured by placing an oscilloscope instead of a voltmeter between the limiting resistor terminals (see Figure C.2). We can calculate the voltage drop on the resistor beforehand using the Ohm's law, such that $V_{drop} = R_L \cdot I \cdot n$. In our particular setup, $I = 5A$, $R_L = 0.25$ and $n = 2$, which gives $V_{drop} = 2.5V$. The oscilloscope shows the profile of the voltage drop while we adjust the input voltage.

C.3 Failure mode and caveats

We have used a microcontroller to switch the LEDs on and off. The period and duty cycle is controlled by a program which accepts commands to change the values as desired. During the microcontroller initialization, the pins can assume voltage levels which might activate the LEDs for a period much higher than specified by our pulse length minimization design. Thus, the standard procedure is to turn on the microcontrollers followed by the illuminators. To turn off, the order is reversed.

We strongly advise the use of quick-acting fuses or resettable PTC fuses to avoid accidental long exposures, which might be over the maximum permissible exposure. With respect to the LEDs, such fuses can avoid them from burning, however, the characteristics of the LEDs are unpredictably changed, as the time a fuse takes to act allows a surge current much larger than the maximum rating for the time frame.

We have used a 500mA rated PTC fuse (model MF-MSMF050) (Bourns C.P. Division, 2006). It keeps the emissions inside the safe range in case of accidental long pulses, which occurs for example, during the microcontroller initialization. The fuse breaks the circuit with $9ms$ at $10A$ (2 LEDs at $5A$) and with $2ms$ at $20A$ (4 LEDs)[3].

[3]Typical performance data from time to trip curve, done in a simulated application environment.

Appendix D

Stroboscopic controller implementations

The asynchronous stroboscopic light controller can be implemented in a number of ways. Two approaches are presented, one based on the classic 555 analog timer and one based on an Arduino board. The 555 timer is a versatile and low cost functional block which can operate as an oscillator or time delay switch with minimal discrete components. The Arduino board, on the other hand, is a general-purpose microcontroller board which eases prototipation, as it incorporates power conditioning filters, a crystal oscillator and USB interface to an Atmel microcontroller, making nearly all input and output available to use through $2.54mm$ sockets.

The timer based solution needs an additional integrated circuit to allow the switching between the bright and dark pupil illuminators and presents a limited accuracy when compared to the microcontroller solution. The Arduino-based approach is easier to implement, as the external circuitry besides the microcontroller is minimal, necessary only to handle the high current demands of the *light-emitting diode*s (LEDs).

D.1 Using a 555 timer

The 555 timer is a popular analog circuit introduced in 1972 capable of generating oscillating frequencies from half a cycle per hour up to 500KHz. It also allow the adjustment of the duty cycle from less than one percent up to 99 percent. The 555 timer consists of two voltage comparators, a bistable flip-flop, a discharge transistor, and a resistor divider network. It exhibits a typical timing accuracy of 1% with a $50ppm/C$ timing drift with temperature (Philips Semiconductors, 1988).

In this work we are interested on the so called astable or free-run operation, which is the oscillatory mode. In such mode, the frequency depends on the value of two resistors, R_A and R_B, and a capacitor C (refer to the green area of Figure D.1) The frequency is given by the time it takes to charge and discharge the capacitor at a given level.

The clock period (Δ_{clke}) generated by the circuit in Figure D.1 is given by (Philips Semiconductors, 1988)

$$\Delta_{clke} = \frac{(R_A + 2R_B)\,C}{1.49} \tag{D.1}$$

and the duty cycle D given by (Philips Semiconductors, 1988)

$$D = \frac{R_A + R_B}{R_A + 2R_B} \tag{D.2}$$

	R_A / Frequency error				
	Strobe length (μs)				
Frame rate	100	200	400	600	800
30	499K/-1.06%	487K/0.73%	221K/-0.32%	215K/1.20%	215K/-0.10%
60	243K/0.96%	243K/-0.28%	107K/0.35%	105K/-0.14%	102K/-0.09%
100	147K/-0.63%	143K/-0.01%	61.9K/0.51%	59K/1.01%	56.6K/0.22%
150	97.6K/-1.21%	93.1K/0.21%	39.2K/1.04%	37.4K/-0.64%	34K/0.38%
187	76.8K/-0.08%	73.2K/0.58%	30.9K/-0.45%	28K/0.49%	24.9K/0.93%
C	100nF	100nF	220nF	220nF	220nF
R_B	1.47K	3.01K	2.74K	4.02K	5.49K
Strobe error	98.49%	100.84%	100.97%	98.76%	101.15%

Table D.1: *Values for R_A, R_B and C for selected frame rates and strobe lengths. The error in frequency and in the strobe length are also given. Standard values for 1% resistors.*

Based on (D.1) and (D.2) we built a table with recommended component values with the expected deviations in the frequency and strobe length. Table D.1 shows the values considering 1% standard resistors. A positive frequency error means the generated frequency is higher than the nominal frame rate. The fixed value resistors can be replaced or mounted in series with potentiometers to allow fine adjustments.

The main components of the circuit are a 555 timer (Fairchild Semiconductor, 2013) and a dual D-type flip-flop (Fairchild Semiconductor, 2000). The timer provides a duty cycle above 50% and is used to provide the ground potential to the infrared LEDs by means of two N-*metal-oxide-semiconductor field-effect transistors* (MOSFETs). The switching between the dark and bright pupil illuminators is done by a flip-flop, which divides the input timer frequency in half and provides a 50% duty cycle. P-MOSFETs with separate supplies are used to provide the LEDs with positive potential.

A switch is provided to allow selecting between the 1×1 or the 2×2 design (refer to Sections 3.3.1 and 3.3.2). If on the B position, the timer frequency is divided in half and the behavior corresponds to the 1×1 design. When on the A position, two flip-flops divide the frequency by four, resulting in a 2×2 design.

Note that only the main components are shown in Figure D.1. Bypass capacitors were omitted but are mandatory, as ripple on the supply line can cause loss of timing accuracy for the 555 timer and malfunction of the flip-flops. Note also the MOSFETs employed are suggestions and can be replaced to address specific demands. For example, the minimum VCC' and VCC" can be reduced down to $1.8V$ (for at most $8A$ current drain) by employing the FDN304P instead of the AO4407 power MOSFET IC.

D.2 Using a microcontroller board

The microcontroller based solution utilizes two outputs to drive either the bright or dark pupil illuminators, thus avoiding the use of an external flip-flop, and allowing arbitrary combination of pulse sequences. The power interface is done with a dual N-MOSFET IC NTGD3148N, which is capable of delivering $3.5A$ continuously and up to $10A$ for pulses $10\mu s$ long in a very small package of just $3 \times 1.5mm$ (ON Semiconductor, 2008).

