Towards Ubiquitous Wearable Eye Tracking

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von

M. Sc. Thiago Caberlon Santini

aus Caxias do Sul, Brasilien

Tübingen
2019

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	12.08.2019
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Enkelejda Kasneci
2. Berichterstatter:	Prof. Dr. Wolfgang Rosenstiel
3. Berichterstatter:	Prof. Dr. Prof. Dr. Andrew T. Duchowski

Acknowledgments

First and foremost, I would like to thank my parents Gilberto and Maria Inês as well as my brother Diego, for their love and continuous support throughout my life. I'm also grateful to Isabel, H. T., and P. T. – a bit of silliness makes life all the more enjoyable :-)

To Prof. Dr. Enkelejda Kasneci and Prof. Dr. Wolfgang Rosenstiel my heartfelt thanks for the incredible opportunity and guidance (academic, scientific, and otherwise) throughout my doctoral studies. You have been a continous source of motivation and inspiration to strive for excellence.

To my dear office colleagues throughout the years – David Geisler, Esteban Gutierrez, Nora Castner, Dr. Thomas Kübler, and Wolfgang Fuhl – a special thanks for all the (many!) laughs, cooperations, discussions, and ideas. I am also grateful to the rest of Perception Engineering and Neuro teams – Benedikt Hosp, Dr. Shahram Eivazi, Tobias Appel, and others. I have throughly enjoyed our time together. Moreover, I would like to thank the staff of the Computer Engineering and Embedded Systems groups, specially Margot Reimold and Dr. Oliver Bringmann, the EAES doctoral college, as well as the BP team for their support.

I would also like to extend my gratitude to all my scientific collaborators, specially: a) Prof. Dr. Marcus Nyström, Dr. Diederick Niehoster, and the Humanities Labs at Lund University, for enabling and welcoming me during my research stay in Lund, as well as b) Prof. Dr. Raphael Rosenberg, Prof. Dr. Helmut Leder, Dr. Hanna Bringmann, Dr. Luise Reistätter, and the CReA and EVA Labs at the University of Vienna, for the unique opportunity and cooperation during the Belvedere studies.

"*Research your own experience: Absorb what is useful, reject what is useless, add what is essentially your own.*"

—Bruce Lee

Abstract

It is often said that the *eyes are the window to the soul*. While science might never be able to prove or disprove the mere existence of a soul, it has nonetheless found strong evidence that *gaze is the window to the mind*. Not only has eye tracking played a key part in these findings, it also holds immeasurable potential to further advance our understanding of the human mind and revolutionize the way we interact with our devices. However, to fully realize this potential, it is imperative to bring eye tracking out of the laboratory and into the wild. Towards enabling ubiquitous wearable eye tracking, this thesis deals with the challenges involved in this transference, focusing on head-mounted video-based devices.

At the feature extraction level, this work introduces novel methods to reliably and robustly detect and track the pupil in real-time, proposing advanced metrics for their evaluation, as well as exploring the influence of such methods on the resulting estimated gaze signal. At the gaze estimation level, the thesis deals with two of the main factors hindering a wider eye tracking adoption: Calibration and Slippage. To improve calibration usability, a innovative approach is proposed, which enables the collection of plentiful calibration points quickly and unsupervisedly virtually anywhere. Slippage is tackled by a novel hybrid method combining a geometrical slippage-robust feature and traditional regression-based gaze-mapping functions. At the eye movement detection level, a probabilistic approach is proposed to identify fixations, saccades, and smooth pursuits at lower sampling rates.

The methods herein introduced form the basis of a complete state-of-the-art eye-tracking platform competitive with expensive commercial eye-tracking systems. Unlike these commercial systems, the platform is open-source, able to track participants outdoors and with glasses, and requires no additional hardware such as stereo eye cameras or multi-glint patterns. The effectiveness of the resulting system is shown through a large-scale, pervasive, and unconstrained eye-tracking study.

Zusammenfassung

Es wird oft gesagt, dass die Augen das Fenster zur Seele sind. Während die Wissenschaft vielleicht nie in der Lage sein wird, die Existenz der Seele zu beweisen oder zu widerlegen, hat die Forschung dennoch etliche Beweise dafür geliefert, dass unsere Augen das Fenster zum Gehirn sind. Die Aufzeichnung der Augenbewegungen, sog. Eye-Tracking, spielt nicht nur eine Schlüsselrolle bei der Erforschung vieler kognitiver Prozesse, sondern birgt darüber hinaus enormes Potenzial zur Entwicklung intelligenter Interaktionsschnittstellen. Um dieses Potenzial jedoch ausschöpfen zu können, ist es unabdingbar, die Eye-Tracking-Technologie aus dem Labor in alltäglichen Anwendungen zu bringen. Diese Dissertation stellt sich den Herausforderungen auf dem Weg zum allgegenwärtigen, tragbaren Eye-Tracking und erforscht Methoden für eine robuste Blickerfassungstechnologie mit Fokus auf kopfgetragenen videobasierten Geräten.

Auf der Ebene der Merkmalsextraktion stellt diese Arbeit neue Verfahren zur zuverlässigen und robusten Erkennung und Verfolgung der Pupille in Augenbildern vor und führt zudem fortgeschrittene Metriken für ihre Bewertung und zur Untersuchung des Einflusses solcher Methoden auf das resultierende, geschätzte Blicksignal ein. In Bezug auf die Blickschätzung beschäftigt sich diese Dissertation mit zwei der wichtigsten Faktoren, die aktuell eine breitere Akzeptanz der Eye-Tracking-Technologie erschweren: Kalibrierung und Verschiebung des kopfgetragenen Geräts. Um die Handhabung der Kalibrierung zu verbessern, wurde ein innovativer Ansatz erforscht, der es ermöglicht, zahlreiche Kalibrierpunkte schnell und unbeaufsichtigt praktisch überall zu sammeln. Zur Vermeidung von Blickschätzungsungenauigkeiten, die durch das Verrutschen des Eye-Tracking-Geräts entstehen können, wurde eine neue Hybridmethode erforscht, die einen geometrischen Ansatz mit traditionellen regressionsbasierten Blickabbildungsfunktionen kombiniert. Auf der Ebene der Augenbewegungserkennung wird ein probabilistischer Ansatz vorgeschlagen, um Fixationen, Sakkaden und Folgebewegungen bei geringen Abtastraten robust zu erkennen.

Die hierin erforschten Methoden bilden die Grundlage für eine komplette, state-of-the-art Eye-Tracking-Plattform, die mit teuren kommerziellen Eye-Tracking-Systemen konkurrenzfähig ist. Im Gegensatz zu kommerziellen Systemen steht die Plattform jedoch als Open-Source-Lösung zur Verfügung, sie ermöglicht eine robuste Blickerfassung in freien Settings und Außenstudien, und erfordert keine zusätzliche Hardware wie Stereo-Augenkameras oder Multi-Glint-Muster. Die Effektivität des resultierenden Systems wird durch eine groß angelegte, umfassende und uneingeschränkte Eye-Tracking-Studie demonstriert.

Contents

List of Figures

List of Tables

List of Abbreviations

ADAS Advanced Driver-Assistance Systems

AFKF Attention Focus Kalman Filter

AOI Area of Interest

ASL Applied Science Laboratories

BPF Binocular Polynomial Fit

CalibMe Calibrating with Movements

COGAIN Communication by Gaze Interaction

COTS Commercial-of-the-Shelf

CNN Convolutional Neural Network

DIY Do-It-Yourself

ElSe Ellipse Selector [1]

ElSe+ Confidence Augmented Ellipse Selector

EOG Electrooculography

ESCaF Ellipse Selection by Candidate Filtering [2]

EUR Euros

ExCuSe Exclusive Curve Selector [3]

ExCuSe+ Confidence Augmented Exclusive Curve Selector

FOV Field of View

GPU Graphic Processing Unit

Grip Gaze regression: instantaneous and pervasive

GUI Graphical User Interface

HCI Human-Computer Interaction

I-BMM Bayesian Mixture Model Identification

I-BDT Bayesian Decision Theory Identification

I-PCA Principal Component Analysis Identification

I-VMP Velocity and Movement Pattern Identification

I-VMPRay Velocity Movement Pattern Rayleigh Identification

I-VMPStd Velocity Movement Pattern Standard Deviation Identification

I-VDT Velocitiy and Dispersion Threshold Identification

I-VVT Velocitiy and Velocity Threshold Identification

LED Light Emitting Diode

LOG Line of Gaze

LOS Line of Sight

LPW Labelled Pupils in the Wild [4]

9-Points 9-Points calibration

NIHS Not Invented Here Syndrome

N-Points N-Points calibration

POG Photooculography

POR Point of Regard

Qt The Qt framework [5]

PuRe Pupil Reconstructor [6]

PuReST Pupil Reconstructor and Subsequent Tracking [7]

RANSAC Random Sample Consensus [8]

ROI Region of Interest

SMI SensoMotoric Instruments GmbH

SOMA Smart Ocular Motility Analysis [9]

UDP User Datagram Protocol

USD United States dollar

UVC USB Video Class [10]

VOG Videooculography

1 Introduction

"*What the eyes see and the ears hear, the mind believes.*"
—Harry Houdini

It is often said that the *eyes are the window to the soul*. While science might never be able to prove or disprove the mere existence of a soul, it has nonetheless found strong evidence that *gaze is the window to the mind*. For instance, it has been shown that through an individual's gaze (i.e., overt attention), it is possible to infer information regarding the individual's cognitive and affective states, activities, as well as intentions [11]–[20]. Not only has eye tracking played a key part in these findings, it also holds immeasurable potential to further advance our understanding of the human mind and revolutionize the way we interact with our devices. However, to fully realize this potential, it is imperative to bring eye tracking out of the laboratory and into the wild.

The individual head-mounted eye-tracking paradigm [21]–[23] is an ideal *wearable* candidate to enable ubiquitous in-the-wild eye tracking given its unobtrusiveness, flexibility, and mobility [24]. State-of-the-art head-mounted eye trackers are video-based glasses-like frames with at least one camera capturing images of the user's eye and one capturing part of the user's Field of View (FOV). Through a *calibration* step, the relationship between pupil[1] and gaze can be learned, after which the eye tracker is able to infer future gaze estimates w.r.t. the field camera by tracking the pupil. Given the current pace of head-mounted eye-tracking system modularization and miniaturization (e.g., see Fig. 1.1), these devices availability as inconspicuous wearable devices or integrated into head-worn hardware – e.g., smart, augmented-reality, and prescription glasses – seems evident.

(a) 2014 **(b)** 2014 **(c)** 2017 **(d)** 2018

Figure 1.1: State-of-the-art head-mounted eye-tracking hardware evolution. SMI ETG2 (1.1a) and Tobii Glasses Pro 2 (1.1b): cannot be used with glasses. Pupil (1.1c): fitting over glasses is cumbersome and sometimes even impossible. Inconspicuous and modular eye tracker proposed by Eivazi et al. [25] (1.1d): these modules can easily be attached to any glasses or 3D-printed frames [26].

[1] Other eye features such as corneal reflections can also be used instead or in combination with the pupil.

As illustrated in Fig. 1.2, we envision a future in which individuals wear miniaturized versions of these eye-tracking devices on a daily basis not only for digitally intermediated interaction with all kinds of devices [27]–[34], but also for other added benefits such as preventive health monitoring [35], [36], self quantization [37], [38], daily and life logging [39], [40], advanced driving assistance [41], work aid [42], training [43], [44], and alternative forms of veillance [45].

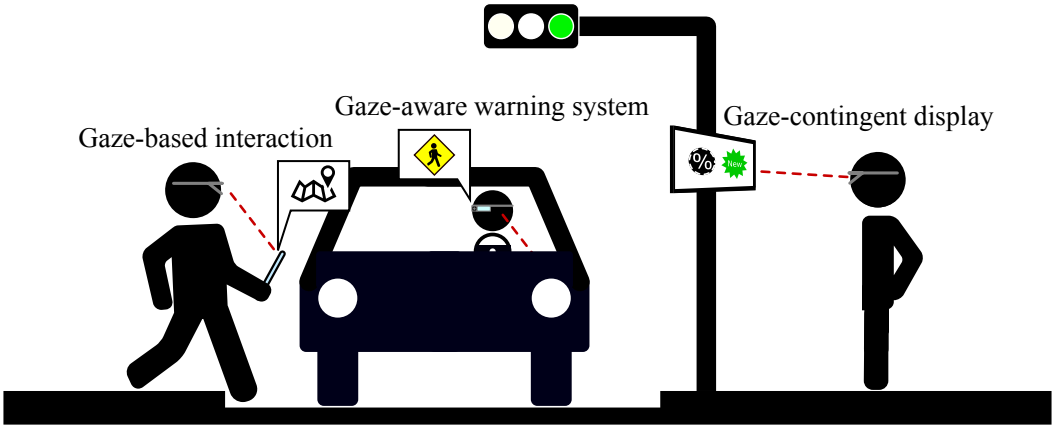


Figure 1.2: Envisioned ubiquitous wearable eye tracking examples. On the left, a distracted user interacts with his map application through his gaze while crossing the street. Meanwhile, a driver checks his dashboard and does not notice the distracted pedestrian; through eye tracking, the ADAS understands the pedestrian has not been perceived and emits a warning before any take over action is necessary. On the right, another pedestrian checks a gaze-contingent display [46] while waiting to cross the street. This user's eye tracker transmit the user-selected public profile, allowing the display to customize advertisements to the users' preferences.

The overarching objective of this work is to technologically enable this vision by providing robust real-time-capable methods able to continuously track subjects gaze even in challenging scenarios. We strive for high-usability, openness, hardware agnosticity, and wide coverage of participants without requiring parameter adjustments. The methods herein introduced form the basis of a complete state-of-the-art eye-tracking platform competitive with expensive commercial eye-tracking systems. Unlike these commercial systems, the platform is open-source, able to track participants outdoors and with glasses, and requires no additional hardware such as stereo eye cameras or multi-glint patterns.

1.1 Challenges

Many decades of research have lead to accurate and precise spatio-temporal eye-tracking methods in well-controlled setups [47]–[49]. Whereas these methods have significantly contributed to Human-Computer Interaction (HCI) research, as well as our understanding of the human oculomotor and visual system in laboratorial scenarios, they are not applicable in an ubiquitous sense because of the remarkably distinct requirements for pervasive eye

tracking:

Robust eye feature detection – particularly of the pupil and glints – form the basis for modern video-based eye tracking. In laboratorial conditions, the setup can be carefully adjusted to produce high-contrast uniformly-illuminated eye images for most users, from which the pupil and glints can be easily identified through simple methods (e.g., thresholding). With the shift from laboratorial to ubiquitous eye tracking, a whole plethora of noise sources surface, such as reflections, occlusions, blur, off-axial images, as well as uneven and erratic illumination [50]. As a result, not only image quality is diminished, but eye features can be partially or wholly occluded. Thus, extracting eye features from images captured in natural environments pose a considerably more challenging task. Moreover, if we expect users to adopt eye tracking on a daily basis, tracking these features should *just work* pervasively, without requiring the user to manually adjust parameters.

Calibration has been described as one of the main factors hindering a wider adoption of eye tracking technologies [51]. In laboratorial conditions, the user typically fixates through a series of reference points with the assistance of a supervising researcher: A slow, tedious, and error-prone procedure. In fact, even in this supervised scenario, eye-tracking user studies generally report calibration problems [52]. Naturally, supervision is unfeasible when ubiquitous eye tracking is considered. Moreover, the nature of this process presents very poor usability. Therefore a fast and unsupervised alternative that can be performed virtually anywhere is required in order to provide a user-friendly system usable on a daily basis.

Slippage is characterized by changes in eye tracker pose (translations or rotations) w.r.t. the calibration pose, thus corrupting the learned mapping function from the eye-camera feature space to gaze in field camera coordinates: A phenomenon also known as *calibration drift* [21], [22], [49]. Very few eye-tracking systems are robust to device slippage, and, in the laboratory, this issue is traditionally addressed by having the user gaze at a reference point between experiment trials. If a calibration drift is visible in the form of an offset between reference and gaze point, the system is recalibrated. In pervasive scenarios, known reference gaze points are rarely available, and periodically recalibrating the eye tracker also results in very poor usability. Consequently, a form of robustness to slippage is key in delivering a smooth and pleasing ubiquitous eye-tracking experience.

Additionally to the aforementioned challenges, ubiquitous wearable eye tracking also imposes extra constraints in the form of mobile, low-latency, and real-time operation – e.g., for HCI. Thus, solutions to these challenges must be attainable using modern embedded systems, in which no high-end desktop Graphic Processing Units (GPUs) can be assumed to exist. Furthermore, these solutions' computational requirements are also constrained as they must be applicable in conjunction with each other (e.g., in a binocular system, images from both cameras must be processed in parallel) and with other required tasks (such as capturing and storing video streams).

1.2 Structure and Contributions

This thesis is organized as follows: Chapter 2 introduces the reader to state-of-the-art head-mounted eye tracking technology and methodology. Novel methods for real-time robust pupil detection and tracking are presented in Chapter 3; additionally, this chapter also investigates the effect of pupil detection and tracking algorithms choice on gaze signal properties. Chapter 4 explores the eye-tracker calibration and gaze estimation design space. On a first moment, a innovative calibration approach is proposed, which enables the collection of plentiful calibration points quickly and unsupervisedly virtually anywhere, thus significantly improving calibration usability. On a second moment, we tackle eye tracker sllipage through a novel hybrid method combining a geometrical slippage-robust feature and traditional regression-based gaze-mapping functions. Chapter 5 introduces a probabilistic approach to identify fixations, saccades, and smooth pursuits at the lower sampling rates found in some head-mounted eye trackers. Chapter 6 ties together existing and proposed methods with other functionalities required to construct a complete state-of-the-art open-source eye-tracking platform, introducing the resulting platform's architecture and interfaces. The methods' and platform's capability and effectiveness are shown through a large-scale, pervasive, and unconstrained eye-tracking study described in Chapter 7.

Research pertaining to this thesis as well as related work developed during the course of its progress were presented in a series of renowed peer-reviewed conferences. These research works are published in these conferences' proceedings as well as renowed peer-reviewed scientific journals. Contributions to eye tracking hardware advancement were published in the

- *Proceedings of the 2018 ACM Symposium on Eye Tracking Research and Applications* [25]

and to applied eye tracking were published in the

- *Proceedings of the 2018 Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction* [53],
- *Proceedings of the 2017 Eye Tracking Enhanced Learning Workshop* [54], and
- *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction* [55],

as well as (*non-peer-reviewed*) at `https://arXiv.org` [56]. Methods to automatically extract features from eye images were published at

- *Proceedings of the 2019 ACM Symposium on Eye Tracking Research and Applications* [57].
- *Proceedings of the 2018 ACM Symposium on Eye Tracking Research and Applications* [7], [58],
- *Elsevier Computer Vision and Image Understanding Journal* [6],
- *Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision* [59],
- *Elsevier Computers in Biology and Medicine* [60],

- *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* [61]–[63], and
- *Proceedings of the 2016 ACM Symposium on Eye Tracking Research and Applications* [1],

as well as (non-peer-reviewed) at `https://arXiv.org` [64], [65]. The slippage-robust gaze estimation, high-usability calibration, and eye movement identification methods contributions were respectively published in the

- *Proceedings of the 2019 ACM Symposium on Eye Tracking Research and Applications* [26].
- *Proceedings of the 2017 ACM Conference on Human Factors in Computing Systems* [24], and
- *Proceedings of the 2016 ACM Symposium on Eye Tracking Research and Applications* [66].

Additionally, multiple eye-tracking-related softwares were published in the

- *Proceedings of the 2017 International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* [67]–[69], and

- *Proceedings of the 2016 Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* [70].

2 Background

> "*We are stuck with technology when what we really want is just stuff that works.*"
>
> —Douglas Adams

To date, the term *eye tracking* has been employed as a catch-all term for two distinct but closely related *technologies*: 1) *oculography*, the measurement of horizontal, vertical or torsional movements of the eye, and 2) *gaze tracking*, the estimation of a user's *visual axis* w.r.t. a known frame of reference. Nevertheless, tracking the eye first started *without techonology at all* but through mere observational methods and qualitative descriptions. As far back as the ancient times, philosophers and mathematicians like Aristotle and Ptolemy already inquired about characteristics of binocular vision [71], [72] despite the lack of eye-tracking technology. This remained the case for many centuries; for instance, the retinal afterimage technique was in use to investigate nystagmus [73] around the late 18th century, as well as ocular torsion [74] and saccades [75] around the mid 19th century. Other rudimentary techniques involved using a finger or a rubber tube to count movements of the *corneal bulge* [75], [76]. The first instances of mechanical systems to actually track eye movements appear to be that of Ahrens [77], which consisted of a bristle attached to an ivory cup placed on the cornea[1], which was later adopted by Delabarre [79] and Huey [80] to investigate eye movements during reading. These methods were, however, exceedingly invasive and disrupted eye motion [81].

Since then, eye-tracking technology has proliferated. Throughout the years, various physical characteristics of the eye have been employed to track its movements, such as retina, corneo-retinal potential, intercanthi impedance, corneal bulge, corneal reflections, Purkinje images, limbus, pupil, scleral blood vessels, and patterns on the iris [82]. By exploiting these physical characteristics, numerous technologies are able to track eye movements to different extends, each with their own advantages and disadvantages. For instance, these technologies include the corneal reflection method [83], Electrooculography (EOG) [84], Photooculography (POG) [85], scleral search coil [86], and Videooculography (VOG) [87]. The corneal reflection and scleral search coil are seldom still employed; EOG is still the technology of choice for studying eye movements during sleep [21], and there is a revived interest in POG in the context of virtual-reality headsets [88], in which light conditions can be controlled. Nevertheless, in practice, video-based methods have mostly superseded other approaches for a multitude of reasons such as flexibility, practicality, and unobtrusiveness,

[1] A similar idea of attaching a bristle through the middle of the cornea is present in Rählmann's work [78] but seems to be only a thought experiment.

while achieving comparable accuracy and precision for most practical use cases. Moreover, video-based methods have been continuously enjoying improvements both in terms of hardware – e.g., smaller, cheaper and higher-frame-rate cameras – and of software / algorithms – e.g., advancements in the field of computer vision are usually applicable to video-based eye-tracking.

2.1 Video-Based Head-Mounted Eye Tracking

In this work, we are mainly interested in tracking users' gaze pervasively, for which video-based head-mounted eye trackers are ideal candidates. These eye trackers typically consist of a wearable device with at least a) one camera capturing one of the user's eyes (henceforth the *eye camera*) and b) one camera capturing part of the user's FOV (henceforth the *field camera*). For this particular device class, the eye-tracking objective consists of estimating a gaze point in the field camera image plane, which corresponds to the *projection* onto the camera image plane of the Point of Regard (POR) – i.e., the intercept between visual axis and gazed object. This *gaze estimation* objective is achieved by tracking eye features automatically detected in the eye camera (e.g., the pupil center), and mapping these features to a point in the field camera image using a mapping function established a priori, as illustrated in Fig. 2.1. This mapping function can be derived from the eye tracker geometry or learned through a *calibration* step.

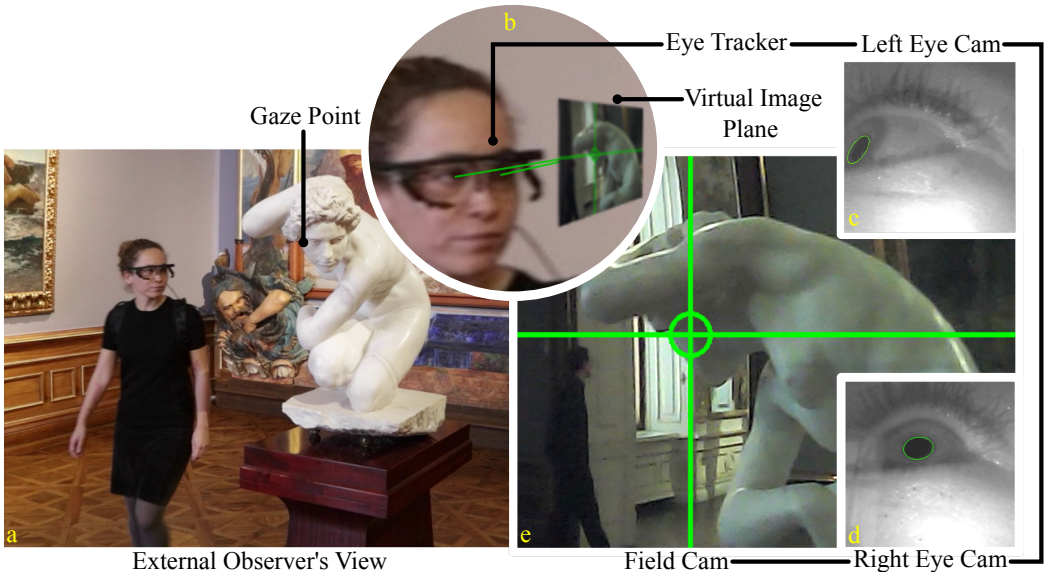


Figure 2.1: Video-based head-mounted eye tracking in a glance. *(a)* shows the perspective from an external viewer. (b) details how the *gaze point* is projected onto the field camera's *virtual image plane*. The *eye tracker* employs automatically detected eye features (in this case the pupils) from the left *(c)* and right *(d)* eye cameras' images to estimate the gaze point in the field camera's image *(e)*, giving us insights into how users perceive the world from their own perspective.

The first such system is likely the head-mounted Eye-Marker from Mackworth and Thomas [89] – an extension of the original Eye-Marker [90]. In this system, the eye marker (a corneal reflection generated by a system light source) is captured by the eye camera, which is then superimposed on the images captured by the *scene* (field) camera; during calibration, the system is adjusted so that, when the user gazes at known points, the eye marker lies over them. Whereas the Eye-Marker employed the corneal reflection technique, numerous other eye-tracking techniques are applicable to video-based head-mounted eye trackers by tracking distinct eye features – or even inferring gaze from appearance alone.

2.1.1 Eye Features

As previously mentioned, the eye has many features that can theoretically be used to track its movements. A few of these features can only be properly tracked in high-resolution and well-illuminated images – e.g., blood vessels and iris patterns. Others, like the limbus, are often too occluded by eyelashes and eyelids to be of general applicability, being particularly unreliable for users with epicanthic folds or droopy eyelids. The pupil and limbus are the most general eye features as these can be tracked under natural illumination without requiring additional light sources (i.e., under *passive illumination*). Under this kind of illumination, the limbus is usually easier to track due to higher contrast between iris and sclera than that of the iris and pupil, specially for dark eyes. Nonetheless, the pupil has its own advantages: It is less often occluded by eyelids due to its smaller size, and the pupil-iris boundary is usually sharper than the iris-sclera boundary [82]. Whereas the limbus can be tracked alone or in combination with the pupil [91], *predominantly only the pupil is tracked*. In practice, eye trackers often track the pupil in the near-infrared spectrum using *active illumination*. This is achieved by irradiating the eye with at least one near-infrared LEDs around the 700 nm to 900 nm band, which yields several benefits without perturbing natural vision – this band is invisible to the human eye and does not evoke any pupillary response. For instance: 1) the pupil-iris contrast is significantly improved, 2) tracking can be done in the absence of natural illumination, and 3) a large part of modern cameras are sensitive in this spectrum, thus allowing eye tracking with Commercial-of-the-Shelf (COTS) components. These infrared illuminators can also be used to achieve a *bright pupil effect* by placing a collimated illuminator close to the camera optical axis such that the light reflects on the retina and back to the camera – e.g., see Fig. 2.2.

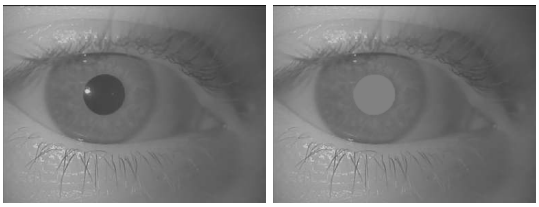


Figure 2.2: Pupil appearance under near-infrared illumination. When the illuminator is off-axis w.r.t. the camera, the pupil appears dark (left side); in contrast, if the illuminator is on-axis, the pupil appears bright (right side).

It is also possible to combine bright and dark pupil effects. For instance, if on- and off-axis illuminators and the camera are synchronized, one can time-multiplex bright and dark pupil images and track consecutive image differences [92], [93]. Alternatively, some modern systems can switch between bright and dark pupil modes automatically based on which one provides better tracking accuracy for the current user [94]. In the particular case of pervasive eye tracking, the bright pupil effect is discouraged as it is not reliable in the presence of extraneous infrared light sources such as the sun.

Aside from improving the tracking of natural eye characteristics, infrared illuminators radiating the eye also create artificial landmarks that can be used for eye tracking. These landmarks are generated by light reflecting on the optics of the eye, in particular the cornea and the lens, and are called Purkinje images [95]. Although both the *first and fourth Purkinje* images can be discerned, in most cases only the first image is reliable with the fourth image being weak and requiring well controlled illumination to be tracked [23]. The first Purkinje image is more commonly referred to as the *corneal reflection* or *glint*. These reflections are exemplified in Fig. 2.3 together with other eye features. In practice the great majority of video-based eye trackers tend to rely on the pupil (outline or center), corneal reflection, or a combination thereof for gaze estimation.

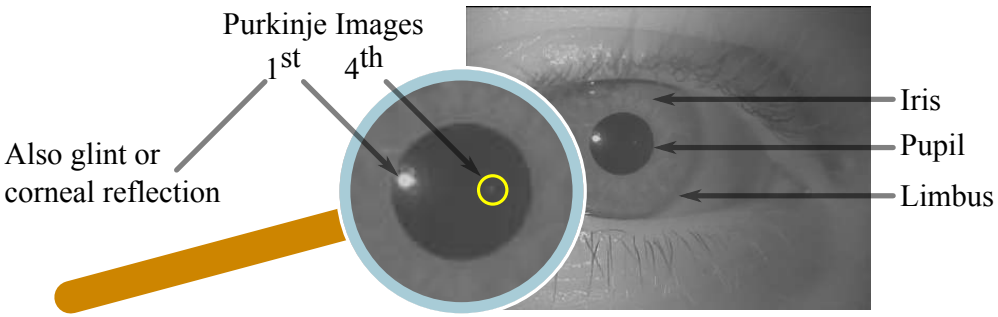


Figure 2.3: Eye features under illumination of a single near-infrared LED, resulting in two artificial landmarks (the first and fourth Purkinje images).

2.1.2 Gaze Estimation

Gaze estimation methods can be roughly divided into two classes – *regression-based* and *model-based* (or *geometrical*) – depending on whether they derive gaze estimates from input features based on a regressed function or geometrical constraints, both of which require some kind of calibration to determine required parameters.

The main eye-tracking-related calibrations are: *1) Camera* calibration, which is used to determine camera intrinsic parameters, *2) Extrinsic* calibration, which is used to determine the pose – i.e., extrinsic parameters – of system components (e.g., cameras, illuminators) relative to one another, *3) Individual* calibration, which is used to determine physiological characteristics such as cornea curvature and visual-to-optical axes offset, and *4) Gaze-Mapping* calibration, which is used to determine the coefficients of the function mapping

the input parameters to a gaze position. Despite the existence of multiple calibrations, geometrical approaches typically require only the first three, whereas regression-based approaches require just the fourth one. Counter intuitively, geometrical methods tend to have better usability despite requiring three distinct calibrations. This is true for *rigid-body* eye trackers – i.e., eye trackers in which components are fixed w.r.t. to each other – because the *camera* and *extrinsic* calibration are required only once per device. Thus, the only user-required calibration is the *individual* one, which is needed only once per user and can be achieved with a single point calibration. In contrast, the *gaze-mapping* calibration typically required by regression-based approaches involves the user looking through a series of distinct locations w.r.t. the eye tracker.

Regression-based gaze estimation methods are a data-driven approach to gaze estimation. In other words, these methods construct a mathematical model based on a data set containing relationships between the input feature space and the desired POR. Typically, such data sets are collected per user-session during a *gaze-mapping* calibration, with the most common input feature spaces being the *pupil center* and *pupil-corneal-reflection vector*. Many distinct machine-learning methods have been applied successfully to this task, including polynomial regressions [96]–[100], Gaussian processes [101], [102], and artificial neural networks [103], [104]. Moreover, there is a particular subclass of regression-based methods that does not rely on tracked eye features as input, instead employing feature-vector representations or the eye image as input to estimate the POR: *appearance-based* methods [105]–[107]. These feature vectors can either be manually crafted or learned from existing data, such as public annotated data sets or artificially created ones (e.g., by rendering eye images [108], [109]), thus possibly allowing for *calibration-free* gaze estimation.

Model-based gaze estimation methods take advantage of knowledge regarding the eye tracker geometry. These can be as simple as disregarding eye curvature and only approximating the eye and field cameras using the pin-hole camera model. In this manner, the pupil center in the eye camera image plane can be mapped to a gaze point in the field camera image plane using a planar *homography* [110], which can be estimated during a *gaze-mapping* calibration with as few as four data points. If the cameras have non-negligible distortions, the homography method also requires *camera calibration*, and the images to be undistorted. More complex approaches also model the eyes to estimate its parameters, such as center and optical axis direction, but require multiple observations from different eye poses or additional hardware. Temporal approaches such as the ones proposed by Świrski and Dogson [111] and Tsukada and Kanade [112] time-integrate distinct observations of the pupil and iris, respectively, to derive eye parameters such as center and optical axis. By using a calibrated stereo-setup, Kohlbecher et al. [113] presented a method to reconstruct the pupil in 3D space and, thus, the optical axis. Based on the pupil center and glint center, Shih et al. [114] showed that a single-camera single-glint does not suffice to achieve eye-camera pose invariance, thus requiring additional components. Guestrian and Eizenman formalized the geometry required to determine the eye visual axis for a series of distinct *fuilly-calibrated* cameras-glints configurations, concluding that at least one camera and two glints are required to achieve eye-camera pose invariance when using glints' and pupil' center information. Villanueva and Cabeza [115] later showed that by also exploiting pupil shape (in contrast to only its center), a single glint suffices. Nevertheless, in practice a

larger amount of glints are employed because *a)* these methods are significantly sensitive to glint position estimate innacuracies, requiring averaging over multiple glints to alleviate this sensitivity [116], *b)* glints are only specularly reflected and usable if they fall on the cornea [117], and *c)* glints are easily occluded by eyelids, eyelashes, and reflections [118], [119]. Nevertheless, glints can also introduce several artifacts to the gaze data [120], [121]. Moreover, these glint-based approaches do not handle well glasses nor extraneous illuminations (e.g., in outdoor environments) [23], making them unsuitable candidates for ubiquitous eye tracking.

In general, VOG only allows one to probe the *optical axis* or Line of Gaze (LOG) – i.e., the line passing through the pupil, cornea, and eye centers. However, this axis differ from our actual *visual axis* or Line of Sight (LOS) – i.e., the line passing through the fovea, cornea center, and POR in which we have sharpest vision. The deviation between these axes (known as *kappa angle*) is user-dependent and can be as large as 5° [122], requiring at least a one-point *individual calibration* to estimate [123]. The aforementioned optical and visual axes estimates are in eye camera coordinates. Thus, for head-mounted eye trackers, it is also necessary to transform these to field camera coordinates. This transformation consists of applying the rotation and translation that align the eye camera to the field camera. If the eye tracker has a rigid-body and *extrinsic calibration* is available, this rotation and translation are known; alternatively, they can be estimated through a multi-point *gaze-mapping* calibration as proposed by Mansouryar et al. [97].

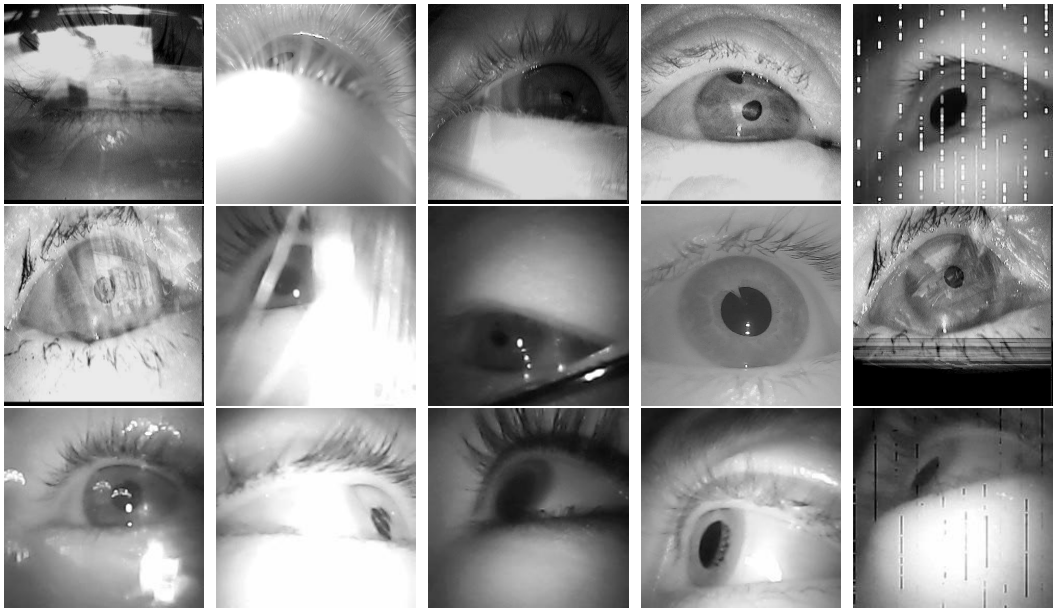
3 Pupil Detection and Tracking

> "*There is nothing like looking, if you want to find something. You certainly usually find something, if you look, but it is not always quite the something you were after.*"
>
> —J.R.R. Tolkien

Head-mounted video-based eye trackers are becoming increasingly more accessible and prevalent. For instance, such eye trackers are now available as low-cost devices (e.g., [124]) or integrated into wearables such as Google Glasses, Microsoft Hololens, and the Oculus Rift [125]–[127]. As a consequence, eye trackers are no longer constrained to their origins as research instruments but are developing into fully fledged pervasive devices. Therefore, guaranteeing that these devices are able to seamlessly operate in *out-of-the-lab* scenarios is not only pertinent to the research of human perception, but also to enable further applications such as pervasive gaze-based HCI [22], health monitoring [36], foveated rendering [128], and conditionally ADAS [13].