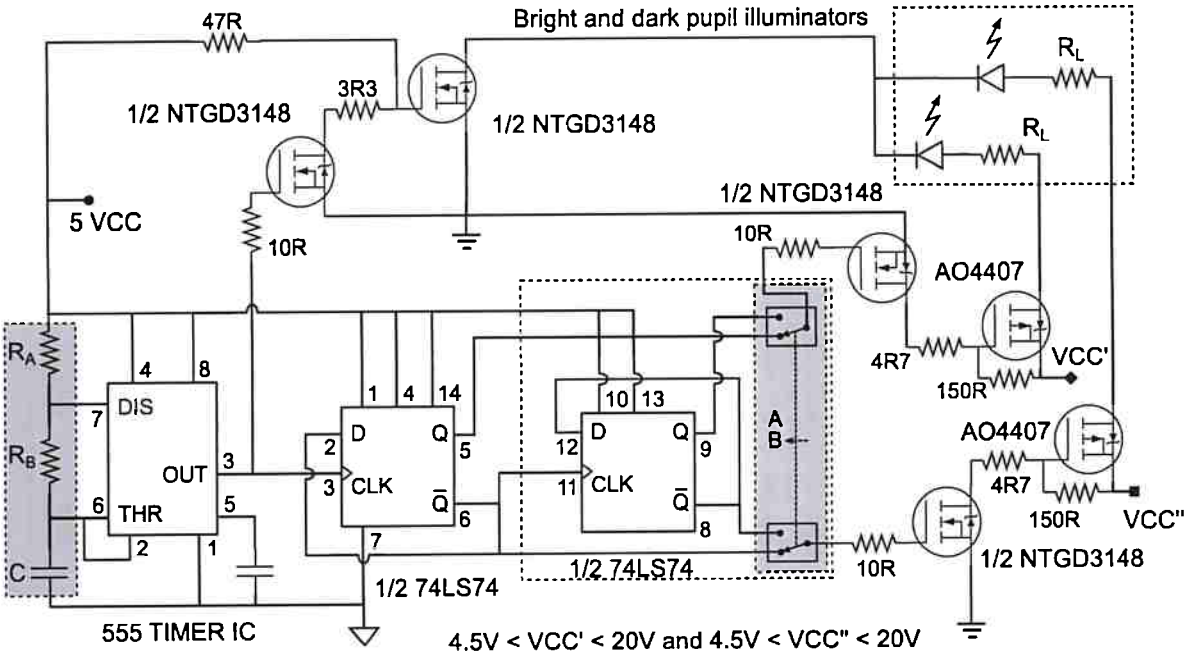


Figure D.1: *Analog circuit based on the 555 timer IC to provide the pulses for the LEDs. Three voltage sources are shown, in case the illumination draws too much current or use another electric potential.*

The circuit is shown in Figure D.2 and its operation is simple. Two microcontroller outputs are connected each to a gate of a MOSFET which acts like a switch. When the gate receives a potential higher than $1.5V$[1] the MOSFET closes the circuit, feeding the LEDs with negative potential. This circuit allows any voltage to feed the LEDs, from $0.5V$ up to $20V$, including different potentials if the design employs different diodes for each illuminator.

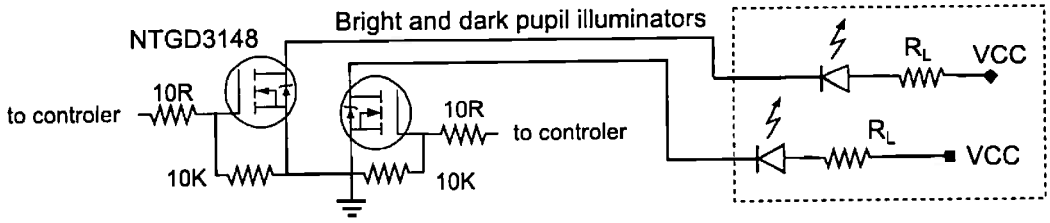


Figure D.2: *Using a microcontroller to provide the pulses for the LEDs. Two voltage sources are shown, one for each LED.*

[1]The drain current depends on the voltage applied. As the circuit is designed to interface 5V microcontrollers, the expected potential is enough to fully switch the power.

Appendix E

Hardware prototypes

E.1 Synchronous implementation

The structured light source has two main parts. The boards equipped with the LEDs (mounted at the camera plane), and the power circuit (responsible for lighting the LEDs). The electronic circuit that drives the infrared lights is shown in Figure E.1. The LEDs are arranged in series and are driven by a low internal resistance N-MOSFET IC. The voltage is controlled by a separate board based on the LM2596 switching power converter IC. In total, sixteen OSRAM SFH4050 OSRAM Opto Semiconductors (2014b) LED chips are used.

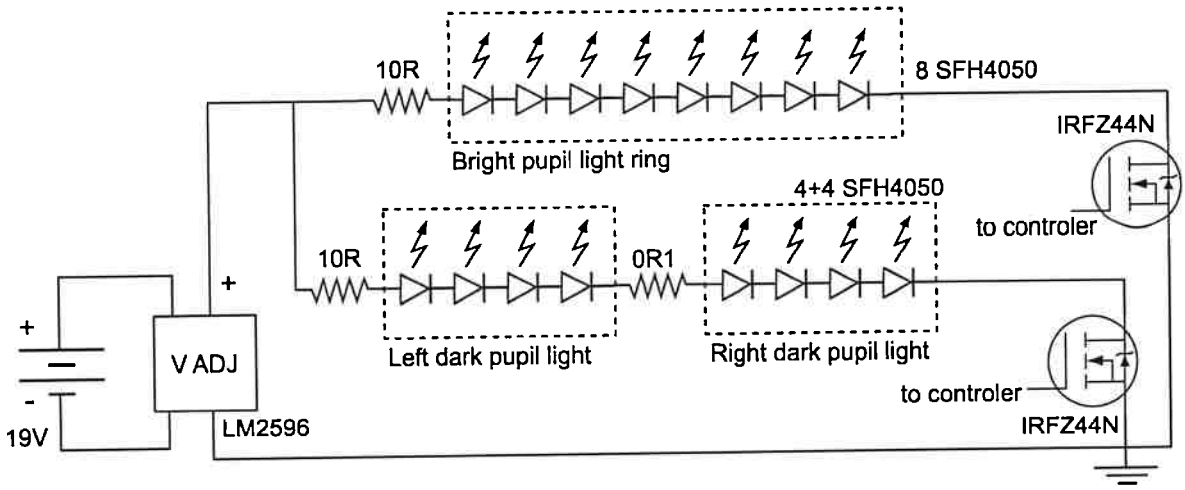


Figure E.1: *Light emitting diodes power board*

The on-axis LEDs, responsible for the bright pupil effect, were soldered to a ring with $13mm$ of external diameter, and mounted directly onto a CCTV 12mm lens. The off-axis LEDs were soldered to a board adapted to the lens mount, divided into two groups $45mm$ apart of each other. Figure E.2 shows the actual prototype.

The controller is built around Arduino Boards that controls the pulses which drive the LED boards. The pulse is triggered by an external interrupt pin on the board or by an internal timer (clk_e). As part of the interrupt routine, the microcontroller configures an internal timer to turn off the LEDs after a desired period (Δ_{strobe}).

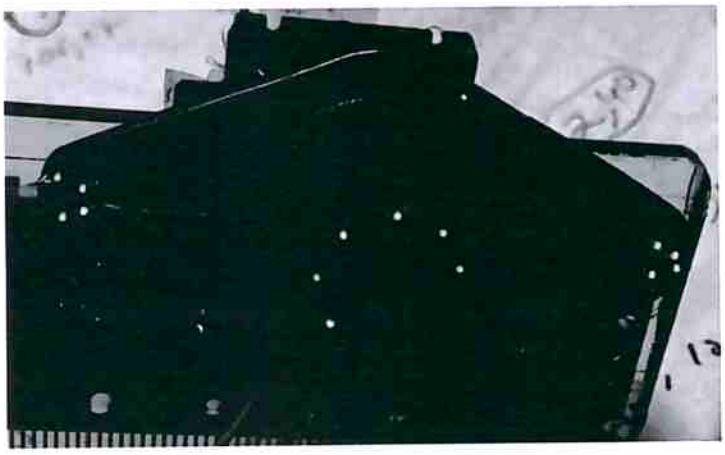


Figure E.2: *Actual LED boards.*

E.1.1 Eye safety assessment

E.1.1.1 Exposure limit for the cornea

In the experiments conducted, at most eight *light-emitting diodes* (LEDs) were on at once. From the diode specification, the maximum radiant intensity ($I_{e,max}$) at $I_F = 100mA$ is $20mW/sr$. With $I_F = 350mA$, the radiant intensity is multiplied by a factor of 3 (see datasheet OSRAM Opto Semiconductors (2014b) diagram $I_e/I_e(100mA)$). The total I_e can be found by multiplying the intensity by the factor and the number of LEDs. This gives $I_{e,tot} = 20 \cdot 3 \cdot 8 = 480mW/sr$. The irradiance E_e can be calculated from the radiant intensity I_e and distance d using the inverse square law (B.6). The distance from the eye to the emitters is approximately $0.13m$, then, $E_{IR} = 28.40W/m^2$. The irradiance found considers continuous emission, which is not the case. In experiments, the lights are kept on for $800\mu s$ in each frame time. This gives a variable duty cycle based on the frame rate. For those cases where pulsed lights are employed, a weighted radiant exposure is used against the limits IEC (2006). The worst case occurs at the fastest frame rate, which is 187 Hz. The duty cycle is then $D = 0.0008 \cdot 187 = 0.1496$, and the irradiance becomes $E_{IR} = 28.40 \cdot 0.1496 = 4.25W/m^2$. The value found for the averaged irradiance ($E_{IR} = 4.25W/m^2$) is much lower than the maximum allowed ($100W/m^2$).