Pupil detection is the fundamental layer in the eye-tracking stack as virtually all subsequent layers rely on the signal generated by this layer – e.g., for gaze estimation [51], model construction [111], and automatic identification of eye movements [66]. Thus, errors in the pupil detection layer propagate to other layers, systematically degrading eye-tracking performance. Unfortunately, robust real-time pupil detection in natural environments has remained an elusive challenge. An elusiveness that is evidenced by several reports of difficulties and low pupil detection rates in natural environments such as driving [129]–[134], museum visit [53], shopping [135], walking [136], [137], in an operating room [138], and during human-robot interaction [55]. These difficulties in pupil detection stems from multiple factors; for instance, reflections (Fig. 3.1a), occlusions (Fig. 3.1b), complex illuminations (Fig. 3.1c), physiological irregularities (Fig. 3.1d), and even camera noise (Fig. 3.1e) [50], [139]–[141].

In this chapter we introduce, describe and evaluate two separate methods: 1) Pupil Reconstructor [6] (PuRe), a method for robust and real-time pupil *detection* (Section 3.1), and 2) Pupil Reconstructor and Subsequent Tracking [7] (PuReST), a method for robust and real-time pupil *tracking* (Section 3.2). Before going further, it is worth clarifying the distinction we make between a pupil *detector* and a pupil *tracker*: Although both aim at locating the pupil in an image, *detectors* use the information of a single frame, whereas *trackers* use information from the current and previous frames. In this thesis, we treat these as separate – albeit related – tasks, and, thus, each method is described in a distinct self-contained section including the appropriate related work for its algorithm class. Both



(a) Reflections **(b)** Occlusions **(c)** Illuminations **(d)** Irregularities **(e)** HW Noise

Figure 3.1: Representative images of pupil detection challenges in real-world scenarios: *(a)* reflections in the glasses, cornea, or contact lenses, *(b)* occlusions from eyelids, eyelashes, hair, contact lenses, glasses frames, *(c)* complex illuminations, such as low contrast or illumination gradients, *(d)* physiological variances, such as non-circular pupils, additional dark blobs in the iris, and *(e)* hardware noise from analog (Dikablis) and digital (Pupil Labs) cameras.

of these can be used for *pupil tracking*: The first as a form of *tracking-by-detection*, and the second as a form of *detection-by-tracking*. Complementarily to the two introduced algorithms, Section 3.3 investigates the influence of pupil tracking algorithm choice in the resulting gaze signal.

3.1 Pupil Detection

3.1.1 Related Work

While there is a plethora of previous work for pupil detection, most methods are not suitable for *out-of-the-lab* scenarios. For an extensive appraisal of state-of-the-art pupil detection methods, we refer the reader to the works by Fuhl et al. [50] and Tonsen et al. [4] for head-mounted eye trackers as well as Fuhl et al. [62] for remote eye trackers. In this work, we focus solely on methods that have been shown to be robust enough for deployment in pervasive scenarios, namely Ellipse Selector [1] (ElSe), Exclusive Curve Selector [3] (ExCuSe), and Świrski's pupil detection method [142] (Świrski). These are described in the sequence:

ElSe consists of two approaches. First, a Canny edge detector [143] is applied, and the

resulting edges are filtered through morphological operations[1]. Afterwards, ellipses are fit to the remaining edges, edges are removed based on empirically defined heuristics, and one ellipse is selected as pupil based on its roundness and enclosed intensity value. If this method fails to produce a pupil, a second approach that combines a mean and a center surround filter to find a coarse pupil estimate is employed; an area around this coarse estimate is then thresholded with an adaptive parameter, and the center of mass of pixels below the threshold is returned as pupil center estimate [1].

ExCuSe first analyzes the input images w.r.t. reflections based on peaks in the intensity histogram. If the image is determined to be reflection free, the image is thresholded with an adaptive parameter, and a coarse pupil position is estimated through an angular integral projection function [144]; this position is then refined based on surrounding intensity values. If a reflection is detected, a Canny edge detector is applied, and the resulting edges are filtered with morphological operations; ellipses are fit to the remaining edges, and the pupil is then selected as the ellipse with the darkest enclosed intensity [3].

Świrski starts with a coarse positioning using Haar-like features [145]. The intensity histogram of an area around the coarse position is clustered using *k-means* clustering [146], followed by a modified Random Sample Consensus [8] (RANSAC)-based ellipse fit [142].

From these algorithms, ElSe has shown a significantly better performance over multiple data sets [50]. Moreover, it is worth noticing that these algorithms employ multiple parameters that were empirically defined, albeit there is usually no need to tune these parameters.

It is worth dedicating part of this section to discuss machine-learning approaches in contrast to the algorithmic ones, particularly Convolutional Neural Networks (CNNs). Similarly to other computer vision problems, from a solely pupil detection stand point, deep CNNs will likely outperform human-crafted pupil detection approaches given enough training data – with incremental improvements appearing as more data becomes available and finer network tuning. Besides labeled data availability, which might be alleviated with developments of unsupervised learning methods, there are other impediments to the use of CNNs in pervasive scenarios since these scenarios typically require the use of embedded systems. For instance, computation time and power consumption. While these impediments might be lessened with specialized hardware – e.g., *cuDNN* [147], *Tensilica Vision DSP* [148], such hardware might not always be available or may incur prohibitive additional production costs. Finally, CNN-based approaches might be an interesting solution from an engineering point of view but remain a *black box* from the scientific one. To date, we are aware of two previous works that employ CNNs for pupil detections: 1) PupilNet [65] (PupilNet), which aims at a computationally inexpensive solution in the absence of hardware support, and 2) Vera-Olmos [149] (Vera-Olmos), which consists of two very deep CNNs – a coarse estimation stage (with 35 convolution plus 7 max-pooling layers for encoding and 10 convolution

[1] [1] also describe an algorithmic approach to edge filtering producing similar results; however the morphological approach is preferred because it requires less computing power.

plus 7 deconvolution layers for decoding), and a fine estimation stage (with 14 convolution plus 5 max-pooling layers for encoding and 7 convolution plus 5 deconvolution layers).

3.1.2 *Pupil Reconstructor [6]* (PuRe)

Similarly to related work, the proposed method was designed for *near-infrared*[2] eye images acquired by head-mounted eye trackers. Our method only makes two uncomplicated assumptions to constrain the valid pupil dimension space without requiring empirically defined values: 1) the eye canthi lie within the image, and 2) the eye canthi cover at least two-thirds of the image diagonal. It is worth noticing that these are *soft* assumptions – i.e., the proposed method still operates satisfactorily if the assumptions are not significantly violated. Fig. 3.2 illustrates these concepts. Furthermore, these assumptions are in accordance to eye tracker placement typically suggested by eye tracker vendor's guidelines to capture the full range of eye movements.

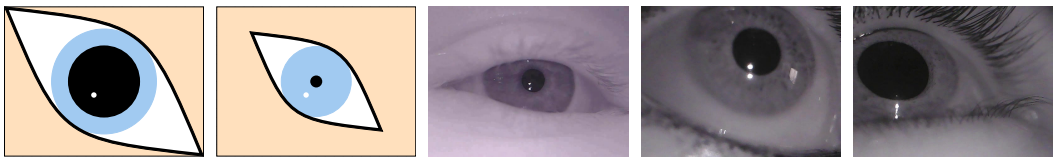


Figure 3.2: PuRe assumptions visualized. (a) illustrates the maximal intercanthal distance, yielding the maximal pupil diameter (pd_{max}), whereas (b) illustrates the lower bound – i.e., minimal pupil diameter (pd_{min}). (c) shows realistic data that respects both assumptions. In contrast, the maximal intercanthal distance assumption is violated in (d) and (e). In the former, the pupil does not approach maximal dilation, and PuRe is still able to detect the pupil. In the latter, the pupil is significantly dilated, and the resulting diameter exceeds pd_{max}; PuRe does not detect such pupils [6].

PuRe works *purely* based on edges, selecting curved edge segments that are likely to be significant parts of the pupil outline. These selected segments are then conditionally combined to construct further candidates that may represent a reconstructed pupil outline. An ellipse is fit to each candidate, and the candidate is evaluated based on its ellipse aspect ratio, the angular spread of its edges relative to the ellipse, and the ratio of ellipse outline points that support the hypothesis of it being a pupil. This evaluation yields a confidence measure for each candidate to be the pupil, and the candidate with the highest confidence measure is then selected as pupil. The remainder of this section describes the proposed method in detail.

3.1.2.1 Preprocessing

Prior to processing, if required, the input image is downscaled to the working size $S_w = (W_w \times H_w)$ through bilinear interpolation, where W_w and H_w are the working width and

[2] This is the standard image format for head-mounted eye trackers and can be compactly represented as a grayscale image.

height, respectively. The original aspect ratio is respected during downscaling. Afterwards, the resulting image is linearly normalized using a *Min-Max* approach.

3.1.2.2 Edge Detection and Morphological Manipulation

PuRe's first step is to perform edge detection using a Canny edge operator [143]. The resulting edge image is then manipulated with a morphological approach to thin and straighten edges as well as to break up orthogonal connections following the procedure described by [1]. The result of this step is an image with unconnected and thinned edge segments, as illustrated in Fig. 3.3.

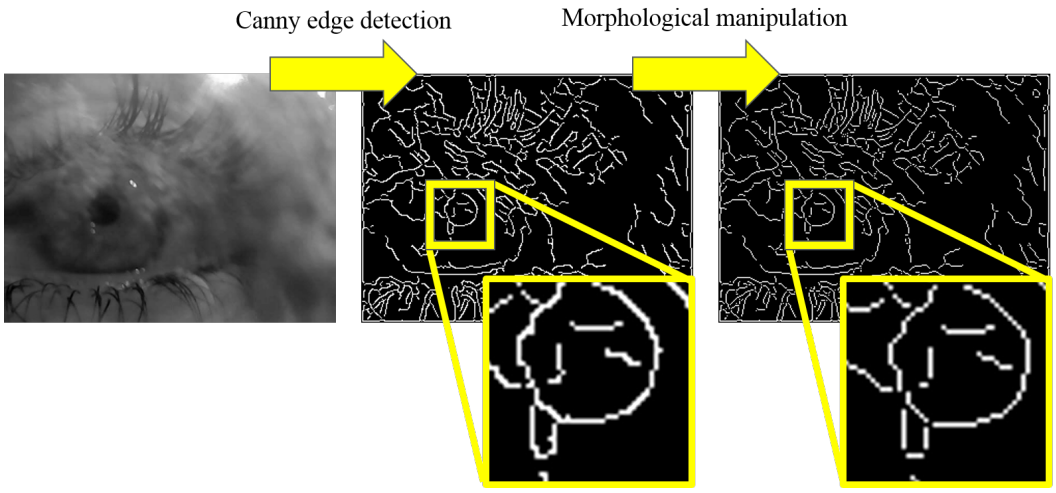


Figure 3.3: Input image (left), resulting Canny edge detection (middle), and edges after morphological manipulation. Notice how the edges are thinned and orthogonal connections are broken [6].

3.1.2.3 Edge Segment Selection

Each edge segment is first approximated by a set of dominant points D following the *k-cosine* chain approximation method described by [150]. This approximation reduces the computational requirements for our approach and typically results in a better ellipse fit in cases where a pupil segment has not been properly separated from surrounding edges. After approximation, multiple heuristics are applied to discard edge segments that are not likely to be part of the pupil outline:

1. Given the general conic equation $ax^2+by^2+cxy+dx+ey+f=0$, at least five points are required to fit an ellipse in a least-squares sense. Therefore, we exclude segments in which D's cardinality is smaller than five. This heuristic discards plain shapes such as small segments and substantially straight lines.

2. Based on the assumptions highlighted in the beginning of this section, it is possible to establish the maximal and minimal distance between the lateral and medial eye

canthus in pixels when frontally imaged as

$$ec_{max} = \sqrt{{W_w}^2 + {H_w}^2} \quad \text{and} \quad ec_{min} = \frac{2}{3} * ec_{max}. \tag{3.1}$$

These estimates can then be used to infer rough values for the maximal (Fig. 3.2a) and minimal (Fig. 3.2b) pupil diameter bounds (pd_{max} and pd_{min}) based on the human physiology. We approximate the eye canthi distance through the palpebral fissure width as 27.6 mm [151]; similarly, the maximal and minimal pupil diameter are approximated as 8 mm and 2 mm, respectively [152]. Therefore,

$$pd_{max} \approx 0.29 * ec_{max} \quad \text{and} \quad pd_{min} \approx 0.07 * ec_{min}. \tag{3.2}$$

Note that whereas maximal values hold independent of camera rotation and translation w.r.t. the eye, minimal values might not hold due to perspective projection distortions and corneal refractions. Nonetheless, pd_{min} already represents a minute part ($\approx 4.8\%$) of the image diagonal, and we opted to retain this lower bound – for reference, see Fig. 3.2b. For each candidate, we approximate the segment's diameter by the largest gap between two of its points. Candidates with a diameter outside of the range $[pd_{min}, pd_{max}]$ violate bounds and are thus discarded.

3. To estimate a segment's curvature, first the minimum rectangle containing D is calculated using the *rotating calipers* method [153]. The curvature is then estimated based on the ratio between this rectangle's smallest and largest sides. The straighter the candidate is, the smaller the ratio. The cut-off threshold for this ratio is based on the ratio between the minor and major axes of an ellipsis with axes extremities inscribed in 45° of a circle, which evaluates to $R_{th} = (1 - cos(22.5°))/sin(22.5°) \approx 0.2$. This heuristic servers to discard relatively linear candidates.

4. At this stage, an ellipse E is fit to the points in D following the least-squares method described in [154]. A segment is discarded if: I) E's center lies outside of the image boundaries, which violates PuRe's assumptions, or II) the ratio between E's minor and major axes is smaller than R_{th}, which assumes that the camera pose relative to the eye can only distort the pupil round shape to a certain extend.

5. Seldom, the ellipse fitting procedure will not produce a proper fit. We identify and discard most such cases inexpensively if the mean point from D does not lie within the polygon defined by the extremities of E's axes.

As a result from this elimination process, the edge segment search space is significantly reduce, as illustrated by Fig. 3.4.

3.1.2.4 Confidence Measure

For each remaining candidate, PuRe takes into account three distinct metrics to determine a confidence measure ψ that the candidate is a pupil:

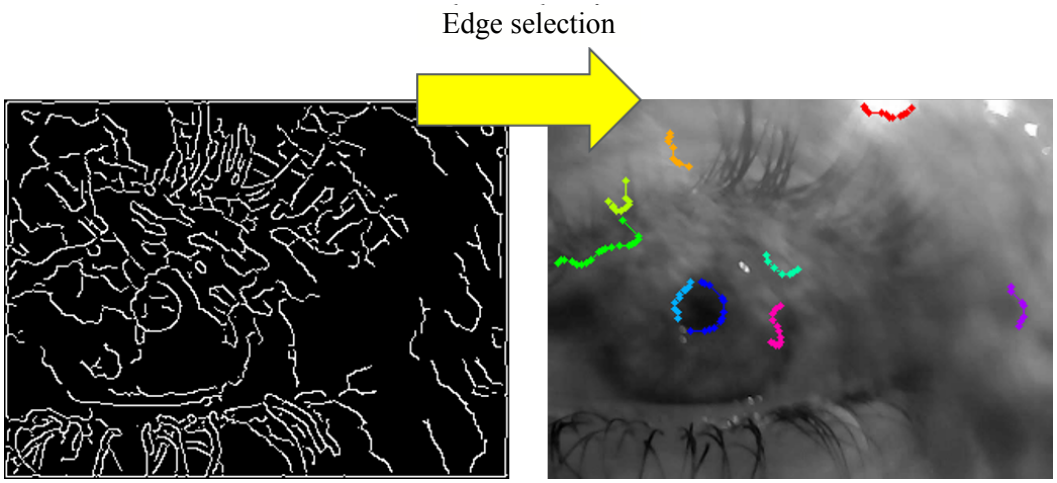


Figure 3.4: Edges after morphological processing (left) and the resulting selected segments that are candidates for the pupil outline (right). Each segment is represented by its *k-cosine* chain approximation and illustrated with a distinct color [6].

Ellipse Aspect Ratio (ρ): measures the roundness of E. This metric favors rounder ellipses (that typically result due to the eye camera placement w.r.t. the eye) and is evaluated as the ratio between E's minor and major axis.

Angular Edge Spread (θ): measures the angular spread of the points in D relative to E, assuming that the better distributed the edges are, the more likely it is that the edges originated from a clearly defined elliptical shape (i.e., a pupil's shape). This metric is roughly approximated as the ratio of E centered quadrants that contain a point from D.

Ellipse Outline Contrast (γ): measures the ratio of the E's outline that supports the hypothesis of a darker region surrounded by a brighter region (i.e., a pupil's appearance). This metric is approximated by selecting E's outline points with a stride of ten degrees. For each point, the linear equation passing through the point and the E's center is calculated, which is used to define a line segment with length proportional to E's minor axis and centered at the outline point. If the mean intensity of the inner segment is lower than the mean intensity of the outer one, the point supports the pupil-appearance hypothesis[3].

If the candidate's ellipse outline is invalid – i.e., violates PuRe's size assumptions or less than half of the outline contrast γ supports the candidate – the confidence metric is set to zero. Otherwise, the aforementioned metrics are averaged when determining the resulting

[3] If a bright pupil eye tracker is used, the inverse holds

confidence. In other words,

$$\psi = \begin{cases} 0 & \text{if the outline is invalid;} \\ \frac{\rho+\theta+\gamma}{3} & \text{otherwise.} \end{cases} \tag{3.3}$$

It is worth noticing that the range of all three metrics (and consequently ψ) is [0,1].

3.1.2.5 Conditional Segment Combination

The segments that remain as candidates are combined pairwise to generate additional candidates. This procedure attempts to reconstruct the pupil outline based on nearby segment pairs since the pupil outline is often broken up due to occlusions from, for example, reflections or eye lashes. Let D_1 and D_2 be the set of dominant points for two segments and S_1 and S_2 the set of points contained by the up-right squares bounding D_1 and D_2, respectively. The segments are combined if these bounding squares intersect but are not fully contained within one another – i.e., $S_1 \cap S_2 \neq \varnothing \neq S_1 \neq S_2$. For instance, see Fig. 3.5. The resulting merged segment is then validated according to Section 3.1.2.3, and its confidence measure evaluated according to Section 3.1.2.4. Since this procedure is likely to produce candidates with high aspect ratio ρ and angular spread θ values, the new candidate is only added to the candidate list if its outline contrast γ improves on the γ from the original segments. After conditional combination, the candidate with highest confidence ψ is selected as the initial pupil, as shown in Fig. 3.6.

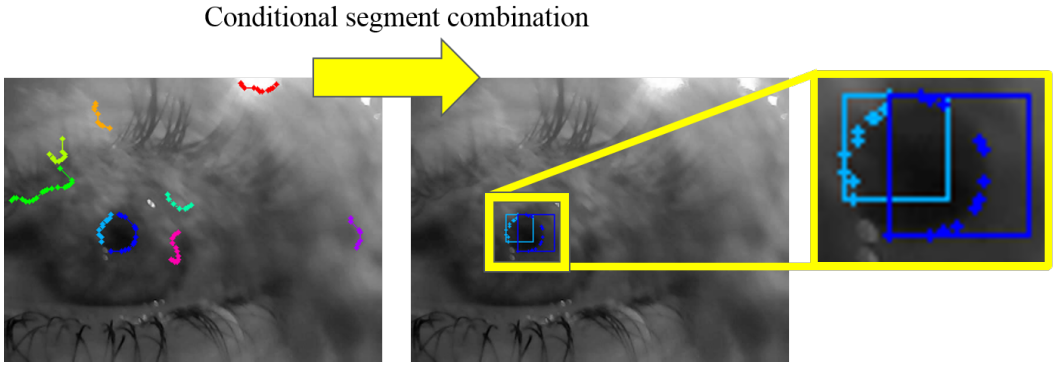


Figure 3.5: Illustration of the conditional segment combination. The highlighted blue and cyan segments meet the intersection requirements and are combined to generate and additional pupil outline segment. The other pairs do not intersect and, therefore, generate no additional candidates [6].

Note that the inner intensities relative to other candidates do not contribute to the pupil selection. Thus, the iris might be selected since it exhibits properties similar to the pupil – e.g., roundness, inner-outer contrast, and size range. For this reason, the inside of the initial pupil is searched for a roughly cocentered candidate with adequate size and strong inner-outer outline contrast. This is done by searching a circular area centered at the center of the initial pupil with radius equal to the initial semi-major axis – i.e., representing a

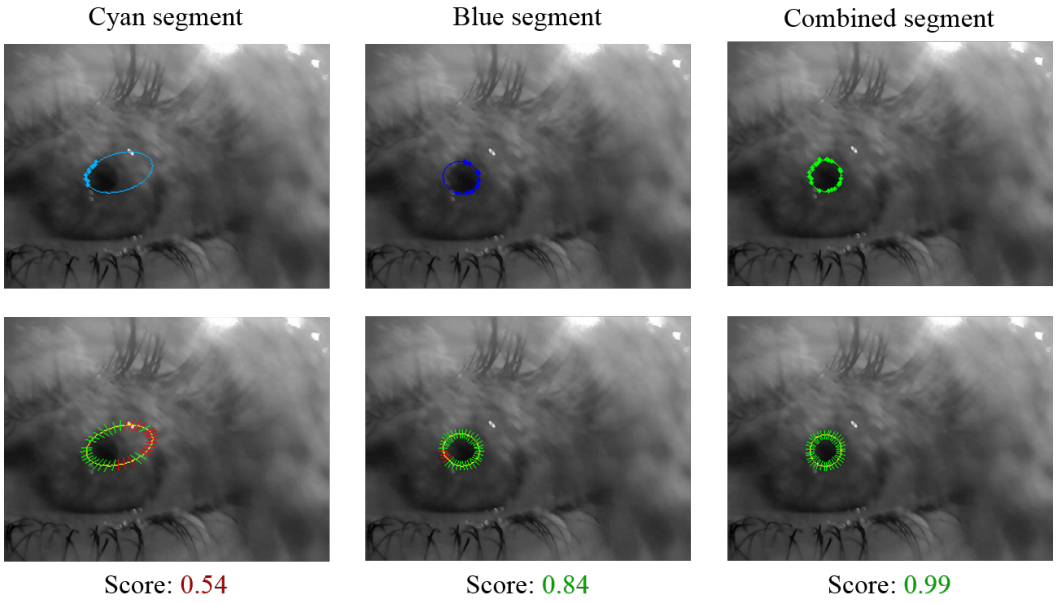


Figure 3.6: Illustration for the confidence measure evaluation using the segments from Fig. 3.5. Segments with confidence smaller than 0.5 are omitted. The first row shows the segment points and resulting ellipse. Second row shows the lines contributing to the ellipse outline contrast (γ); green lines support the pupil-appearance hypothesis, whereas red lines do not. Notice how the cyan segment results in an incorrect outline estimation due to the ellipse fit even though the segment is part of the pupil outline. The blue segment results in an acceptable outline estimate even though the left side of the outline is slightly shifted. In contrast, the combined segment *reconstructs* the whole range of the pupil outline and yields a higher confidence (ψ), thus being selected as pupil estimate [6].

circular iris. Candidates 1) lying inside this area, 2) with major axis smaller than the search radius, and 3) with at least three thirds of the outline contrast (γ) valid are collected. The collected candidate with highest confidence is then chosen as new the pupil estimate. If no candidate is collected in this procedure, the initial pupil remains as the pupil estimate. As output, PuRe returns not only a pupil center, but also its outline and a confidence metric.

3.1.3 Experimental Evaluation

As previously mentioned, we evaluate PuRe only against robust state-of-the-art pupil detection methods, namely ElSe[4], ExCuSe, and Świrski. All algorithms were evaluated using their open-source C++ implementations; default parameters were employed unless specified otherwise. For ExCuSe, the input images were downscaled to *240p* (i.e., 320×240 px) as there is evidence that this is a favorable input size detection-rate-wise [4]. Similarly, the working size for PuRe (S_w, Section 3.1.2.1) was set to *240p* as well to keep run time com-

[4] With morphological split and validity threshold.

patible with state-of-the-art head-mounted eye trackers (see Section 3.1.7). ElSe provides an embedded downscaling and border cropping mechanism, effectively operating with a resolution of 346×260 px. Notice that whenever the input images are downscaled, the results must be upscaled to be compared with the ground truth. No preprocessing downscaling was performed for Świrski since evidence suggests it degrades performance for this method [4]. Additionally, we juxtapose our results with the ones from PupilNet [65] and Vera-Olmos [149] whenever possible.

In this work, we use the term **use case** to refer to each individual eye video. For instance, the Labelled Pupils in the Wild [4] (LPW) data set contains 22 subjects with three recordings per subject in distinct conditions (e.g., indoors, outdoors), resulting in 66 distinct *use cases*. Furthermore, we often compare PuRe with the **rival**, meaning the best performant from the other algorithms for the metric in question. For instance, for the aggregated detection rate, ElSe performs better than ExCuSe and Świrski and is, therefore, the *rival*.

3.1.4 Pupil Detection Rate

A pupil is considered detected if the algorithm's pupil center estimate lies within a radius of n pixels from the ground-truth pupil center. Similar to previous work, we focus on an error up to five pixels to account for small deviations in the ground-truth labeling process – e.g., human inaccuracy [1], [3], [4], [149]. This error magnitude is illustrated in Fig. 3.7. For this evaluation, we employed five data sets totaling 266,786 realistic and challenging images acquired with three distinct head-mounted eye tracking devices, namely, the Świrski [142], ExCuSe [3], ElSe [1], LPW [4], and PupilNet [65] data sets. In total, these data sets encompass 99 distinct *use cases*. It is worth noticing that we corrected[5] a disparity of one frame in the ground truth for five *use cases* of the ElSe data set and for the whole PupilNet data set, which increased the detection rate of all algorithms (by $\approx 3.5\%$ on average).

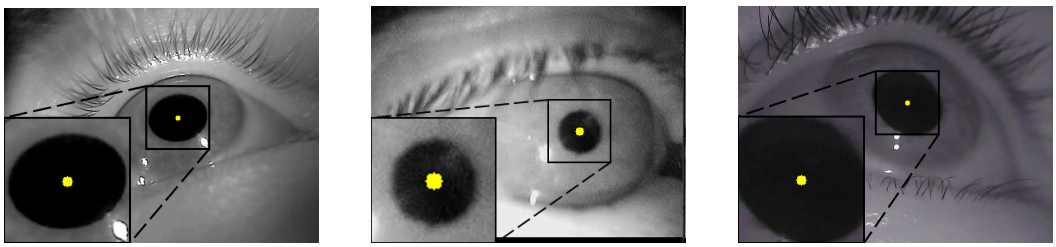


Figure 3.7: Five pixels validity range (in yellow) around the ground-truth pupil center for the pupil estimate to be considered correct and, thus, the pupil detected. Reference range relative to the data from the Świrski (left), ExCuSe/ElSe/PupilNet (center), and LPW (right) data sets [6].

Fig. 3.8 shows the cumulative detection rate per pixel error of the evaluated algorithms for the aggregated 266,786 images as well as the detection rate distribution per *use case* at five pixels. As can be seen, PuRe outperforms all algorithmic competitors for all pixel errors. In particular, PuRe achieved a detection rate of 72.02% at the five pixel error

[5] All ground truth data employed in this work are available at www.ti.uni-tuebingen.de/perception

mark, further advancing the state-of-the-art detection rate by a significant margin of 6.46 percentage points when compared to the *rival*. Moreover, the proposed method estimated the pupil center correctly 80% of the time for the majority of *use cases*, attesting for PuRe's comprehensive applicability in realistic scenarios.

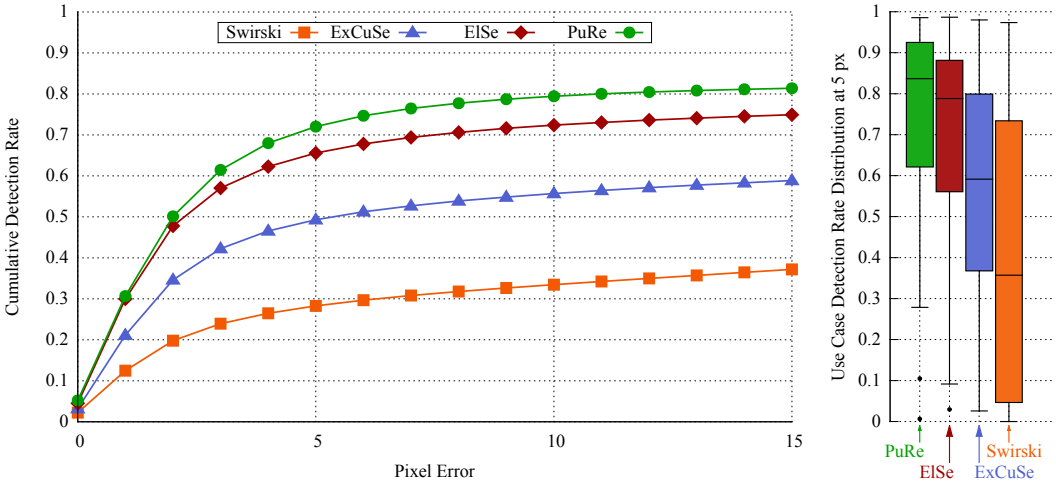


Figure 3.8: On the left, the cumulative detection rate for the aggregated 266,786 images from all data sets. On the right, the distribution of the detection rate per *use case* as a *Tukey boxplot* [155] [6].

It is worth noticing that the aggregated detection rate does not account for differences in data set sizes. As a consequence, this metric is dominated by the ExCuSe and LPW datasets, which together represent 63.44% of the data. Since these two data sets are *not the most challenging ones*[6], the algorithms tend to perform better on them, and differences between the algorithms are less pronounced. Inspecting the detection rates per data set in Fig. 3.9 gives a better overview of the real differences between the algorithms and data sets, revealing that PuRe improves the detection rate by more than 10 percentage points w.r.t. the *rival* for the *most challenging data sets* (i.e., ElSe and PupilNet). To allow for a more fine-grained appreciation of the method's performance relative to the other algorithms, Fig. 3.10 presents PuRe's detection rate at five pixels relative to the *rival* for each *use case*. In 71.72% of all *use cases*, PuRe outperformed all contenders. In particular, for the two most challenging data sets, PuRe surpassed the competition in 100% of the *use cases*. In contrast, the rivals noticeably outperformed PuRe in five *use cases*: `Swirski/p1-right`, `ExCuSe/data-set-II`, `LPW/4/12`, `LPW/9/17`, and `LPW/10/11`, from which representative frames are shown in Fig. 3.11. These five *use cases* also highlight some of PuRe's imperfections. For instance, `Swirski/p1-right` and `ExCuSe/data-set-II` have weak and broken pupil edges due to inferior illumination and occlusions due to eye lashes/corneal reflections; ElSe compensates this lack of edges with its second step. `LPW/4/12` contains large pupils that violate PuRe's assumptions; in fact, relaxing the maximum pupil size by only ten percent increases PuRe's detection rate from 44.2% to 65.5% (or +6.55% w.r.t.

[6] As evidenced by higher detection rates for all algorithms in Fig. 3.9

the *rival*). `LPW/9/17` often has parts of the pupil outline occluded by eye lashes and reflections, whereas `LPW/10/11` contains pupils in extremely off-axial positions combined with occlusions caused by reflections. However, visually inspecting the latter two *use cases*, we did not find any particular reason for Świrski to outperform the other algorithms.

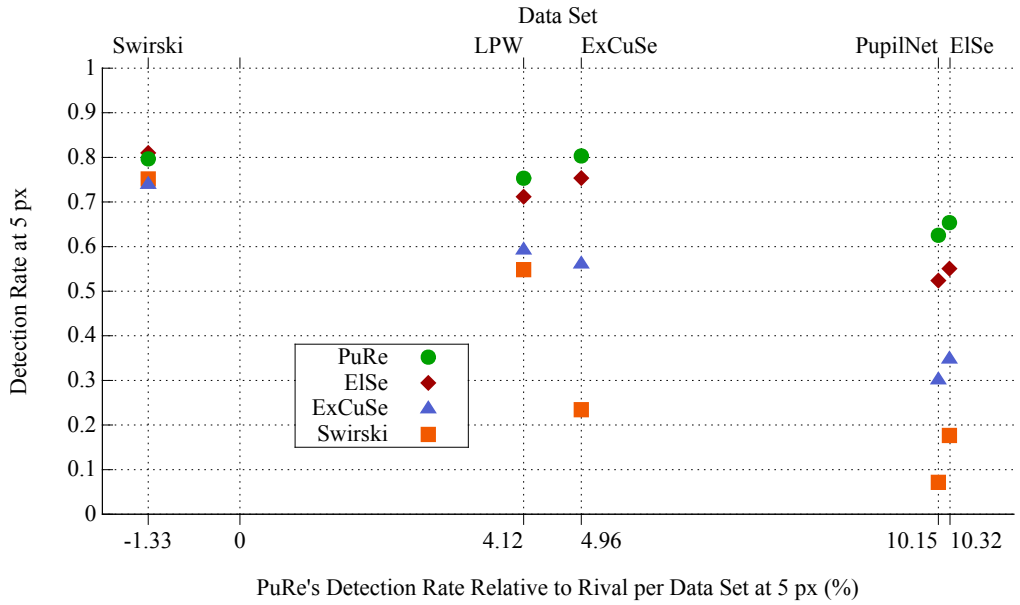


Figure 3.9: Detection rate per data set plotted against PuRe's performance relative to the *rival*. The lower the points, the harder the data set; the further right the points, the larger PuRe's performance w.r.t. the *rival* is. Notice that as the data sets become more difficult (i.e., the detection rate decreases for all algorithms), the gap between PuRe and the other algorithms increases [6].

Regarding CNN-based approaches[7]: 1) For PupilNet, [65] report a detection rate of 65.88% at the five pixel error range when trained in half of the data from the ExCuSe and PupilNet data sets and evaluated on the remaining data. In contrast, PuRe reached 71.11% on all images from these data sets – i.e., +5.23%. 2) For Vera-Olmos, [149] report an unweighted[8] detection rate of 82.17% at the five pixel error range averaged over a *leave-one-out* cross validation in the ExCuSe and ElSe data sets. In contrast, PuRe reached 76.71% on all images from these data sets – i.e., −5.46%. Nevertheless, these results indicate that PuRe is able to compete with state-of-the-art CNN-based approaches while requiring only a small fraction of CNN computational requirements. In fact, PuRe outperformed Vera-Olmos for 37.5% of use cases. Furthermore, it is worth noticing that the training data is relatively similar to the evaluation data (same eye tracker, similar conditions and positioning) in both cases, which might bias the results in favor of the CNN approaches.

[7] These approaches used the uncorrected ElSe and PupilNet data sets, which might slightly affect the detection rate.

[8] Averaged over the *use cases*.

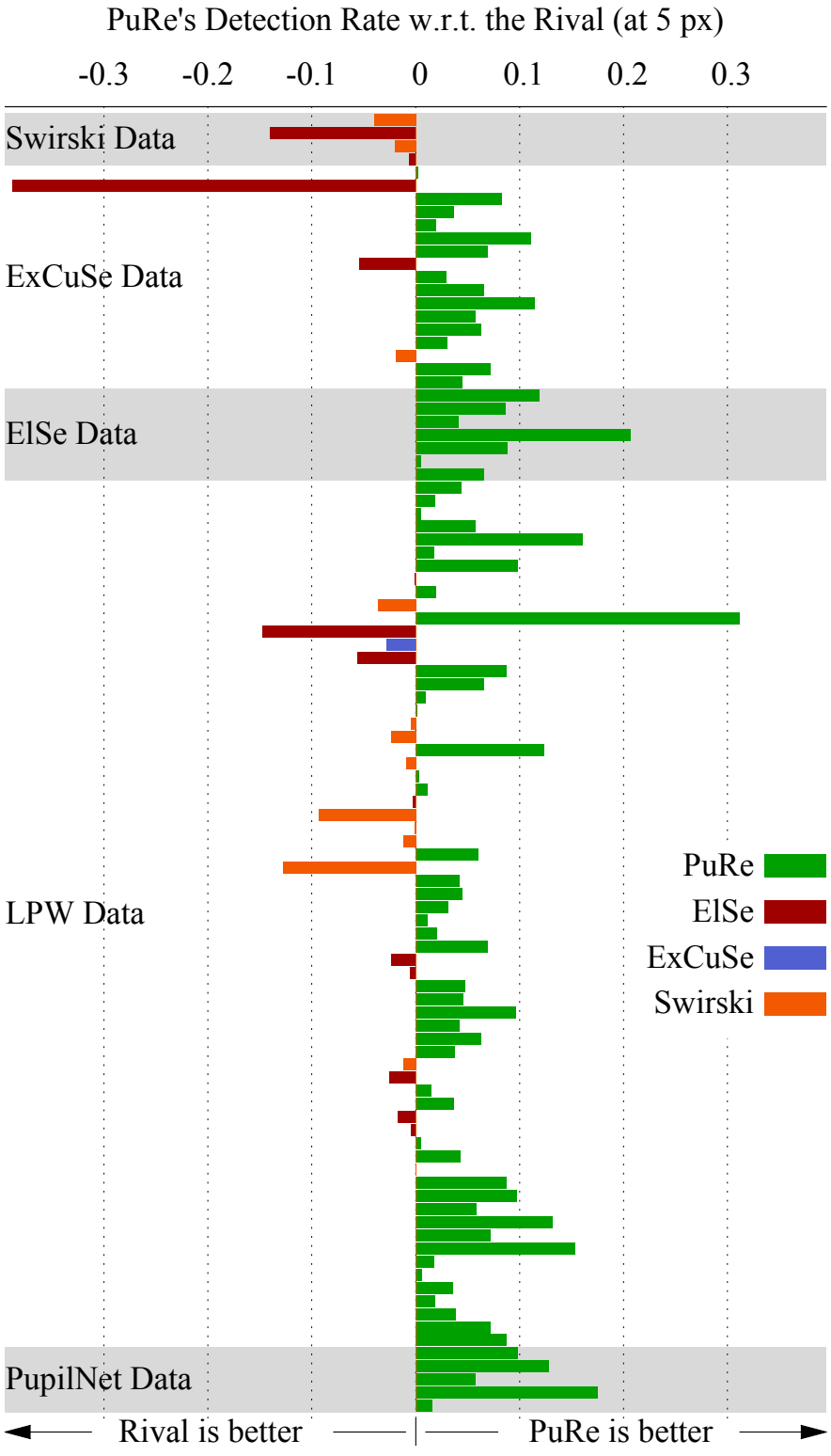


Figure 3.10: PuRe's performance relative to the rival for each *use case*. PuRe is the best algorithm in 71.72% of cases, ElSe in 14.14%, Świrski in 12.12%, and ExCuSe in 1.01% [6].