E.1.1.2 Exposure limit for the retina

The die size for the diode considered is $0.3mm \times 0.3mm$ and the angular subtense (α) for a distance $d = 130mm$ can be approximated by $\alpha = 0.3mm/130mm = 0.0023$.

As the LEDs are driven in pulsed mode, the radiance must be compared against the limit for the two time scales: a) considering the single pulse; and b) considering the averaged pulses for a longer exposure.

Radiance of a single pulse

Considering the pulse duration of $800\mu s$, the maximum radiance allowed using (B.9) is $L_R^{EL} = 1.29 \cdot 10^5 mW/mm^2/sr$.

The actual exposure can be found using (B.13) and $I_e(350mA) = 60mW/sr$. Then, $L_R = I_e \cdot R(\lambda)/A = 60mW/sr.0.5/(0.3mm)^2 = 3.33 \cdot 10^2 mW/mm^2/sr$. This value is much lower than the safe limit.

Radiance considering a long term averaged emission

The allowed radiance is the one given in (B.8) with the effective minimum subtended angle of $0.011rad$ (see Table B.2), which gives a limit of $L_{IR} = 545.5mW/mm^2/sr$. The radiance based on datasheet values and considering the duty cycle of approximately 15% is $L = 49.8mW/mm^2/sr$, tenfold bellow the exposure limit.

E.2 Dual-camera synchronous implementation

A system based on the *differential stroboscopic lighting* (DSL) technique is composed by a camera, a structured light source, a controller to trigger the lights and a computer. Instead of connecting the lights directly to the camera, a separate computer feed controller is used (see Figure 2.2 for details). In addition to that basic configuration, a second camera is employed on the constructed prototype.

The prototype was developed to evaluate the technique and compare its performance over dark pupil techniques which employ constant illumination. Thus, to capture two independent images simultaneously of the subject's eye, independent light sources and cameras were employed.

A CS mount $6-60mm$ varifocal lens, with a $27mm$ objective was used to capture the images. A long pass filter of $760nm$ were mounted in front of the lens. Two independent illumination systems were used, one structured to produce both bright and dark pupil images, and the other to produce only dark pupil images.

The structured illumination system were driven by the stroboscopic controller and two light sources. On the lens optical axis, an infrared illuminator composed by four EPISTAR dies ES-SASFPN42D (Epistar Corporation, 2014), was placed to induce the bright pupil effect. The off-axis illuminator was built around two high power LEDs equipped with four EPISTAR dies each, $58mm$ apart from each other. Both LEDs present a center frequency of $850nm$, and were filtered using matching narrow bandpass filters, with a half bandwidth of $30nm$.

The constant illumination system was built around a LED equipped with four EPISTAR ES-SAUFPN42A dies (Epistar Corporation, 2012), with $940nm$ center wavelength, driven by a constant source. A plano-convex lens of $20mm$ focal length was used to improve the LED efficiency. This source was also filtered using a narrow bandpass filter centered on the emitter wavelength. The safety of using such illuminators is discussed in Section E.2.1.

After leaving the lens, the captured light is separated into two paths by a custom made beam splitter. Each path ends in the sensor of a PS3 Eye camera (Sony, 2014). Such camera was chosen due to the low cost and reasonable performance, being able to sustain a $187Hz$ in a Linux system. Before reaching the camera sensor, the light pass through narrow bandpass filters which match the illuminators. One centered on $850nm$ for a camera and the other centered on $940nm$ for the other. Thus, the eye image is captured from the same perspective using both the stroboscopic technique and constant illumination. Figure E.3 shows the setup described.

The two cameras are synchronized in hardware. The frame synchronization input (FSYN) of one camera is fed by the synchronization output (VSYNC) from the other[1].

The constant illuminator is driven by an independent constant voltage source built around a LM2596 switching power converter *integrated circuit* (IC) with a current limiting resistor. The

[1]See the camera synchronization report for details.

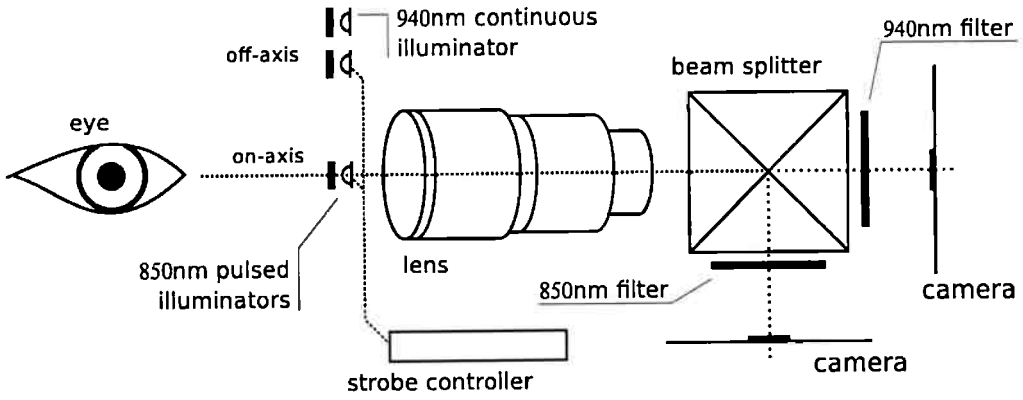


Figure E.3: *Prototype constructed to compare the accuracy of two eye trackers, one employing the DSL with stroboscopic light and the other using algorithms based on the dark pupil and corneal reflection using continuous illumination.*

voltage was reduced as well as the resistance to limit the power dissipation at the resistor. The structured illuminator have two independent power sources, one for each group of LEDs. The LEDs are arranged in series and are driven by a low internal resistance NTGD3148N dual N-Channel *metal-oxide-semiconductor field-effect transistor* (MOSFET) IC (ON Semiconductor, 2008). The MOSFET IC was chosen instead of common NPN or PNP transistors due to their low internal resistance and lower gate current requirements, allowing for a simpler linkage to the controller. The electronic circuit that drives the *infrared* (IR) lights is shown in Figure E.4.

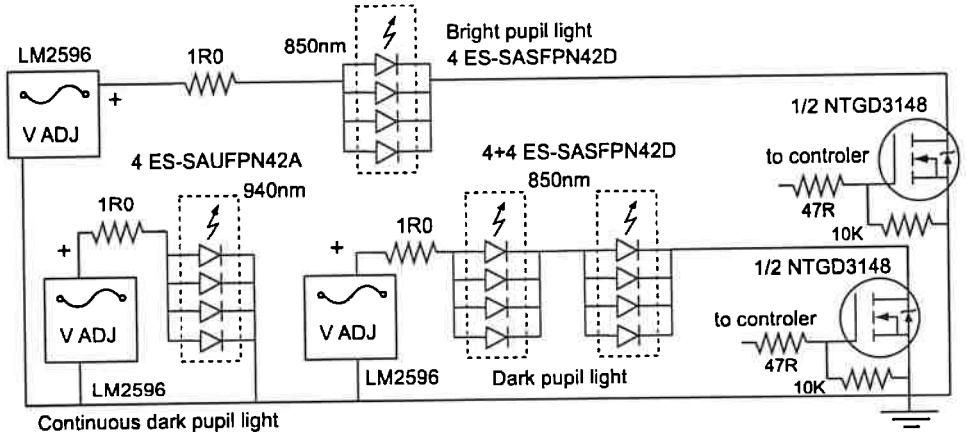
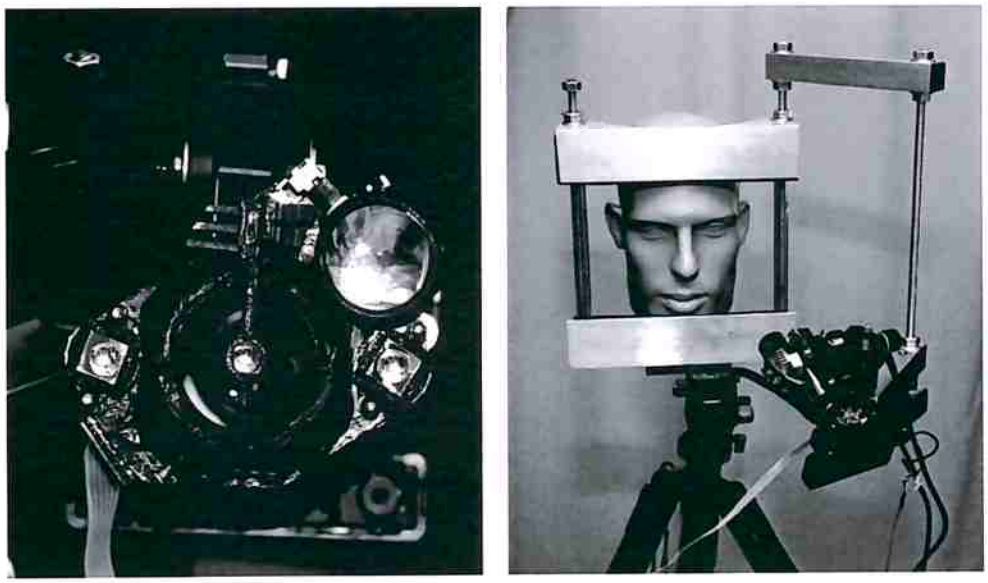