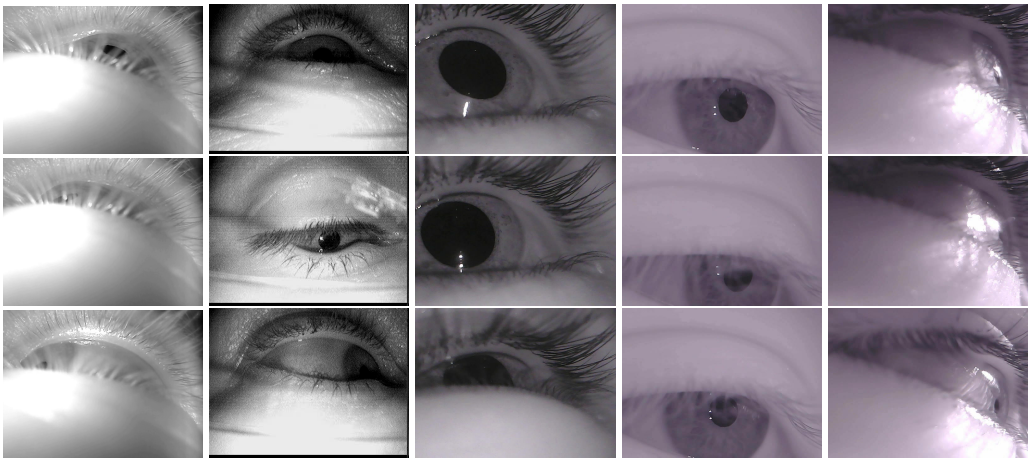


Figure 3.11: Representative frames for *use cases* in which the *rival* outperforms PuRe. Each column contains frames from one *use case*. From left to right: `Swirski/p1-right`, `ExCuSe/data-set-II`, `LPW/4/12`, `LPW/9/17`, and `LPW/10/11` [6].

3.1.5 Beyond Pupil Detection Rate: Improving Precision, and Specificity Through the Confidence Measure

One aspect that is often overlooked when developing pupil detection algorithms is the rate of incorrect pupil estimates returned by the algorithm. For instance, the aforementioned CNN-based approaches *always* return a pupil estimate, regardless of one actually existing in the image. Intuitively, one can relax pupil appearance constraints in order to increase the detection rate, leading to an increase in the amount of incorrect pupils returned. However, these incorrect pupil estimates later appear as noise and can significantly degrade gaze-estimation calibration [24], automatic eye movement detection [156], glanced-area ratio estimations [157], eye model construction [111], or even lead to wrong medical diagnosis [158]. Therefore, it is imperative to also analyse algorithms in terms of incorrect detected pupils. The pupil detection task can be formulated as a classification problem – similar to the approach by [159] for frame-based tracking metrics – such that:

True Positive (*TP*) represents cases in which the algorithm and ground truth agree on the presence of a pupil. We further specialize this class into Correct True Positive (*CTP*) and Incorrect True Positive (*ITP*) following the detection definition from Section 3.1.4.

False Positive (*FP*) represents cases in which the algorithm finds a pupil although no pupil is annotated in the ground truth.

True Negative (*TN*) represents cases in which the algorithm and ground truth agree on the absence of a pupil.

False Negative (*FN*) represents cases in which the algorithm fails to find the pupil annotated in the ground truth.

Note that this is not a proper binary classification problem, and the *relevant* class is given only by *CTP*. Therefore, we redefine *sensitivity* and *precision* in terms of this class as

$$sensitivity = \frac{CTP}{TP + FN} \tag{3.4}$$

and

$$precision = \frac{CTP}{TP + FP} \tag{3.5}$$

respectively, such that *sensitivity* reflects the (correct) pupil detection rate and *precision* the rate of pupils that the algorithm found that are correct. Thus, these metrics allows us to evaluate 1) the trade-off between detection of correct and incorrect pupils, and 2) the meaningfulness of PuRe's confidence measure.

Unfortunately, the eye image corpus employed to evaluate pupil detection rates (in Section 3.1.4) do not include negative samples – i.e., eye images in which a pupil is not visible, such as during a blink. Therefore, the capability of the algorithm to identify frames without a pupil as such cannot be evaluated since *specificity* $\left(\frac{TN}{TN+FP}\right)$ remains undefined without negative samples. To evaluate this aspect of the algorithms, we have recorded a new data set (henceforth referred to as Closed-Eyes [6] (Closed-Eyes)) containing in its majority (99.49%) negative samples. This data set consists of 83 *use cases* and contains 49,790 images with a resolution of 384×288 px. These images were collected from eleven subjects using a *Dikablis Professional* eye tracker [160] with varying illumination conditions and camera positions. A larger appearance variation was achieved by asking the subjects to perform certain eye movement patterns[9] while their palpebrae remained shut in two conditions: 1) with the palpebrae softly shut, and 2) with the palpebrae strongly shut as to create additional skin folds. In $\approx 56\%$ of *use cases*, participants wore glasses. Challenges in the images include reflections, black eyewear frames, prominent eye lashes, makeup, and skin folds, all of which can generate edge responses that the algorithms might identify as parts of the pupil outline. Fig. 3.12 shows representative images from the data set.

Figure 3.12: Samples from the Closed-Eyes data set. First row shows samples from softly shut palpebrae, and the second one shows samples from strongly shut palpebrae [6].

[9] Although the eye is hidden underneath the palpebrae, eye globe movement results in changes in the folds and light reflections in the skin.

We evaluated the four aforementioned algorithms using all images from the data sets from Section 3.1.4 and the Closed-Eyes data set, totaling 316,576 images. We assessed PuRe's confidence measure using a threshold within [0:0.99] with strides of 0.01 units. A pupil estimate was considered correct only if its confidence measure was above the threshold. Similarly, ElSe offers a *validity threshold* (default=10) to diminish incorrect pupil rates, which we evaluated within the range [0:110] with strides of 10 units. ExCuSe and Świrski do not offer any incorrect pupil prevention mechanisms and, therefore, result only in a single evaluation point. The results from this evaluation are presented in Fig. 3.13 and Fig. 3.14. As can be seen in these figures, PuRe dominates over the other algorithms, and PuRe's confidence metric is remarkably meaningful, allowing to significantly reduce incorrect pupil detections while preserving the correct pupil detection rate and increasing identification of pupil-less frames. In fact, when compared to threshold 0, the threshold that maximizes the F_2 score (0.66) increased *precision* and *specificity* by 20.78% and 89.47%, respectively, whereas *sensitivity* was decreased by a negligible 0.49%. In contrast, ElSe exhibited negligible ($< 1\%$) changes for *sensitivity* and *precision* when varying the threshold from 0 to 10, with a small gain of 2.69% in *specificity*; subsequent threshold increments increase *specificity* at the cost of significantly deteriorating ElSe's performance for the other two metrics. Compared to the *rival* for each metric, $\text{PuRe}_{th=0.66}$ improved *sensitivity*, *precision*, and *specificity* by 5.96, 25.05, and 10.94 percentage points, respectively.

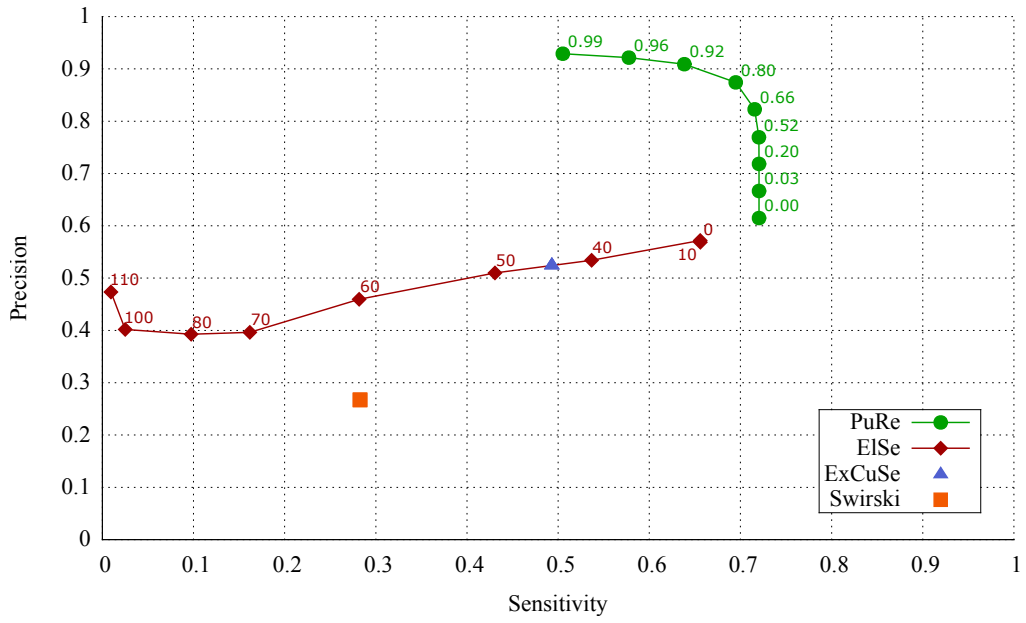


Figure 3.13: Trade-off between *sensitivity* and *precision* for different pupil validation thresholds for PuRe and ElSe. Algorithms were evaluated over all images from the Świrski, ExCuSe, ElSe, LPW, PupilNet, and Closed-Eyes data sets. For the sake of visibility, points are only plotted when there's a significant (> 0.05) change in one of the metrics. The Z_2 score is maximized at thresholds 0.66 (for PuRe) and 10 (for ElSe) [6].

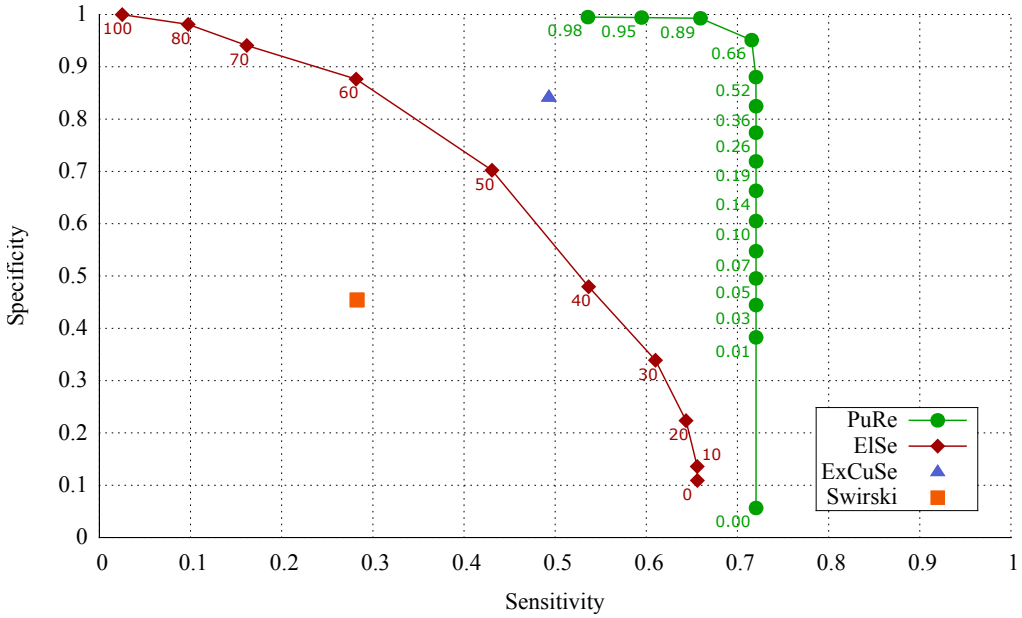


Figure 3.14: Trade-off between *sensitivity* and *specificity* for different pupil validation thresholds for PuRe and ElSe. Algorithms were evaluated over all images from the Świrski, ExCuSe, ElSe, LPW, PupilNet, and Closed-Eyes data sets. For the sake of visibility, points are only plotted when there's a significant (> 0.05) change in one of the metrics. The Z_2 score is maximized at thresholds 0.66 (for PuRe) and 10 (for ElSe) [6].

3.1.6 Pupil Signal Quality

From the point of view of the image processing layer in the eye-tracking stack, the (correct/incorrect) detection rates stand as a meaningful metric to measure the quality of pupil detection algorithms. However, the remaining layers (e.g., gaze estimation, eye movement identification) often see the output of this layer as a discrete *pupil signal* (as a single-object tracking-by-detection), which these detection rates do not fully describe. For example, consider two pupil detection algorithms: A_1, which detects the pupil correctly every two frames, and A_2, which detects the pupil correctly only through the first half of the data. Based solely on the pupil detection rate (50% in both cases), these algorithms are identical. Nonetheless, the former algorithm enables noisy[10] eye tracking throughout the whole data, whereas the latter enables noiseless eye tracking during only the first half of the data. Which algorithm is preferable is then application dependent, but a method to assess these properties is required nonetheless.

Recent analyses of widely-used object tracking performance metrics have shown that most existing metrics are strongly correlated and propose the use of only two weakly-correlated metrics to measure tracker performance: *accuracy* and *robustness* [161], [162].

[10] Note that the values are not necessarily missing but might be incorrect pupil detections; thus interpolation/smoothing might actually degrade the pupil signal even further.

Whereas in those works *accuracy* was measured by average region overlaps, for pupil detection data sets, only the pupil center is usually available. Thus, we employ the center-error-based *detection rate* as accuracy measure. As an indicator of *robustness*, [161] proposes the *failure rate* considering the tracking from a reliability engineering point of view as a supervised system in which an operator reinitializes the tracker whenever it fails. For the pupil signal, our formulation differs slightly since there is no operator reinitialization. Instead, we evaluate the *robustness* as the *reliability*

$$r = e^{-\lambda t}, \tag{3.6}$$

where $\lambda = \frac{1}{MTBF}$ is the failure rate estimated through the Mean Time Between Failures (*MTBF*) not accounting for *repair time* – i.e., periods of no/incorrect pupil detection are considered as latent faults. In this manner, the *reliability* is a measure of the likelihood of the algorithm correctly detecting the pupil for t successive frames. Furthermore, by measuring the Mean Time To Repair (*MTTR*) – i.e., the mean duration of periods in which the correct pupil signal is not available – we can achieve a similar metric in terms of the likelihood for the algorithm to *not* detect the pupil correctly for t successive frames. Henceforth, we will define this metric as the *insufficiency* (i), evaluated as

$$i = e^{-\kappa t}, \tag{3.7}$$

where $\kappa = \frac{1}{MTTR}$. The smaller an algorithm's *insufficiency*, the more *sufficient* it is. It is worth noticing, that r and i are not true probabilities since the events they measure are not likely to be independent nor uniformly distributed. Consequently, these metrics only offer a qualitative and relative measure between algorithms. Thus, we simplify their evaluation by fixing $t = 1$. As an illustration, let us return to our initial example considering a sequence of L frames: A_1 yields $r_{A_1} = i_{A_1} = e^{-1/1}$, whereas A_2 yields $r_{A_2} = i_{A_2} = e^{-1/(0.5L)}$. Since $\forall L > 2 \implies r_{A_1} < r_{A_2} \wedge i_{A_1} < i_{A_2}$, we can conclude that A_2 is more reliable but less sufficient w.r.t. A_1 for sequences longer than two frames. A quantitative conclusion is, however, not possible. For the sake of understandability, we further define *sufficiency* (s) as the complement of *insufficiency* such that

$$s = 1 - i. \tag{3.8}$$

In this manner, higher values are better for all metrics in this section.

We evaluated the four aforementioned algorithms in terms of *reliability* and *sufficiency* using only the data sets from Section 3.1.4. The Closed-Eyes data set was excluded since it is not realistic from the temporal aspect – i.e., users are not likely to have their eyes closed for extended periods of time. Furthermore, it is worth noticing that each *use case* from the ExCuSe, ElSe, and PupilNet data sets consists of images sampled throughout a video based on the pupil detection failure of a commercial eye tracker; these *use cases* can be seen as videos with a low and inconstant sampling rate. Results aggregated for all images are shown in Fig. 3.15 and indicate PuRe as the most reliable and sufficient algorithm. Curiously, the second most reliable algorithm was Świrski, indicating that during use cases in which it was able to detect the pupil, it produced a more stable signal than ElSe and ExCuSe –

although its detection rate is much lower relative to the other algorithms for challenging scenarios. This lower detection rate reflects on the *sufficiency*, in which Świrski is the worst performer; ElSe places second, followed by ExCuSe. Furthermore, Fig. 3.16 and Fig. 3.17 details *reliability* and *sufficiency* per *use case*, respectively. In this scenario, PuRe was the most reliable algorithm in 66.67% of the *use cases*, followed by Świrski (23.23%), ElSe (7.07%), and ExCuSe (3.03%). These results demonstrate that PuRe is more reliable not only when taking into account all images but also for the majority of *use cases*. This higher reliability also reflects on PuRe's longest period of consecutive correct pupil detections, which contained 859 frames (in `LPW/21/12`). In contrast, the longest sequence for the *rival* was only 578 frames (ExCuSe, also in `LPW/21/12`). ElSe's longest period was of 386 frames in `LPW/10/8`, for which PuRe managed 411 frames. In terms of *sufficiency*, ElSe had a small lead with 41.41% of *use cases*, closely followed by PuRe (40.40%); ExCuSe and Świrski were far behind, winning 14.14% and 4.04% of use cases, respectively. The advantage of ElSe here is likely due to its second pupil detection step, which might return the correct pupil during periods of mostly incorrect detections, fragmenting these periods into smaller ones.