Figure E.4: *Light emitting diodes power board. Three voltage sources are used to allow the adjust of the mean frame illumination without changing the gain of the cameras.*

The strobe controller is built around two Arduino boards. The strobe is triggered by an external interrupt pin on the first board which can be driven by either the camera VSYNC signal or the second microcontroller board. This second board runs a 16-bit timer with no prescaling, resulting in a clock resolution of $62.5ns$. The board is connected to the computer which sends the updated clock period when needed. The strobe is kept on for a period of time (Δ_{strobe}) controlled by the first board, which is also responsible for switching between the on and off-axis LEDs as the interrupts are set. The choice for two boards made it easy to change from true camera synchronous operation to the DSL during prototipation, as the second board is used solely for triggering the interrupt in place of the camera (clk_e). The boards' firmware are available to download at the authors website (to be released upon acceptance of the paper).



a) b)

Figure E.5: *Actual prototype. a) Frontal view of the camera lens and illuminators; b) Camera mounted on a head-rest.*

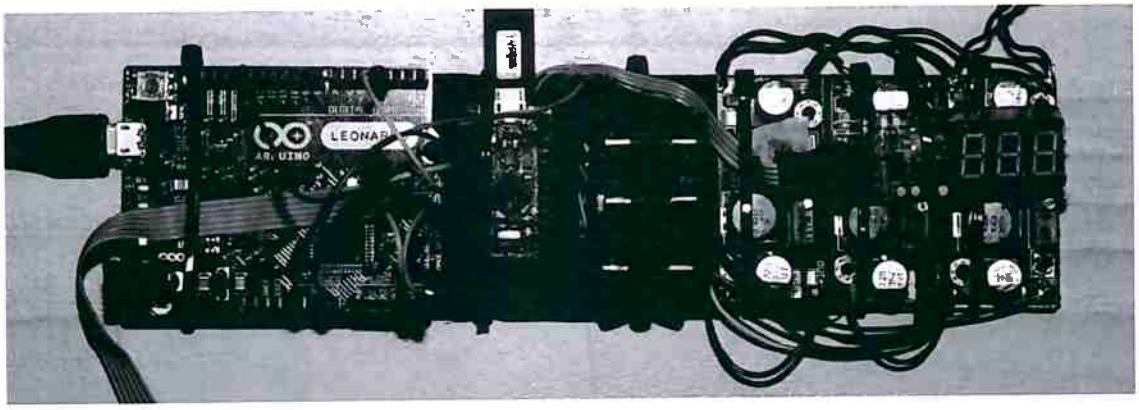


Figure E.6: *Strobe controller boards and independent voltage sources.*

The cameras and the microcontroller boards are connected via USB ports to the computer. A computer software capture the images from the cameras and adjusts the external clock (clk_e) when needed.

E.2.1 Eye safety assessment

There are several standards and guidelines developed by international organizations associated to the light hazard assessment (Altkorn *et al.*, 2005). The most relevant ones have their maximum allowed exposures discussed on the Appendix B.

It has already been shown that the illumination necessary for a typical eye tracking equipment stays well below the established limits (Mulvey *et al.*, 2008). However, our prototype introduce losses not typical of a camera-filter arrangement. Ideally, a beam splitter in the *near-infrared* (NIR) range would transmit from 41% to 45% of the light. CMOS sensors are also less sensitive as the wavelength extends over the 900nm range. Taken together, such characteristics make our prototype more light demanding.

Despite the short experimental sections, the constructed prototype employs powerful infrared LEDs. Thus, before starting any experiment with human subjects, we adjusted the power needed to produce clear frames using artificial eyes. With the illumination adjusted, the total power emitted was calculated according to the power drawn and the worst case (more efficient) device specifications.

E.2.1.1 Exposure limit for the cornea

The experimental sections were designed to take between 100 and 130 seconds. As a safety margin, we consider sections no longer than $200s$. For exposures smaller than $1000s$, the safe limit for the cornea is time dependent, and given by (B.5). Thus, the exposure limit for our sections, taking $t = 200s$ become $E_{IR} = 338.45W/m^2$.

Our prototype employ lenses in front of the LEDs, which changes the propagation pattern. As a safety margin, we assumed that our lenses capture all the radiant flux produced and project it at a given radiant intensity ($I_{e,max}$). To find $I_{e,max}$, we measured the size of the beam projected at the nominal distance of our prototype to the face, $d = 0.2m$. The measure gave a diameter of about $0.12m$, but as the intensity varies as we become far from the center, we considered a diameter of $0.06m$. With that, we found that our illumination propagate on a conical section of $0.0707sr$. Now, we are able to convert the total radiant flux into radiant intensity.

The radiant flux were estimated based on the current consumption of the LEDs and the *current* × *relative total radiant flux* curve. The current was measured indirectly with the help of an oscilloscope. The voltage drop at a precision current limiting resistor allowed the estimation of the current flowing throw it. We found that for the 940nm illuminator, the maximum radiant intensity was $I_{e,max,940} = 2.425W/sr$. For the $850nm$ illuminator, the maximum radiant intensity was $I_{e,max,850} = 24.575W/sr$ for the bright pupil illuminator.

The irradiance E_e can be calculated from the radiant intensity I_e and distance d (for die size $\ll$ d) using the inverse square law (B.6). The distance from the eye to the emitters is $0.20m$, then, $E_{IR,940} = 60.625W/m^2$ and $E_{IR,850} = 614.375W/m^2$.

The irradiance found considers continuous emission, which is not the case for the 850nm illuminator. In the experiments, the lights were kept on for $800\mu s$ in each frame time. This gives a variable duty cycle based on the frame rate. For those cases where pulsed lights are employed, a weighted radiant exposure is used against the limits (IEC, 2006). The frame rate employed was $60Hz$, then the duty cycle is $D = 0.0008 \cdot 60 = 0.048$, and the irradiance becomes $E_{IR,850} = 614.375 \cdot 0.048 = 29.49W/m^2$.

The value found for the averaged irradiance ($E_{IR} = 29.49W/m^2$) plus the irradiance of the continuous illuminator ($E_{IR}^{940} = 60.625W/m^2$) is inside the range allowed for $t \leq 200s$ ($90.115 < 338.45W/m^2$). At $60Hz$, the emitted power is bellow the limit for continuous exposure also ($90.115 < 100W/m^2$, for $t > 1000s$).

E.2.1.2 Exposure limit for the retina

Besides the exposure time, the limits for retina are associated with the pupil diameter, the wavelength, and the emitting source size. The wider the pupil, greater the risk as more light is allowed to enter the eye. As the NIR radiation presents low to none visual stimulus, a fully open $7mm$ pupil diameter is assumed. As the danger to the retina is basically thermal, the exposure limits are

bound by a so-called burn hazard weighting, calculated using (B.7). Accordingly, $R(850nm) = 0.50$ and $R(940nm) = 0.33$.

The emitter is focused by the eye lenses which projects an image into the retina. That small spot receiving light at the retina depends on the size of the emitter and its distance (these two variables define the subtended angle (α), the angle in which the rays enter the eye). Due to the inherent eye movements and the dynamic heat dissipation of the tissue, there is a lower and higher bound for α as shown in Table B.2.

The die size for the diode considered is $1mm \times 1mm$. However, the lens leads to a larger virtual die size. The $850nm$ illuminators have lenses with $3mm$ radius, which lead the die size to be viewed as $3.34mm$ wide. The $940nm$ illuminator has a lens with $10mm$ radius, and the die is viewed as being $8.35mm$ wide. We will calculate two angular subtenses (α_{850} and α_{940}). For a distance $d = 200mm$, α_{940} can be approximated by $\alpha_{940} = 8.35mm/200mm = 0.0418$ radians. And $\alpha_{850} = 3.34mm/200mm = 0.0167$ radians.

Considering the α obtained, and using (B.8) and (B.9), we find the exposure limits to be $L^{EL}_{IR,940} \leq 143mW/mm^2/sr$, $L^{EL}_{IR,850} \leq 359mW/mm^2/sr$ and $L^{EL}_{R,850} \leq 17.8 \cdot 10^4 mW/mm^2/sr$ for a pulse duration of $800\mu s$.

As the $850nm$ illuminators are driven in pulsed mode, the radiance must be compared against the limit for the two time scales: a) considering the single pulse; and b) considering the averaged pulses for a longer exposure. The $940nm$ illuminator is evaluated only on the long term.

Radiance of a single pulse

The actual exposure can be found using (B.13) and $I_{e,max,850} = 24.575W/sr$. Then, $L_R = I_e \cdot R(\lambda = 850nm)/A = 2.4575 \cdot 10^4 mW/sr \cdot 0.5/11.1mm = 1.1 \cdot 10^3 mW/mm^2/sr$.

Radiance considering a long term averaged emission

The radiance of the pulsed illuminator is $L_{R,850} = 1.1 \cdot 10^3 mW/mm^2/sr$. Considering the duty cycle of approximately 4.8% gives an averaged radiance of $L_{R,850} = 53.2mW/mm^2/sr$. The radiant intensity of the $940nm$ illuminator is $I_{e,max,940} = 2.425W/sr$. Using (B.13) we find a radiance of $L_{R,940} = 11.5mW/mm^2/sr$. Both radiances are well below the safe limit.

E.2.1.3 Eye safety summary

To avoid any additional exposure due to failure mode effects, fuses close to the average operating current were added in parallel to each LED. In case of erratic pulses or continuous activation, the fuses burn ceasing the emission. Additionally, the minimum distance to the illuminators is assured by a head rest. All the calculations are very conservative, despite the approximations. The radiance found considered all the power coming from a single die, while four are used. The lens cover 80°, while the half emitting angle of the LEDs is ±70°. Thus, there is a considerable amount of power being lost sideways which we assumed to be otherwise focused.

Considering the latest available *maximum permissible exposures* (MPEs) stated by the IEC/EN 62471 and recommendations by *International Commission on Non-Ionizing Radiation Protection* (ICNIRP), and taking into account the approximations, the prototype as proposed and implemented is safe to be used, both for the anterior segment of the eye and the retina. Moreover, according to the standard, the prototype can be classified on the exempt risk group (running at $60Hz$ and considering a standardized distance $d = 200mm$).

E.3 Asynchronous implementation

A prototype was constructed to assess the technique and is the same employed on the sample design calculations, in Section C.2. It is composed by a camera capable of capturing infrared light, a controller built around an Arduino board and illuminators.

The camera employed is a Sony PS3 Eye camera, which delivers 187 frames per second at *quarter video graphics array* (QVGA) resolution (Sony, 2014). The stock lens were replaced by a 16mm focal length lens, adjusted to capture clear images of objects 130mm away. The structured illumination were mounted directly on the lens, as can be seem on Figure E.8a. On the lens optical axis, an infrared illuminator composed by two LEDs OSRAM SFH4715 (OSRAM Opto Semiconductors, 2014a) was placed to induce the bright pupil effect. The off-axis illuminator is similar, but the LEDs were 44mm apart from each other. The LEDs present a center frequency of 850nm, and were filtered using a matching narrow bandpass filter, with a half bandwidth of 30nm.

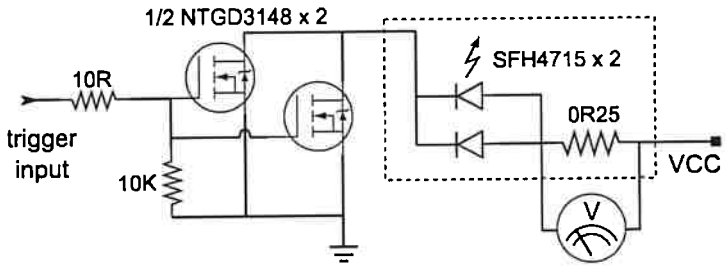
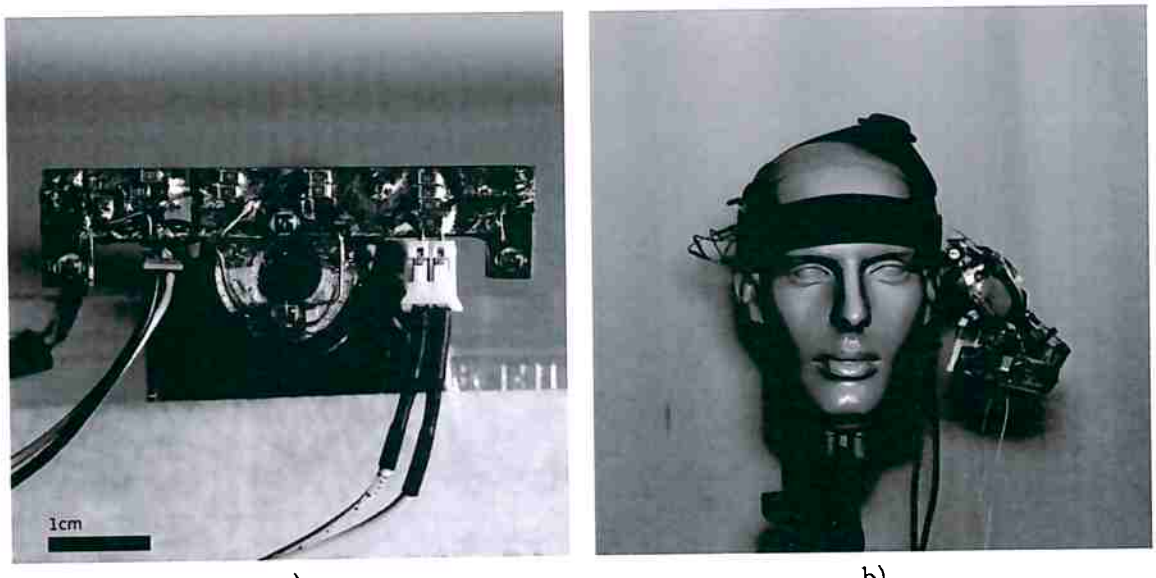


Figure E.7: *Half prototype circuit corresponding to a bright or dark illuminator.*

The structured illuminator is driven by an independent constant voltage source built around a XL4015 switching power converter IC (XLSEMI, 2016). The LEDs are arranged in parallel and are driven by a low internal resistance NTGD3148N dual N-Channel MOSFET IC (ON Semiconductor, 2008). The electronic circuit that drives the infrared lights is shown in Figure E.7.



a) b)

Figure E.8: *Actual prototype. a) Frontal view of the camera lens and illuminators; b) Camera mounted on a head-band.*

The Arduino board was configured to run a 16-bit timer with no prescaling. This timer is used to set the clock period with a resolution of $62.5ns$. Two pins on the board turn on either the bright or dark pupil illuminators. The board is connected to a computer which allows the configuration of the clock period as our will. Figure E.8a shows a picture of the illuminators and drive circuit mounted on the camera lens. The cable connected at the right of the lens is the voltage output which allows the measurement of the LED current. Figure E.8b shows the prototype mounted on a head-band. The Arduino board is also mounted on this band on the opposite side.