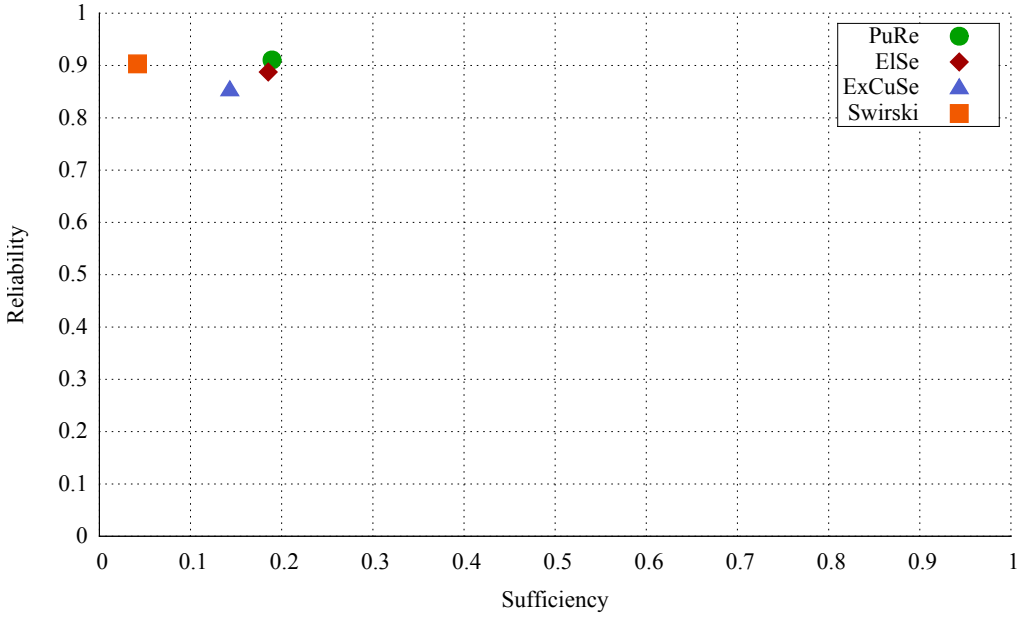


Figure 3.15: *Reliability* and *sufficiency* for all algorithms based on the sequence of all aggregated images from the Świrski, ExCuSe, ElSe, LPW, and PupilNet data sets – higher is better [6].

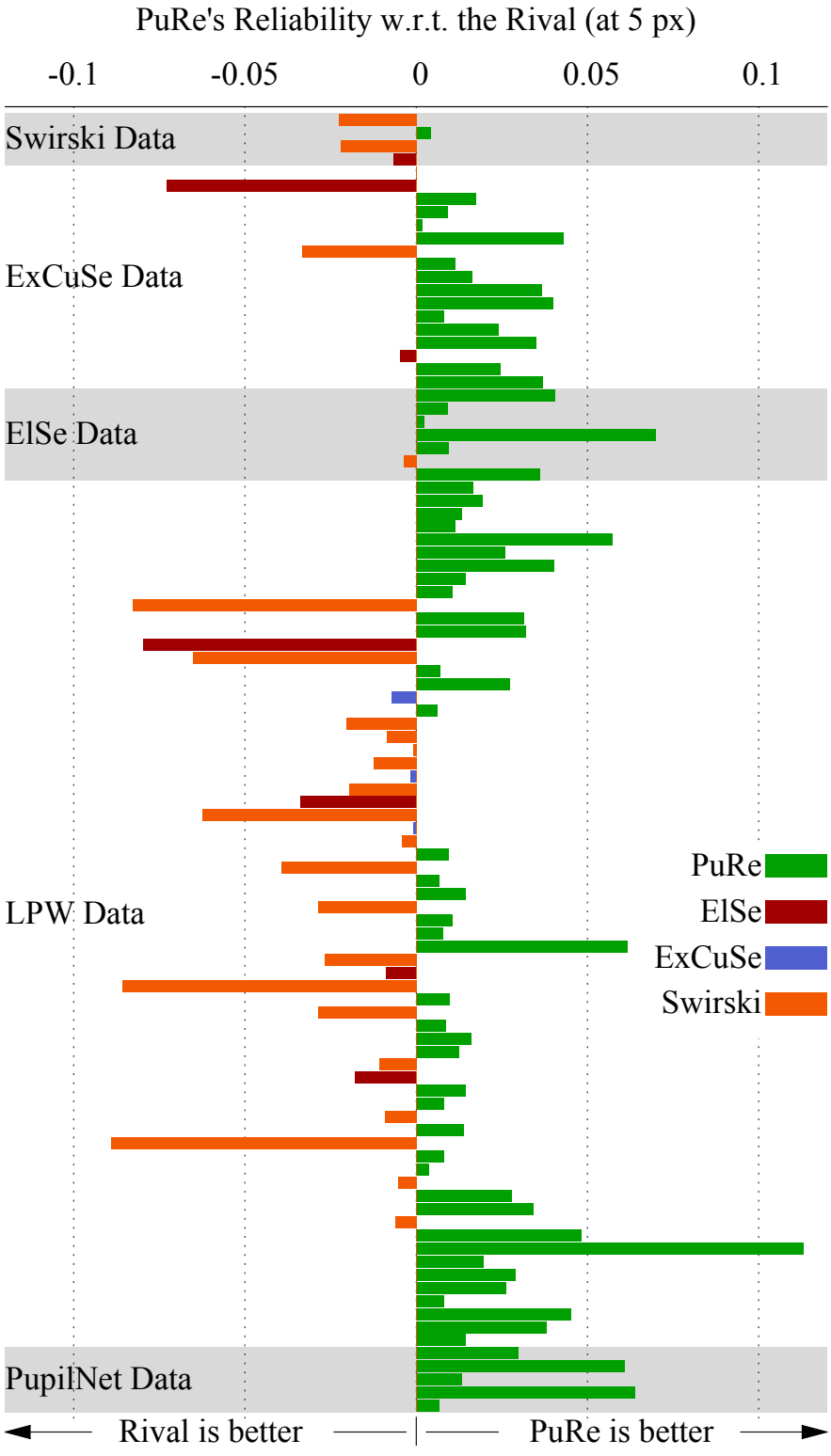


Figure 3.16: PuRe's *reliability* relative to the rival for each *use case* [6].

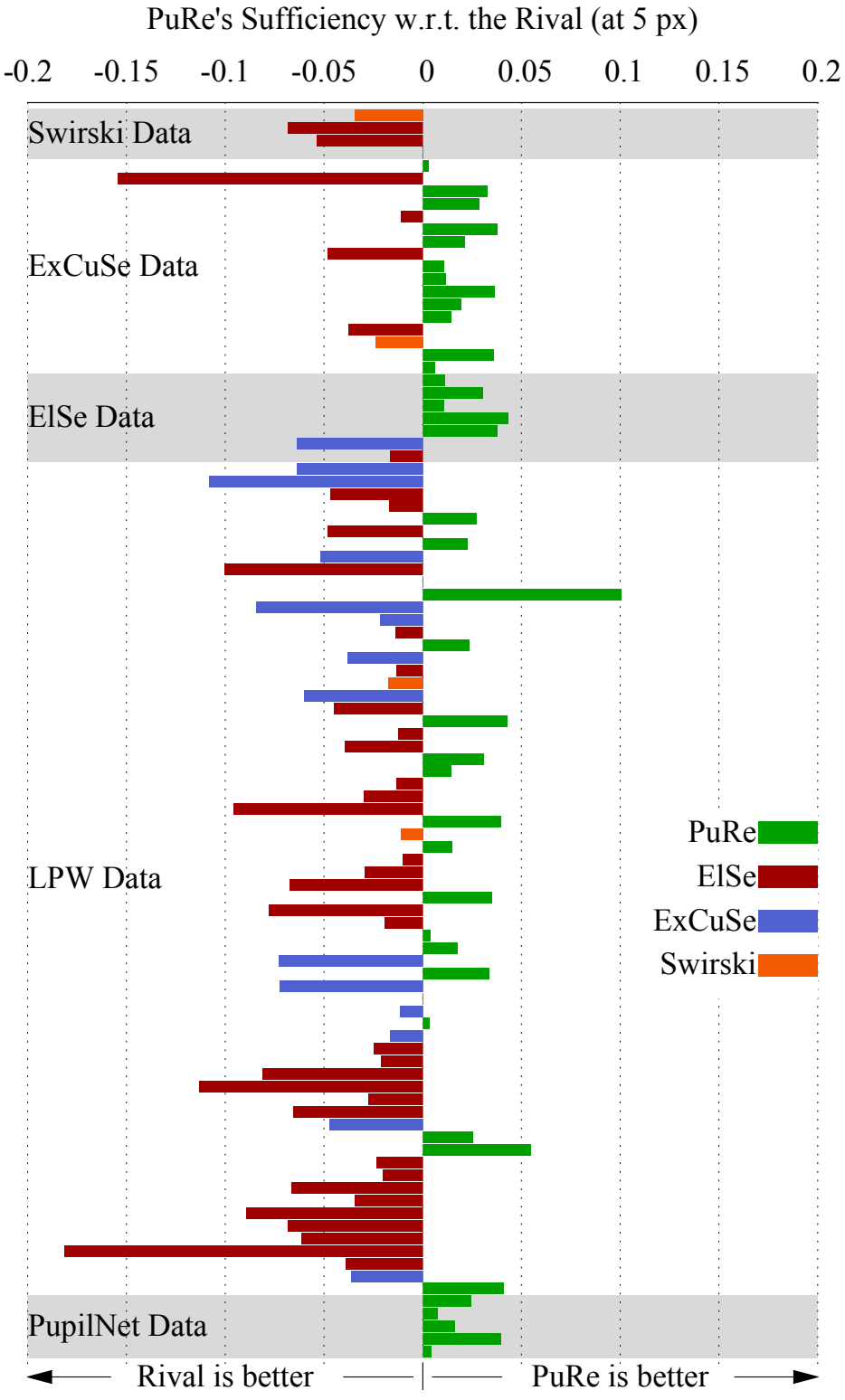


Figure 3.17: PuRe's *sufficiency* relative to the rival for each *use case* [6].

3.1.7 Run Time

The run time of pupil detection algorithms is of particular importance for real-time usage – e.g., for human-computer interaction. In this section, we evaluate the temporal performance of the algorithms across all images from the Świrski, ExCuSe, ElSe, LPW, PupilNet, and Closed-Eyes data sets. Evaluation was performed on a Intel® Core™ i5-4590 CPU @ 3.30GHz with 16GB RAM under Windows 8.1, which is similar to systems employed by eye tracker vendors. Results are shown in Fig. 3.18. All algorithms exhibited competitive performance in terms of run time, conforming with the slack required for operation with state-of-the-art head-mounted eye trackers. For instance, the [124] eye tracker, which provides images at 120 Hz – i.e., a slack of ≈8.33 ms. Henceforth, we will use the notation μ for the mean value and σ for the standard deviation. Run time wise, ExCuSe was the best performer ($\mu = 2.51$, $\sigma = 1.11$), followed by Świrski ($\mu = 3.77$, $\sigma = 1.77$), PuRe ($\mu = 5.56$, $\sigma = 0.6$), and ElSe ($\mu = 6.59$, $\sigma = 0.79$). It is worth noticing that ElSe operates on slightly larger images (346×260 px) w.r.t. PuRe and ExCuSe (320×240 px). Furthermore, Świrski operates on the original image sizes, but its implementation is parallelized using *Intel Thread Building Blocks* [163], whereas the other algorithms were not parallelized. In contrast to the algorithmic approaches, [149] report run times for their CNN-based approach of ≈ 36 ms and ≈ 40 ms running on a *NVidia Tesla K40 GPU* and a *NVidia GTX 1060 GPU*, respectively. It is worth noticing that these run times are still more than four times larger than the slack required by modern eye trackers and almost one order of magnitude larger than the algorithmic approaches running on a CPU.

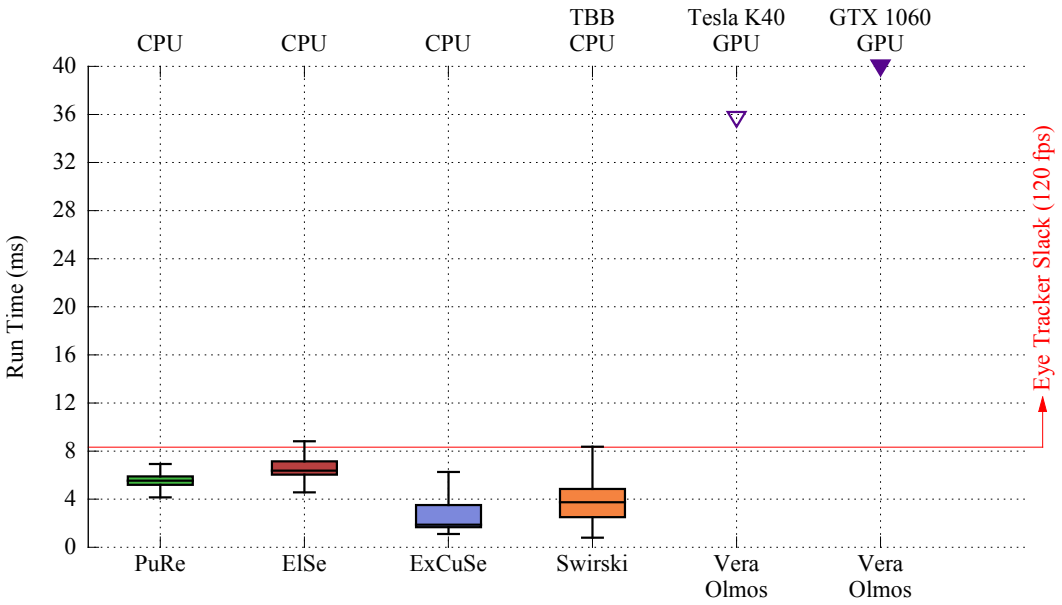


Figure 3.18: For PuRe, ElSe, ExCuSe, and Świrski: Run time distribution across all images in the Świrski, ExCuSe, ElSe, LPW, PupilNet, and Closed-Eyes data sets. Note that these algorithms were evaluated on a CPU and only Świrski was parallelized. For Vera-Olmos: Run time as reported in [149], which were obtained with parallelized implementations using GPUs [6].

3.1.8 Discussion

Evaluation results show that our single-method edge-based approach outperformed even two-method approaches (e.g., ElSe and ExCuSe). However, there are clear (but uncommon) cases when an edge-based approach will not suffice due to lack of edge-information in the image. For instance, extremely blurred images, or if a significant part of the pupil outline is occluded. These challenges might lead PuRe to 1) detect only a small part of the pupil outline, which results in a shifted pupil center and an underestimated pupil size, or 2) to fail. In general, PuRe has three failure modes, which are depicted in Fig. 3.19:

1. *Lack of edges*: when the pupil outline does not have a contrast strong enough to be detected by the Canny edge detector or is occluded by eyelids / eyelashes / reflections.

2. *Broken edges*: when the pupil outline is broken into smaller parts by eyelids / eyelashes / reflections, which end up removed by the edge segment selection stage.

3. *Deceptive candidate*: when another element in the image *resembles* a pupil more than the pupil itself (according to the definitions of the confidence metric ψ).

It is worth noticing that PuRe offers a meaningful confidence measure for the detected pupil, which can be used to identify the great majority of cases in which PuRe fails. Following from our analysis in Section 3.1.5, we recommend a threshold of 0.66 for this confidence measure. Thus, whenever PuRe can not find a pupil, an alternative pupil detection method can be employed – e.g., ElSe's fast second step. Nonetheless, care has to be taken not to compromise *specificity* through this second step.

Moreover, there are extreme cases in which pupil detection might not be feasible at all, such as when the bulk of the pupil is occluded due to inadequate eye tracker placement relative to the eye. For instance, *use cases* `LPW/5/6` and `LPW/4/1`, for which the best detection rates were measly 3.45% (by ExCuSe) and 14.15% (by Świrski), respectively. Sample images throughout these *use cases* are shown in Fig. 3.20. As can be seen in this figure, in the former not only the eye is out of focus, but there are lenses obstructing most of the pupil, whereas in the latter, the pupil is mostly occluded by the eyelid and eye lashes. In such cases, PuRe's confidence measure provides a quantitative measure of the extend to which it can detect the pupil in current conditions: By observing the ratio of confidence measures above the required threshold during a period[11]. If this ratio is too small, it can be inferred that either the pupil is not visible or PuRe can not cope with current conditions. In the former case, the user can be prompted to readjust the position of the eye tracker in real time – this is the case for `LPW/5/6` ($ratio_{th=0.66} = 0.15$) and `LPW/4/1` ($ratio_{th=0.66} = 0.54$). In both cases, the confidence measure ratio is useful for researchers to be aware that the data is not reliable and requires further processing, such as manual annotation. An example of the cases in which adjusting the eye tracker is not likely to improve detection rates is *use case* `LPW/3/16` ($ratio_{th=0.66} = 0.65$), for which reflections cover most of the image as seen in Fig. 3.20. The best detection rate for this *use case* was 31.95% (by PuRe). To

[11] The period should be significantly larger than expected blink durations since the confidence measure is also expected to drop during blinks; in this section we report the ratio for the whole *use case*.

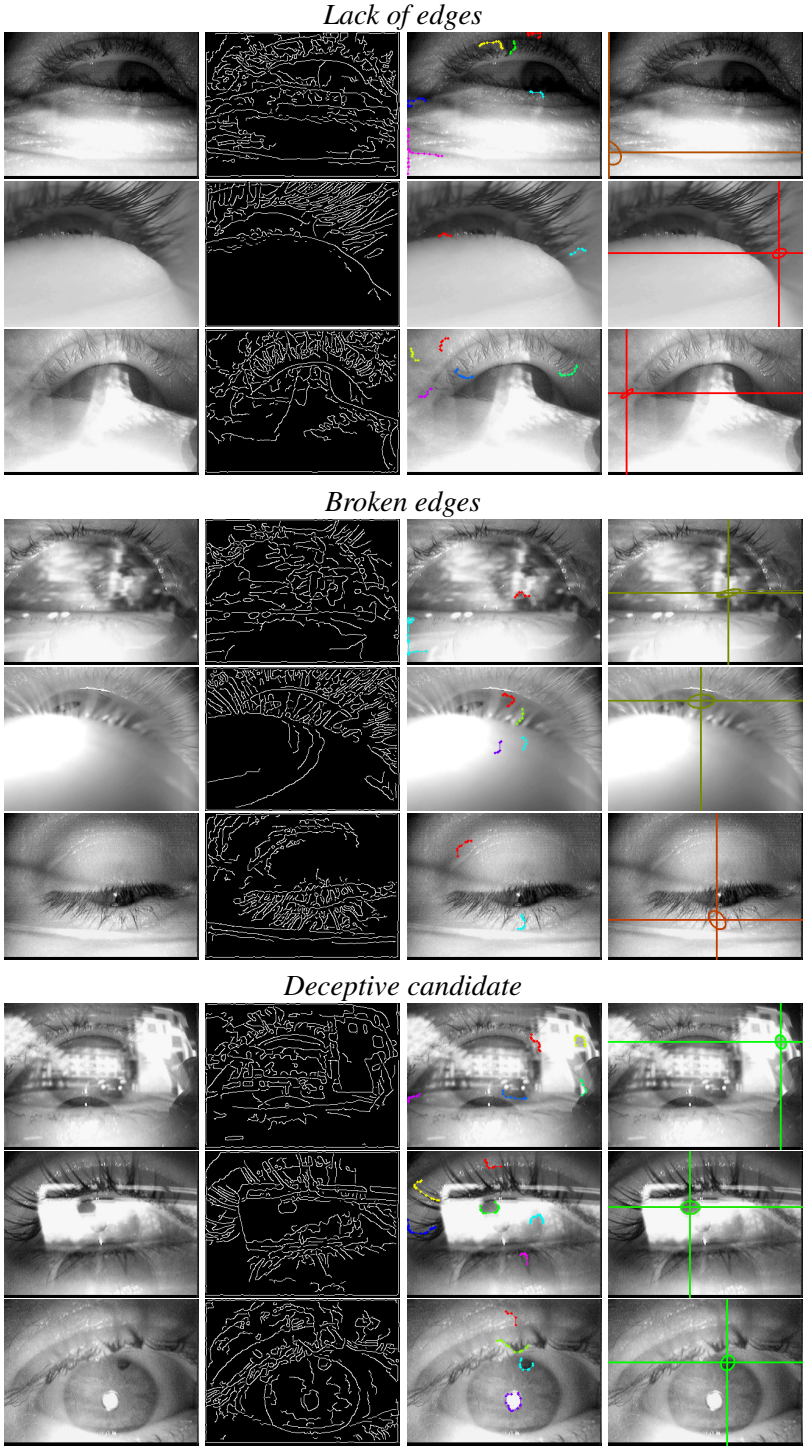


Figure 3.19: Illustrative failure cases for PuRe. First column displays the input image, whereas the second column unveils the resulting edges. The third column shows segments remaining after edge segment selection (Section 3.1.2.3) using distinct colors per segment. The last column presents the pupil returned by PuRe, encoding the confidence measure linearly in the overlay color such that red represents the lowest confidence ($\psi = 0$) and green the highest ($\psi = 1$). Notice that, except for the *deceptive candidates*, the confidence for failures cases is usually low [6].