E.3.1 Eye safety assessment

The safety was guaranteed by using the method described in the pulse length minimization design, Section C.1. The actual values are presented in Section C.2.

Bibliography

Agustin *et al.* (2006) Javier San Agustin, Arantxa Villanueva and Rafael Cabeza. Pupil brightness variation as a function of gaze direction. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, pages 49–49. ACM. Cited on pag. 87

Altkorn *et al.* (2005) Robert Altkorn, Scott Milkovich and Gene Rider. Light emitting diode safety and safety standards. In *Product Safety Engineering, 2005 IEEE Symposium on*, pages 1–5. IEEE. Cited on pag. 91, 111

Aly and Mohamed (2011) Eman Mohamed Aly and Eman Saad Mohamed. Effect of infrared radiation on the lens. *Indian journal of ophthalmology*, 59(2):97. Cited on pag. 92

Andersson *et al.* (2010) Richard Andersson, Marcus Nyström and Kenneth Holmqvist. Sampling frequency and eye-tracking measures: how speed affects durations, latencies, and more. *Journal of Eye Movement Research*, 3(3). ISSN 1995-8692. URL https://bop.unibe.ch/index.php/JEMR/article/view/2300. Cited on pag. 3

Bass *et al.* (1996) Michael Bass, E Van Stryland, D Williams and W Wolfe. *Handbook of Optics*, volume 2. McGraw-Hill. ISBN 0-07-047974-7. Cited on pag. 95

Behar-Cohen *et al.* (2011) Francine Behar-Cohen, C Martinsons, Françoise Viénot, Georges Zissis, A Barlier-Salsi, JP Cesarini, O Enouf, M Garcia, Serge Picaud and D Attia. Light-emitting diodes (led) for domestic lighting: Any risks for the eye? *Progress in retinal and eye research*, 30 (4):239–257. Cited on pag. 91

Blignaut and Beelders (2012) Pieter Blignaut and Tanya Beelders. The precision of eye-trackers: a case for a new measure. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 289–292. ACM. Cited on pag. 71

Borsato *et al.* (2015) Frank Borsato, Fernando Aluani and Carlos Morimoto. A Fast and Accurate Eye Tracker Using Stroboscopic Differential Lighting. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 110–118. Cited on pag. 44

Bourns C.P. Division (2006) Bourns C.P. Division. *MF-MSMF Series - PTC Resettable Fuses*, Junho 2006. URL https://www.bourns.com/pdfs/mfmsmf.pdf. Cited on pag. 103

Bradley *et al.* (2009) Derek Bradley, Bradley Atcheson, Ivo Ihrke and Wolfgang Heidrich. Synchronization and rolling shutter compensation for consumer video camera arrays. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 1–8. IEEE. Cited on pag. 9

Bulling and Gellersen (2010) Andreas Bulling and Hans Gellersen. Toward mobile eye-based human-computer interaction. *IEEE Pervasive Computing*, 9(4):8–12. Cited on pag. 3

Buswell (1935) Guy Thomas Buswell. How people look at pictures: a study of the psychology and perception in art. Cited on pag. 1

C3 Tech (2015) C3 Tech. WEBCAM FULL HD 1080P WB411. [Online, 14/jan/2015], 2015. URL http://www.c3technology.com.br/2013/produto.php?id=WB411FULLHD. Cited on pag.

Castro (2008) Fabio Lo Castro. Class i infrared eye blinking detector. *Sensors and Actuators A: Physical*, 148(2):388–394. ISSN 0924-4247. doi: http://dx.doi.org/10.1016/j.sna.2008.09.005. URL http://www.sciencedirect.com/science/article/pii/S0924424708004718. Cited on pag. 91

Charlier and Hache (1982) JR Charlier and JC Hache. New instrument for monitoring eye fixation and pupil size during the visual field examination. *Medical and Biological Engineering and Computing*, 20(1):23–28. Cited on pag. 2

Cree Inc. (2013) Cree Inc. *Eye Safety with LED Components*, nov 2013. URL http://www.cree.com/~/media/Files/Cree/LEDComponentsandModules/XLamp/XLampApplicationNotes/XLamp_EyeSafety.pdf. Cited on pag. 91

Duchowski (2002) Andrew T Duchowski. A breadth-first survey of eye-tracking applications. *Behavior Research Methods, Instruments, & Computers*, 34(4):455–470. Cited on pag. 1

Duchowski *et al.* (2016) Andrew T. Duchowski, Sophie Jörg, Tyler N. Allen, Ioannis Giannopoulos and Krzysztof Krejtz. Eye Movement Synthesis. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ETRA '16, pages 147–154, New York, NY, USA. ACM. ISBN 978-1-4503-4125-7. doi: 10.1145/2857491.2857528. URL http://doi.acm.org/10.1145/2857491.2857528. Cited on pag. 3

Dvornychenko (1983) VN Dvornychenko. Bounds on (deterministic) correlation functions with application to registration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):206–213. Cited on pag. 44

Ebisawa (1998) Y. Ebisawa. Improved video-based eye-gaze detection method. *IEEE Transactions on Instrumentation and Measurement*, 47(4):948–955. doi: 10.1109/19.744648. Cited on pag. 2, 8, 22, 44

Ebisawa (1995) Yoshinobu Ebisawa. Unconstrained pupil detection technique using two light sources and the image difference method. *Visualization and Intelligent Design in Engineering and Architecture*, pages 79–89. Cited on pag. 2, 81, 82

Ebisawa and Satoh (1993) Yoshinobu Ebisawa and S-I Satoh. Effectiveness of pupil area detection technique using two light sources and image difference method. In *Engineering in Medicine and Biology Society, 1993. Proceedings of the 15th Annual International Conference of the IEEE*, pages 1268–1269. IEEE. Cited on pag. 2

Enright (1998) JT Enright. Estimating peak velocity of rapid eye movements from video recordings. *Behavior Research Methods, Instruments, & Computers*, 30(2):349–353. Cited on pag. 3

Epistar Corporation (2014) Epistar Corporation. *EPISTAR ES-SASFPN42D AlGaAs PN-series LED Chip*, Setembro 2014. URL http://www.epistar.com.tw/upfiles/files_/all_product_caty01430891928.pdf. Cited on pag. 109

Epistar Corporation (2012) Epistar Corporation. *EPISTAR ES-SAUFPN42A AlGaAs PN-series LED Chip*, Maio 2012. URL http://www.epistar.com.tw/upfiles/files_/ES-SAUFPN42A.pdf. Cited on pag. 109

Fairchild Semiconductor (2000) Fairchild Semiconductor. *DM74ALS74A Dual D Positive-Edge-Triggered Flip-Flop*, Fevereiro 2000. URL https://www.fairchildsemi.com/datasheets/DM/DM74ALS74A.pdf. Cited on pag. 105

Fairchild Semiconductor (2013) Fairchild Semiconductor. *LM555 Single Timer*, Janeiro 2013. URL https://www.fairchildsemi.com/datasheets/LM/LM555.pdf. Cited on pag. 105

Ferhat *et al.* **(2014a)** O. Ferhat, F. Vilarino and F.J. Sanchez. A cheap portable eye-tracker solution for common steups. *Journal of Eye Movement Research*, 7(3):1–10. Cited on pag. 3

Ferhat *et al.* **(2014b)** Onur Ferhat, Fernando Vilarino and FJ Sánchez. A cheap portable eye-tracker solution for common setups. *Journal of Eye Movement Research*, 7(3):2. Cited on pag. 3

Fischler and Bolles (1981) Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395. Cited on pag. 47

Fitzgibbon *et al.* **(1999)** Andrew Fitzgibbon, Maurizio Pilu and Robert B Fisher. Direct least square fitting of ellipses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21 (5):476–480. Cited on pag. 48

Fuhl *et al.* **(2015a)** Wolfgang Fuhl, Thomas Kübler, Katrin Sippel, Wolfgang Rosenstiel and Enkelejda Kasneci. *Computer Analysis of Images and Patterns: 16th International Conference, CAIP 2015, Valletta, Malta, September 2-4, 2015 Proceedings, Part I*, chapter ExCuSe: Robust Pupil Detection in Real-World Scenarios, pages 39–51. Springer International Publishing, Cham. ISBN 978-3-319-23192-1. doi: 10.1007/978-3-319-23192-1_4. URL http://dx.doi.org/10.1007/978-3-319-23192-1_4. Cited on pag. xv, 2, 61, 62, 66, 70

Fuhl *et al.* **(2015b)** Wolfgang Fuhl, Thomas Kübler, Katrin Sippel, Wolfgang Rosenstiel and Enkelejda Kasneci. Excuse: Robust pupil detection in real-world scenarios. In *Computer analysis of images and patterns*, pages 39–51. Springer. Cited on pag. 49

Goni *et al.* **(2004)** Sonia Goni, Javier Echeto, Arantxa Villanueva and Rafael Cabeza. Robust algorithm for pupil-glint vector detection in a video-oculography eyetracking system. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 941–944. IEEE. Cited on pag. 2

Hansen and Ji (2010a) Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(3):478–500. Cited on pag. 1

Hansen and Pece (2005) Dan Witzner Hansen and Arthur EC Pece. Eye tracking in the wild. *Computer Vision and Image Understanding*, 98(1):155–181. Cited on pag. 1, 2

Hansen and Ji (2010b) D.W. Hansen and Q. Ji. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3). Cited on pag. 2

Hennessey *et al.* **(2006)** C. Hennessey, B. Noureddin and P. Lawrence. A Single Camera Eye-gaze Tracking System with Free Head Motion. In *Proceedings of the 2006 Symposium on Eye Tracking Research & Applications*, pages 87–94. Cited on pag. 8, 71

Hennessey *et al.* **(2008)** C. Hennessey, B. Noureddin and P. Lawrence. Fixation Precision in High-Speed Noncontact Eye-Gaze Tracking. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(2):289–298. Cited on pag. 3

Holland and Komogortsev (2012) Corey Holland and Oleg Komogortsev. Eye tracking on unmodified common tablets: challenges and solutions. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 277–280. ACM. Cited on pag. 9

Holmqvist *et al.* **(2011)** Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka and Joost Van de Weijer. *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford. Cited on pag. 1

Hooke and Jeeves (1961) Robert Hooke and To A Jeeves. "Direct Search"Solution of Numerical and Statistical Problems. *Journal of the ACM (JACM)*, 8(2):212–229. Cited on pag. 49

ICNIRP (2013) ICNIRP. Guidelines on limits of exposure to incoherent visible and infrared radiation. *Health Physics*, 105(1):74–96. Cited on pag. xiii, 92, 93, 94, 95

IEC (2007) IEC. Safety of laser products – part1: Equipment classification and requirement. Number IEC 60825-1:2007. Cited on pag. 91

IEC (2006) IEC. Photobiological safety of lamps and lamp systems. Number IEC 62471:2006. Cited on pag. xv, 80, 91, 92, 93, 94, 95, 108, 112

Ivins and Porrill (1998) P. James Ivins and John Porrill. A deformable model of the human iris for measuring small three-dimensional eye movements. *Machine Vision and Applications*, 11(1):42–51. ISSN 1432-1769. doi: 10.1007/s001380050089. URL http://dx.doi.org/10.1007/s001380050089. Cited on pag. 1

Jäger (2010) Claus Jäger. Eye safety of ireds used in lamp applications. Cited on pag. 93, 95

Jelinkov *et al.* (2004) Helena Jelinkov, Jiri Pasta, Jan Sulc, Michal Nemec, Petr Koranda and Petr Hrabal. Near-infrared nd: Yap laser radiation transmission through the human eye tissue structure. In *Biomedical Optics 2004*, pages 315–319. International Society for Optics and Photonics. Cited on pag. 93

Kalman (1960) Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45. Cited on pag. 30, 31

Kass *et al.* (1988) Michael Kass, Andrew Witkin and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331. Cited on pag. 72, 81

Kassner *et al.* (2014a) Moritz Kassner, William Patera and Andreas Bulling. Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction. *arXiv:1405.0006*. URL http://arxiv.org/abs/1405.0006. Cited on pag. 3

Kassner *et al.* (2014b) Moritz Kassner, William Patera and Andreas Bulling. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 1151–1160. ACM. Cited on pag. 2, 3, 47, 49

Keil *et al.* (2010) Andrea Keil, Georgia Albuquerque, Kai Berger and Marcus Andreas Magnor. Real-time gaze tracking with a consumer-grade video camera. Cited on pag. 2

Kim *et al.* (2014) Elizabeth S. Kim, Adam Naples, Giuliana Vaccarino Gearty, Quan Wang, Seth Wallace, Carla Wall, Michael Perlmutter, Jennifer Kowitt, Linda Friedlaender, Brian Reichow, Fred Volkmar and Frederick Shic. Development of an Untethered, Mobile, Low-cost Head-mounted Eye Tracker. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '14, pages 247–250, New York, NY, USA. ACM. ISBN 978-1-4503-2751-0. doi: 10.1145/2578153.2578209. URL http://doi.acm.org/10.1145/2578153.2578209. Cited on pag. 3, 9

Kim *et al.* (2006) Soo Chan Kim, Ki Chang Nam, Won Sang Lee and Deok Won Kim. A new method for accurate and fast measurement of 3d eye movements. *Medical Engineering & Physics*, 28(1):82–89. ISSN 1350-4533. doi: http://dx.doi.org/10.1016/j.medengphy.2005.04.002. URL http://www.sciencedirect.com/science/article/pii/S1350453305000779. Cited on pag. 1

Kinsman *et al.* (2012) Thomas Kinsman, Karen Evans, Glenn Sweeney, Tommy Keane and Jeff Pelz. Ego-motion Compensation Improves Fixation Detection in Wearable Eye Tracking. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 221–224, New York, NY, USA. ACM. ISBN 978-1-4503-1221-9. doi: 10.1145/2168556.2168599. URL http://doi.acm.org.ez67.periodicos.capes.gov.br/10.1145/2168556.2168599. Cited on pag. 3

Lemire (2007) Daniel Lemire. A Better Alternative to Piecewise Linear Time Series Segmentation. In *SIAM Data Mining 2007.* Cited on pag.

Li and Parkhurst (2005) Dongheng Li and Derrick J Parkhurst. Starburst: A robust algorithm for video-based eye tracking. *Elselvier Science*, page 6. Cited on pag. xv, 2, 14, 41, 42, 48, 49, 61, 62, 66, 70

Logitech (2015) Europe S.A. Logitech. HD Webcam C270. [Online, 14/jan/2015], 2015. URL http://www.logitech.com/en-us/product/hd-webcam-c270. Cited on pag.