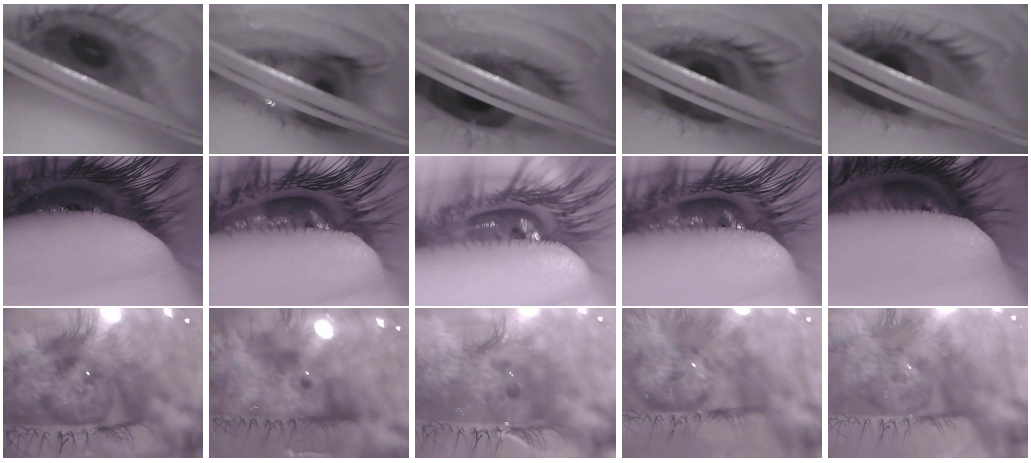


Figure 3.20: Extreme cases for pupil detection. For `LPW/5/6` (top row) and `LPW/4/1` (middle row), the eye tracker can be readjusted to improve detection rates. For `LPW/3/16` (bottom row), readjusting the eye tracker is not likely to improve the conditions for pupil detection. In all cases, researchers should be aware that the automatic pupil detection is not reliable. PuRe's confidence measure allows for users to be prompted in real time for adjustments and provides researchers with a quantitative metric for the quality of the pupil detection [6].

further support this claim, we measured the correlation between this confidence-measure ($ratio_{th=0.66}$) and the pupil detection rate, which resulted in a correlation coefficient of 0.88.

3.1.9 Conclusion

In this section, we have proposed and evaluated PuRe, a novel edge-based algorithm for pupil detection, which significantly improves on the state-of-the-art in terms of *sensitivity*, *precision*, and *specificity* by 5.96, 25.05, and 10.94 percentage points, respectively. For the most challenging data sets, detection rate was improved by more than ten percentage points. PuRe operates in real-time for modern eye trackers (at 120 *fps*) and is fully integrated into EyeRecToo (EyeRecToo) – an open-source state-of-the-art software for pervasive head-mounted eye tracking. An additional contribution was made in the form of new metrics to evaluate pupil detection algorithms and a data set containing negative samples in its majority.

3.2 Pupil Tracking

3.2.1 Related Work

As previously mentioned at the beginning of this chapter, we explicitly make a distinction between pupil *detectors* and *trackers*. PuReST is a pupil tracker, and, thus, we focus on evaluating it against approaches from the same algorithm class. Nevertheless, PuRe

was also included in this evaluation as a representative from the *detector* class and to allow a cross-comparison between the *trackers* and *detectors*. PuRe was chosen since it is the base for PuReST and also the current best performing real-time detector. For a recapitulation of PuRe, please consult Section 3.1.2. For the *tracker* class, we initially considered Starburst [164], but its detector performance was found to be particularly low (< 15%, see [1], [3], [50], [165]). Thus, instead we settled for two state-of-the-art methods with default parameters provided as part of the Pupil (v1.1) software platform [165], which is officially supported by Pupil Labs [124], to compare against, namely the Pupil Labs 2D Tracker [124] (Pupil Labs 2D) and Pupil Labs 3D Tracker [124] (Pupil Labs 3D). A brief description of the algorithms follows:

Pupil Labs 2D: This tracker uses the Pupil Labs pupil detector [124] (Pupil Labs Detector), a variant on the Świrski detector, to first locate the pupil outline. This detector uses a center-surround to estimate a coarse location for the pupil, which is used as Region of Interest (ROI) for the remaining of the algorithm. Afterwards, the lowest and highest spikes in this ROI's intensity histogram are located. Dark areas are defined using an offset from this lowest spike, and areas with intensity above the highest threshold are considered spectral reflections. Edges are detected within the ROI, and those outside of the dark areas or inside spectral reflections are discarded, resulting in *selected edges*. These selected edges are extracted into contours and split using a curvature continuity criteria. Candidate pupil ellipses are found using ellipse fitting, and the final ellipse fit is found through an augmented combinational search [165]. The tracker stage is integrated after the *selected edges* step and tracks the pupil outline by considering edges that support the pupil outline based on their distance to the outline ellipse[12].

Pupil Labs 3D: This tracker builds on top of the Pupil Labs 2D, augmenting it with 3D eye model information derived similarly to the approach proposed by Swirski et. al [111]. Whenever the evidence from the Pupil Labs 2D tracker is considered weak, constraints from competing eye models are used to robustly fit the pupil [166].

3.2.2 *Pupil Reconstructor and Subsequent Tracking [7]* (PuReST)

PuReST workflow is shown in Fig. 3.21 and consists of three distinct parts orchestrated to produce a fast and robust pupil tracking algorithm. When no *reliable* pupil information from the previous frame is available, PuReST employs PuRe [6] to find a *seed* pupil estimate. Otherwise, the search space is spatially constrained to a square ROI centered at the *previous pupil*'s center. This ROI's size can be tuned to cover a precise range of eye movements if the inter-frame period and eye position w.r.t. the camera are known[13] – e.g., by modeling the pupil movement range [111]. In this work, we take a more straightforward approach by

[12] We could not find reviewed references to this tracking stage; please consult the *strong prior* part in https://github.com/pupil-labs/pupil/blob/v1.1/pupil_src/shared_modules/pupil_detectors/detect_2d.hpp#L189 for further details.

[13] Note that one can use the ROI size to trade-off covered range and run time.

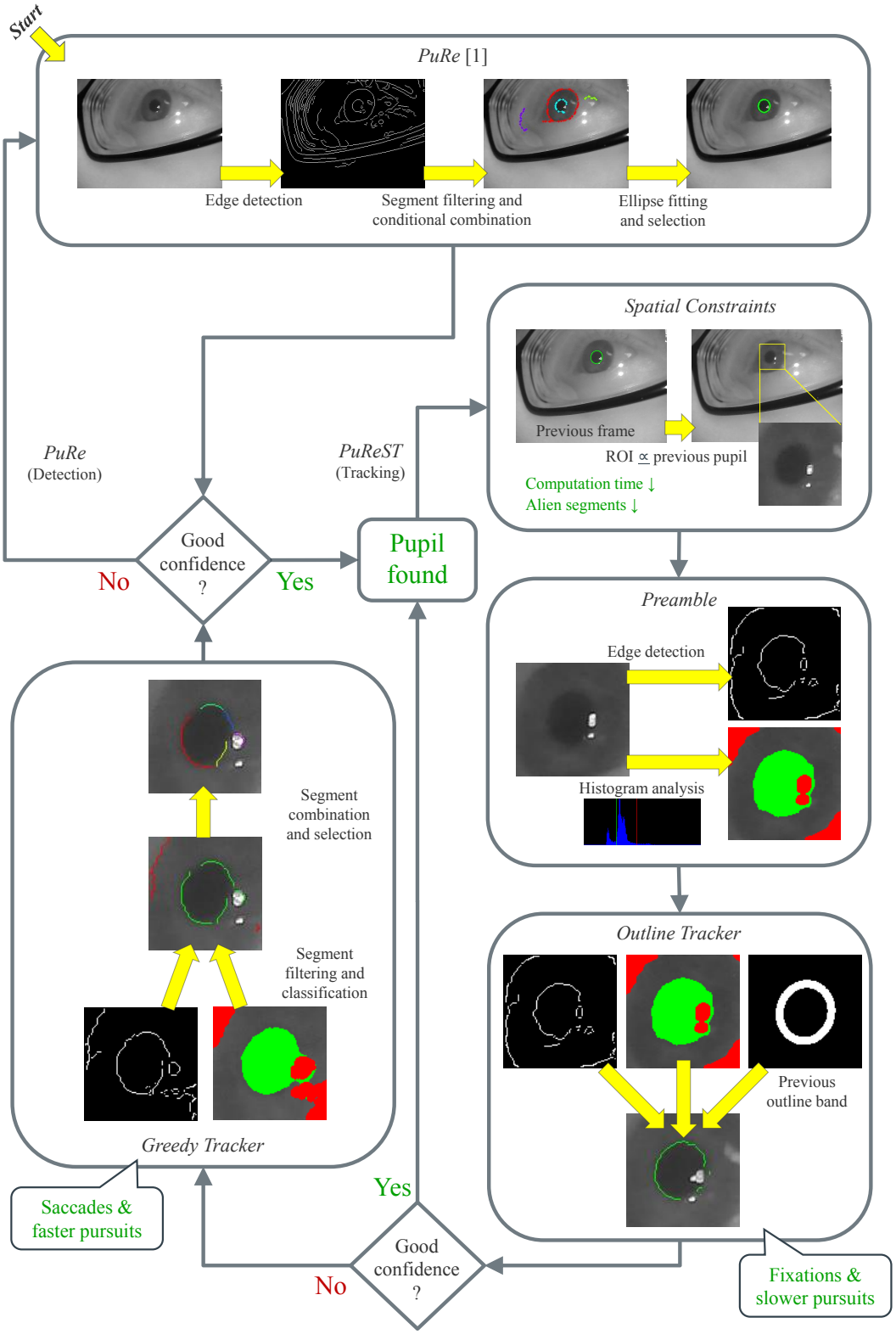


Figure 3.21: Graphical representation of the interaction between the different algorithms that compose PuReST.

using an adaptive ROI with lateral equal to twice the *previous pupil*'s major axis. Within this ROI, two methods attempt to track the *previous pupil*. The first method's (the Outline Tracker) goal is to locate the pupil during *fixations*, *slow smooth pursuits / vestibulo-ocular reflexes*, and *micro saccades* by evaluating the alignment between the *previous pupil*'s outline and the edges lying in a small band around this outline. The second method's (the Greddy Tracker) goal is to *greedily* combine *good* edge segments to reconstruct and detect the pupil when it has moved from the previous location – e.g., during *saccades* and *smooth pursuits*. These methods are described in detail in the sequence.

3.2.2.1 Initial Pupil Detection

As previously mentioned, we employ PuRe [6] to perform an initial pupil detection. PuRe is an edge-based pupil detection method that approaches the task by 1) selecting curved edge segments that are likely to belong to the pupil outline, 2) conditionally combining segments pair-wise to construct further candidates, 3) fitting an ellipse to the candidates, and 4) producing a confidence measure for each candidate based on the ellipse aspect ratio, angular edge spread w.r.t. the ellipse, and ratio of outline points whose inner-to-outer contrast supports the hypothesis of the ellipse being a pupil. Naturally, the candidate with the highest confidence is taken as pupil estimate. In [6], Santini et. al suggest a *cut-off threshold* ($\tau_{confidence}$) of 0.66 for this confidence metric, which we have adopted. Therefore, whenever a pupil with enough confidence is detected, the subsequent frame pupil detection is performed with the tracking methods.

3.2.2.2 Shared Tracking Preamble

Since both the Outline Tracker (Outline Tracker) and the Greddy Tracker (Greddy Tracker) operate on edge images using information from histogram analysis, PuReST starts with a shared preamble that generates all data common to both methods. The first step of this preamble is to downscale the ROI image if necessary. We have empirically chosen a maximum working size of 100×100 px [14]. It is worth noticing that when downscaling happens, it has a denoising effect, and the results must be upscaled to the original size, introducing an intrinsic error. Afterwards, edges are extracted using a Canny edge operator. The resulting edge image is then manipulated with morphological operations to thin and straighten edges as well as break up orthogonal connections following the procedure described by Fuhl et. al [1]. Additionally, the histogram analysis establishes two masks: *dark* and *bright*. The *dark* mask assumes that the pupil is the darkest region in the ROI and tries to estimate this region. The threshold for this mask is found iteratively by accumulating the histogram counts – starting at the darkest value – until the accumulated pixel count is larger than the *previous pupil* area. The resulting binary image is then morphologically *closed* to lessen

[14] This scale was chosen based on run time measurements from three mobile ultra low-power CPUs (Intel® Core™ i7-4510U, i7-4600U, and i5-6300U). Such processors power devices that might be effortlessly carried around but are powerful enough to run a fully fledged eye tracking framework at high frame rates ($\approx$ 120 Hz) – e.g., EyeRecToo and Pupil Labs Capture [165]), making them excellent candidates for pervasive real-time eye tracking platforms.

spurious regions that might result from small reflections or eyelashes. The *bright* mask aims at identifying glints and small reflections. A brightness threshold is first selected by accumulating histogram counts – starting at the brightest value – until the accumulated pixel count is larger than 5% of the ROI area. To cover the outskirts of corneal reflections, the resulting threshold is further decreased by a bias of 5, and the resulting binary image is morphologically *dilated*. In our implementation, the aforementioned morphological operations employ an elliptical 7×7 kernel.

3.2.2.3 Outline Tracker

First, edges outside of the *dark* mask and edges inside the *bright* mask are removed to lessen the influence from edges that belong to reflections and near-pupil iris features. The resulting edge image is then intersected with a mask of the *previous pupil*'s outline (width=5 px), and an ellipse is fit[15] to the remaining pixels. Afterwards, the *alignment ratio* between the intersected edges and the *previous pupil*'s outline circumference is calculated, akin to the metric proposed by Prasad and Leung [167][16]. If the ratio is below a minimum alignment threshold ($\tau_{align} = 0.65$), the Outline Tracker gives up. Otherwise, the edge pixel intersection and *alignment ratio* procedures are repeated using a new ellipse fit to the initially intersected edges. The edges resulting from this second iteration are then used to fit a final pupil outline candidate, and a confidence measure is established following PuRe's method.

Note that this procedure might be dangerous: In its ingenuity, it consumes any edges that lie within the pupil outline enclosing band assuming these edges to belong to the pupil outline. Hence, the reasoning for eliminating edges belonging to reflections. Nonetheless, spurious edges from the eyelids and eyelashes might still remain and cause the outline tracker to attach to these edges. To prevent such behavior, the Outline Tracker keeps track of the initial pupil seed throughout consecutive outline trackings and breaks the tracking if the major axis of the estimated pupil is larger than 1.05 times the seed pupil's major axis. Consequently, the Outline Tracker does not track continuously dilating pupils by design.

3.2.2.4 Greddy Tracker

The Greddy Tracker starts by clustering the edge pixels into connected segments using the topological structural analysis proposed by Suzuki et. al [168]. To remove significantly plain shapes, the segments are approximated with polylines using the Douglas-Pecker algorithm [169] ($\varepsilon = 1.5$), and approximations with three or less points are discarded. Afterwards, *good* segments are identified. In this context, we define *good* candidates as those segments whose majority of pixels fall within the *dark* mask. The following step gives the Greddy Tracker its name: this tracker *greedily* generates all possible segment combinations *without repetition*. Therefore, for N initial segments, the number of candidates evaluated by

[15] Throughout this work, we employed the least-squares ellipse fitting method proposed by Fitzgibbon and Fisher [154]

[16] The alignment ratio measures the ratio between edges and ellipse circumference. We saturate this metric at value one since it might result larger than that due to discretization and the nature of the edge operator.

the Greddy Tracker is $\sum_{i=1}^{N} \binom{N}{i}$. In practice, however, the number of combinations quickly becomes unfeasible to process in a timely manner. For this reason, we sort the candidates according to their diameter, which is evaluated as the largest distance between two of the segment's points, and use only the largest five segments as seeds for the combination process. Subsequently, the convex hull [170] of each candidate is calculated so that straight lines within curved segments are simplified, and an ellipse is fit to the hull points. A confidence for the candidate is calculated using PuRe's method. In this process, there are four candidate discarding mechanisms: 1) if the ellipse fit is not possible, 2) if the ellipse major axis is smaller than PuRe's minimum pupil size (pd_{min}), 3) if the ellipse aspect ratio is smaller than PuRe's ratio threshold (R_{th}), and 4) if the pupil confidence is smaller than the confidence cut-off threshold ($\tau_{confidence}$). If the Greddy Tracker finds a pupil, the new estimate is considered a new pupil seed. Otherwise, PuReST falls back to detecting using PuRe.

3.2.3 Experimental Evaluation

For this evaluation, we employed five data sets acquired with three distinct head-mounted eye tracking devices, namely, the Świrski [142], ExCuSe [3], ElSe [1], LPW [4], and PupilNet [65] data sets. In total, these data sets contain 266,786 realistic and challenging images, encompassing 99 distinct *use cases* – i.e., 99 individual eye videos.

3.2.3.1 Pupil Detection Rate

A pupil is considered detected if the algorithm's pupil center estimate lies within a radius of n pixels from the ground-truth pupil center. Similar to previous work, we use $n = 5$ to account for small deviations in the ground-truth annotation process [1], [3], [4], [6], [149]. As can be seen on the left side of Fig. 3.22, PuReST surpassed other algorithms for all pixel errors, improving the detection rate at the 5 px mark by 5.44 and 29.92 percentage points w.r.t. PuRe and the Pupil Labs algorithms, respectively. The right side of this figure indicates the generality of PuReST, showing that the proposed method reaches detection rates above 87.62% for the majority of the individual uses cases. Furthermore, Fig. 3.23 contrasts PuReST with the *rival* (i.e., the best performer from the other algorithms) in a use case granularity level. At this level, PuReST outperformed other approaches in 81.82% of use cases, where as PuRe, Pupil Labs 3D, and Pupil Labs 2D were the best performers in 13.13%, 3.03%, and 2.02%, respectively.