Lukander *et al.* (2013) Kristian Lukander, Sharman Jagadeesan, Huageng Chi and Kiti Müller. OMG!: A New Robust, Wearable and Affordable Open Source Mobile Gaze Tracker. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 408–411, New York, NY, USA. ACM. ISBN 978-1-4503-2273-7. doi: 10.1145/2493190.2493214. URL http://doi.acm.org/10.1145/2493190.2493214. Cited on pag. 3

Maisonobe (2006) L. Maisonobe. Quick computation of the distance between a point and an ellipse, Feb 2006. URL https://www.spaceroots.org/documents/distance/distance-to-ellipse.pdf. Cited on pag. 49

Morimoto and Mimica (2005a) C. H. Morimoto and M.R.M. Mimica. Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98:4–24. Cited on pag. 61

Morimoto and Mimica (2005b) Carlos H. Morimoto and Marcio R.M. Mimica. Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98 (1):4–24. ISSN 1077-3142. Cited on pag. 1

Morimoto *et al.* (1999) Carlos H Morimoto, D Koons, A Amir and M Flickner. Frame-rate pupil detector and gaze tracker. In *Proceedings of the IEEE ICCV*, volume 99. Cited on pag. 2

Morimoto *et al.* (2000) C.H. Morimoto, D. Koons, A. Amir and M.D. Flickner. Pupil detection and tracking using multiple light sources. *Image and Vision Computing*, 18(4):331–335. Cited on pag. 4, 8, 22, 44

Mulvey *et al.* (2008) 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)*, (1). Cited on pag. 91, 93, 111

Nagamatsu *et al.* (2010) Takashi Nagamatsu, Michiya Yamamoto and Hiroshi Sato. MobiGaze: Development of a gaze interface for handheld mobile devices. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 3349–3354. ACM. Cited on pag. 3

Nguyen *et al.* (2002) Karlene Nguyen, Cindy Wagner, David Koons and Myron Flickner. Differences in the infrared bright pupil response of human eyes. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 133–138. ACM. Cited on pag. 87

Ohno *et al.* (2002) Takehiko Ohno, Naoki Mukawa and Atsushi Yoshikawa. FreeGaze: a gaze tracking system for everyday gaze interaction. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 125–132. ACM. Cited on pag. 2, 44

OmniVision Technologies, Inc. (2007) OmniVision Technologies, Inc. *OV7725 Color CMOS VGA OmniPixel2TM CAMERA CHIP Sensor*, Março 2007. Cited on pag. 6, 10

ON Semiconductor (2014) ON Semiconductor. *AR0130: 1/3-Inch CMOS Digital Image Sensor*, Novembro 2014. URL http://www.onsemi.com/pub/Collateral/AR0130CS-D.PDF. Cited on pag. 10

ON Semiconductor (2008) ON Semiconductor. *NTGD3148N Power MOSFET 20V, 3.5A, Dual N-Channel, TSOP-6*, 2008. URL http://www.onsemi.com/pub_link/Collateral/NTGD3148N-D.PDF. Cited on pag. 80, 105, 110, 114

OSRAM Opto Semiconductors (2014a) OSRAM Opto Semiconductors. *SFH 4715 OSLON Black Series (850 nm)*, Setembro 2014a. URL http://www.osram-os.com/Graphics/XPic5/00100752_0.pdf. Cited on pag. 80, 97, 102, 114

OSRAM Opto Semiconductors (2014b) OSRAM Opto Semiconductors. *SFH 4050 High Power Infrared Emitter (850 nm)*, Novembro 2014b. URL http://www.osram-os.com/Graphics/XPic1/00206346_0.pdf. Cited on pag. 107, 108

Philips Semiconductors (1988) Philips Semiconductors. *AN170 - NE555 and NE556 Application Notes*, Dezembro 1988. URL http://www.williamson-labs.com/pdf/555AN.pdf. Cited on pag. 104

Point Grey (2015) Point Grey. *BlackFly USB 3.0 Digital Camera Technical Reference*, Junho 2015. URL https://www.ptgrey.com/support/downloads/10264. Cited on pag. 5

Pupil-Labs.com (2016) Pupil-Labs.com. Pupil-labs binocular headset. [Online, 2013-Aug], 2016. URL https://pupil-labs.com/. Cited on pag. 2, 3

Rayner (1998) Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372. Cited on pag. 53

Robinson (1964) DA Robinson. The mechanics of human saccadic eye movement. *The Journal of physiology*, 174(2):245. Cited on pag. 55

Ryan *et al.* (2008a) Wayne J Ryan, Andrew T Duchowski and Stan T Birchfield. Limbus/pupil switching for wearable eye tracking under variable lighting conditions. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 61–64. ACM. Cited on pag. 72

Ryan *et al.* (2008b) Wayne J Ryan, Damon L Woodard, Andrew T Duchowski and Stan T Birchfield. Adapting starburst for elliptical iris segmentation. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, pages 1–7. IEEE. Cited on pag. 72, 81

Ryer and Light (1997) Alex Ryer and Visible Light. Light measurement handbook. [Online, 18/dez/2013], 1997. URL http://www.intl-light.com/handbook/. Cited on pag. 94

San Agustin *et al.* (2010) Javier San Agustin, Henrik Skovsgaard, Emilie Mollenbach, Maria Barret, Martin Tall, Dan Witzner Hansen and John Paulin Hansen. Evaluation of a Low-cost Open-source Gaze Tracker. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, pages 77–80, New York, NY, USA. ACM. ISBN 978-1-60558-994-7. Cited on pag. 49

SensoMotoric Instruments (2016) SensoMotoric Instruments. Smi eye tracking glasses 2 wireless. [Online, 2013-Aug], 2016. URL http://www.eyetracking-glasses.com/. Cited on pag. 2, 3

Sony (2014) Computer Entertainment America LLC Sony. Playstation eye camera. [Online, 14/jan/2014], 2014. URL http://us.playstation.com/ps3/accessories/playstation-eye-camera-ps3.html. Cited on pag. 5, 109, 114

Świrski *et al.* (2012) Lech Świrski, Andreas Bulling and Neil Dodgson. Robust real-time pupil tracking in highly off-axis images. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 173–176. ACM. Cited on pag. xii, xv, 2, 48, 49, 52, 61, 62, 66, 79

Texas Instruments Incorporated (2015) Texas Instruments Incorporated. *SN65LVDS324 1080p60 Image Sensor Receiver*, Março 2015. URL http://www.ti.com/lit/ds/sllsed9a/sllsed9a.pdf. Cited on pag. 10

Theobalt *et al.* **(2004)** Christian Theobalt, Irene Albrecht, Jörg Haber, Marcus Magnor and Hans-Peter Seidel. Pitching a baseball: tracking high-speed motion with multi-exposure images. *ACM Transactions on Graphics (TOG)*, 23(3):540–547. Cited on pag. 9

Tobii AB (2016) Tobii AB. Tobii pro glasses 2. [Online, 2013-Aug], 2016. URL http://www.tobiipro.com/. Cited on pag. 2, 3

Topal *et al.* **(2014)** C. Topal, S. Gunal, O. Kocdeviren, A. Dogan and O. Nezih Gerek. A low-computational approach on gaze estimation with eye touch system. *Cybernetics, IEEE Transactions on*, 44(2):228–239. ISSN 2168-2267. doi: 10.1109/TCYB.2013.2252792. Cited on pag. 91

USB Implementers Forum (2005) USB Implementers Forum. Universal Serial Bus Device Class Definition for Video Devices, Jun 2005. Cited on pag. 29

Vishay Intertechnology Inc. (2013) Vishay Intertechnology Inc. *Drift Calculation for Thin Film Resistors - Technical Note*, Abril 2013. URL http://www.vishay.com/docs/28809/driftcalculation.pdf. Cited on pag. 38

Vishay Roederstein (2012) Vishay Roederstein. *Film Capacitors - General Technical Information*, Junho 2012. URL http://www.vishay.com/docs/26033/gentechinfofilm.pdf. Cited on pag. 38

W. Fuhl (2016) T. Kübler E. Kasneci W. Fuhl, T. Santini. Else: Ellipse selection for robust pupil detection in real-world environments. In *ACM Symposium on Eye Tracking Research & Applications, ETRA 2016*. Cited on pag. xv, 2, 61, 62, 66, 70

Wade (2010) Nicholas J Wade. Pioneers of eye movement research. *i-Perception*, 1(2):33–68. Cited on pag. 1

XLSEMI (2016) XLSEMI. *XL4015 - 5A 180KHz 36V Buck DC to DC Converter*, Maio 2016. URL http://www.xlsemi.com/datasheet/XL4015datasheet.pdf. Cited on pag. 114

Yarbus (1967) Alfred. L. Yarbus. *Eye Movements and Vision.* Plenum Press. Cited on pag. 1

Zhu and Yang (2002) Jie Zhu and Jie Yang. Subpixel eye gaze tracking. In *Automatic face and gesture recognition, 2002. proceedings. fifth ieee international conference on*, pages 124–129. IEEE. Cited on pag. 2

Zhu *et al.* **(2002)** Zhiwei Zhu, Kikuo Fujimura and Qiang Ji. Real-time eye detection and tracking under various light conditions. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 139–144. ACM. Cited on pag. 2