3.2.3.2 Run Time

A key requirement for real-time gaze-based applications, such as human-computer interaction, is the algorithm run time. In particular, as faster cameras become more accessible, the challenge of meeting the imposed processing deadline grows. For instance, Pupil Labs recently released a new camera capable of 200 fps, which corresponds to a slack of 5 ms. This challenge is further increased by the fact that system resources must be shared to process (and sometimes record) video streams from multiple cameras – e.g., two eye cameras

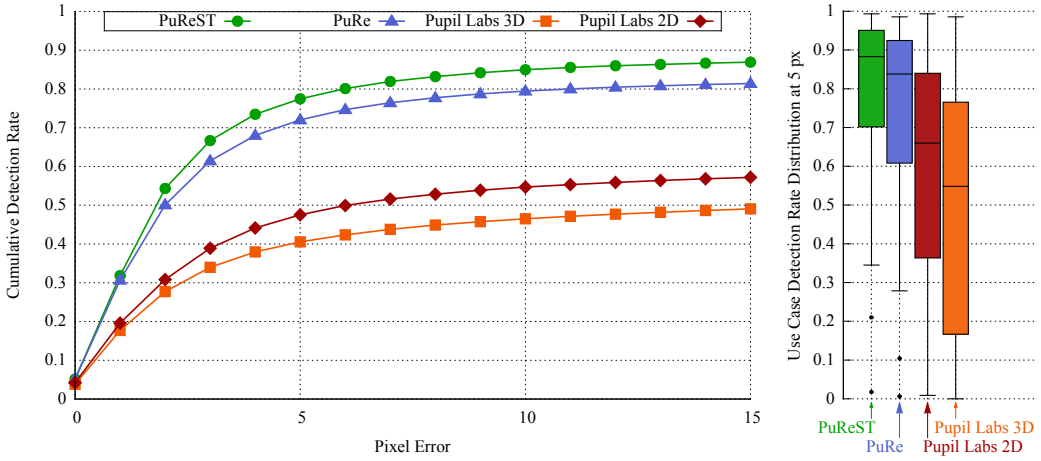


Figure 3.22: On the left, the cumulative detection rate for the aggregated 266,786 images from all data sets. On the right, the distribution of the detection rate per *use case* as a *Tukey boxplot* [155] [7].

and a field camera. We evaluated the algorithms using a Intel® Core™ i5-4590 CPU @ 3.30GHz with 16GB RAM under Windows 8.1, which is similar to systems employed by eye tracker vendors. All algorithms are coded in C++. Fig. 3.24 shows the resulting run time distribution for the evaluated algorithms. PuReST exhibited the fastest average run time (1.89 ms), reducing the average run time by a factor of 2.74 w.r.t. PuRe (5.17 ms). The Pupil Labs trackers have a very competitive average run time (Pupil Labs 2D≈2.08 ms and Pupil Labs 3D≈2.47 ms). Relative to these algorithms, PuReST's run time reduction factor is limited to only ≈ 1.1. It is worth noticing that PuRe already struggles and does not meet the required slack, but PuReST is the fastest algorithm regardless of using PuRe, indicating that the trackers are responsible for the majority of the detections. In fact, the Outline Tracker, Greddy Tracker, and PuRe were responsible for 72.55%, 22.47%, and 4.98% of the correctly detected pupils, respectively. In a first glance, this distribution misleadingly indicates that the Outline Tracker is the main PuReST driver. However, this distribution is skewed due to the LPW data set, which was collected using a slowly moving object, resulting in an overwhelming majority of slow smooth pursuits. When excluding this data set, the distribution is more evenly spread with the Outline Tracker, Greddy Tracker, and PuRe contributing 49.32%, 40.81%, and 9.87%, respectively. It is worth noticing the upper bound for run time results for cases in which both trackers fail, and PuRe must be run; if no pupil is detect to be tracked, run time results similar to PuRe.

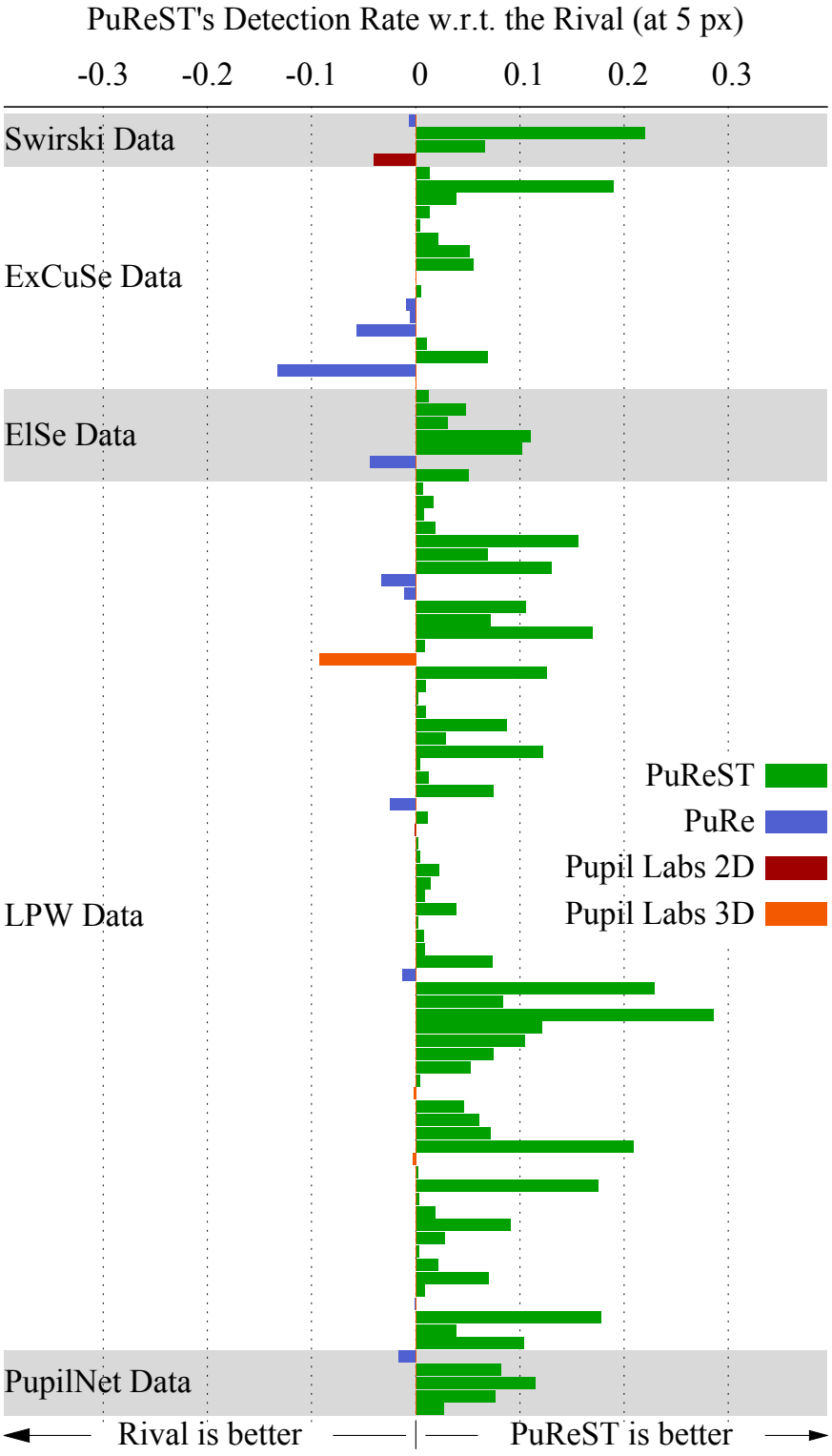


Figure 3.23: PuReST w.r.t. to the rival; each line within a data set represents a distinct *use case*. PuReST is the best algorithm in 81.82% of cases, PuRe in 13.13%, Pupil Labs 2D in 3.03%, and Pupil Labs 2D in 2.02% [6].

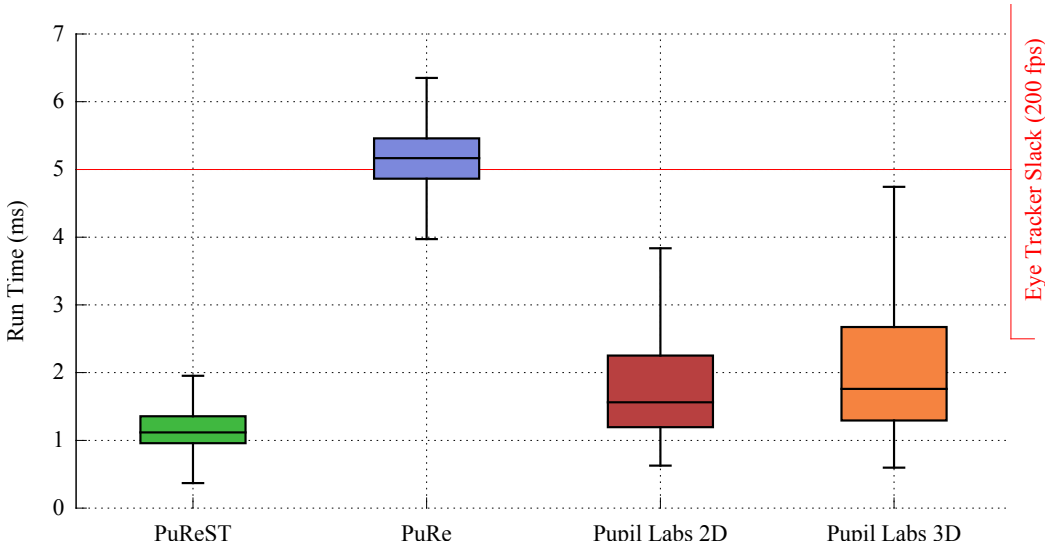


Figure 3.24: Run time distribution across all evaluation images using a Tukey schematic boxplot; outliers are not shown for the sake of visualization. PuReST has an average run time of $\mu = 1.88$ ms and standard deviation $\sigma = 2.19$ ms, whereas the others resulted: PuRe ($\mu = 5.17$ ms,$\sigma = 0.51$ ms), Pupil Labs 2D ($\mu = 2.08$ ms,$\sigma = 1.65$ ms), and Pupil Labs 3D ($\mu = 2.47$ ms,$\sigma = 5.14$ ms) [6].

3.2.4 Conclusion

In this section, we have proposed and evaluated PuReST, a novel algorithm for fast and robust pupil tracking. The proposed method was evaluated on over 266,000 realistic and challenging images acquired with three distinct head-mounted eye tracking devices, increasing pupil detection rate by 5.44 percentage points while reducing average run time by a factor of 2.74 when compared to state-of-the-art pupil detectors. Relative to state-of-the-art pupil trackers provide by a vendor, PuReST increases detection rate by 29.92 percentage points while reducing average run time by a factor of 1.1. Overall, PuReST outperformed other methods in 81.82% of use cases. While this is our first iteration of PuReST, which uses prior knowledge only from a single past frame, this initial evaluation resulted in a significant improvement in term of both detection rate and run time. However, the *tracking* design space offers a wide range of possibilities, and a significant amount of work remains for future work. In particular, we find the approach of the Pupil Labs 3D tracker to point in an interesting direction by attempting to derive and employ a 3D eye model to improve pupil detection. Furthermore, the applicability and required modifications to more generic trackers should also be investigated such as Kernelized Correlation Filters [171], [172] and Tracking-Learning-Detection [173]. Finally, we hypothesize that the tracking can generate a more *stable* gaze estimation signal given that prior information is taken into account, which we plan to evaluate in the near future.

3.3 Influence on Gaze Signal

In the context of head-mounted eye trackers, robust pupil tracking has been the subject of considerable research in the past decade as it serves as a foundation for virtually every competitive eye-tracking framework. For instance, there are several works a) proposing new algorithmic approaches [1]–[3], [6], [7], [142], [164], [165], [174]–[177], b) proposing machine-learning approaches [58], [64], [65], [149], [178]–[182], and c) evaluations of existing methods and dataset contributions [4], [50], [62], [183]. In common, most of this research has concentrated on evaluating pupil tracking algorithms in terms of runtime and detection rates – i.e., the capability of an algorithm to correctly estimate the pupil center w.r.t. an annotated ground-truth up to a certain threshold distance (typically 5 px). The exceptions here are the works from Świrski et al. [142], which performs the evaluation based on the Hausdorff distance [184], Perez et al. [183], which also considers image size, and Santini et al. [6], which additionally proposes a series of metrics for a more comprehensive pupil tracking evaluation, including *reliability*, *sufficiency*, as well as a confidence metric in combination with an extended binary classification metric to evaluate true/false positives/negatives rates.

These works, however, consider the pupil tracking as an isolated element, despite the fact that its deterioration can significantly degrade gaze-estimation calibration [24], automatic eye movement detection [156], glanced-area ratio estimations [157], eye model construction [111], human-computer interaction [185], and even lead to wrong medical diagnosis [158]. For instance, there is no consideration regarding the resulting gaze error magnitude stemming from the typical 5 px tolerance, which can be expected to be quite significant depending on camera resolution and eye distance. In this work, we take an initial step towards evaluating pupil tracking algorithms as a part of a larger head-mounted eye-tracking system, investigating its influence directly on calibration and gaze estimation. Our main contributions are:

- Demonstrating that pupil tracking algorithms can easily be augmented with a generic and meaningful confidence metric with negligible overhead.
- An adaptive quadvariate binocular polynomial regression gaze estimation method that employs the aforementioned confidence metric to automatically switch between binocular / monocular mode, significantly improving gaze estimation results for all evaluated metrics.
- Showing that higher detection rate might not directly translate into higher accuracy and precision.
- Evidencing that whereas there is little difference in terms of accuracy and availability between the investigated pupil detection-by-tracking and tracking-by-detection methods, detection-by-tracking is significantly more precise.
- We provide the eye-tracking data collected during our study as an open data set to foster further research at:
  `www.ti.uni-tuebingen.de/perception`

3.3.1 Experiment Design

To measure the influence of distinct pupil tracking algorithms on the gaze signal, we designed an experiment to collect eye-tracking data from participants while they gazed at a target displayed in 49 homogeneously distributed positions of a display unit (center-to-center distances of approximately 5.4 cm horizontally and 4 cm vertically). These stimuli covered $\approx 30^\circ$ and $\approx 22^\circ$ of the horizontal and vertical visual field, respectively, covering the vast majority of spatial gaze distribution during unconstrained-head visual behavior [137]. This data was then *post-hoc* processed to derive pupil center estimations for distinct state-of-the-art pupil tracking algorithms. *Nine* of the positions were used to calibrate the system, and the remaining *forty* positions were used for evaluation.

3.3.1.1 Participants

Experiments were managed by an expert with more than three years of experience in conducting mobile and stationary eye tracking experiments. In total, sixteen (13 males, 3 females) healthy adult (Ages: $\mu = 28.31, \sigma = 3.9$) subjects participated in the experiment. One participant was excluded due to nystagmus occurrences. Moreover, six of the participants use eye glasses in their everyday lives; for these participants, we collected data both *with* and *without* glasses on separate trials. We treat these additional trials as independent observations since the task changes both in terms of pupil tracking and gaze estimation (see Fig. 3.25), thus making our sample size $(16 - 1 + 6)$ twenty one.

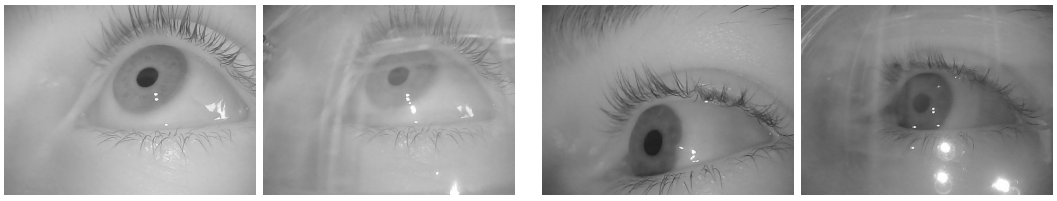


Figure 3.25: Examples of two participants with and without glasses gazing at the same position. Glasses completely change the pupil tracking task difficulty, and the eye cameras must be shifted to accommodate the glasses, resulting in distinct camera perspectives as well as changes in the eye tracker geometry.

3.3.1.2 Apparatus and Software

The experiment was conducted using a binocular Pupil eye tracker [124], with eye cameras capturing *480p* images at 120 Hz and field camera capturing *720p* images at 30 Hz. EyeRecToo was used to drive the eye tracker and record the data; the evaluation platform was also derived from this software since it provides modular implementations of multiple state-of-the-art pupil tracking methods as well as gaze estimation methods. In total, we investigated all four readily available methods in EyeRecToo: one *detection-by-tracking* and three *tracking-by-detection* methods, namely PuReST, PuRe, ElSe, and ExCuSe. We

followed the same parameter configurations as [6], [7]. Participants sat in front of a *Samsung SyncMaster 2443BW*[17] color display unit. The gaze was transformed from the eye tracker to the display coordinate frame through an homography using a set of automatically detected ArUco [186] markers digitally displayed on the lateral areas of the display, introducing an estimated error smaller than 0.35° – inline with reported accuracies in the literature [187]. During the data collection, a chin rest was used to hold participants in place at ≈ 65 cm from the display; at this distance, one degree of visual angle corresponds to ≈ 1.1 cm on the display. At a first glance, it might seem counterintuitive to use a chin-rest to investigate gaze estimation using a head-mounted eye tracker. However, a chin rest *does not impact head-mounted eye tracker functionality*, only altering participant behavior. Thus, this allows us to control for several variables during the experiment and focus on the effects and interactions between the selected gaze estimation method and the distinct investigated pupil tracking algorithms. For instance, if participants look through the targets by moving their head instead of their eyes, from the head-mounted eye tracker perspective, we are measuring a single thing: How well the eye tracker estimates gaze w.r.t. a single 3D position in front of the eye tracker. Thus, by fixating user head position, we actually *maximize* the amount of eye poses measured, consequently better characterizing the whole range of device operation.

3.3.1.3 Task

During the task, participants were instructed to gaze at the center of a CalibMe collection marker augmented with a small red dot (encompassing about ≈ 0.5° of visual angle) and press any key to start data collection for that position. After 1.5 s, an audible feedback signaled to the participant that data collection was finished, and the marker automatically moved to the next position. During this period, approximately 180 samples per position were collected. For the remainder of this Section, we will refer to each set of samples per stimuli position as a *fixation* although no automatic eye movement detection (e.g., [66], [188]) was employed. In total, 1029 distinct fixations were collected (49 per trial). Despite participants being instructed to avoid blinking between the key press and the audible feedback, we identified 131 out of the 1029 (≈ 12.73%) fixations across all subjects due to blinks nonetheless[18]. We performed this identification manually to avoid possible biasing due to false positives from automated methods such as [61]. These blinks include 117 of the 840 evaluation fixations (189 were used for calibration, i.e., 9 per trial), whose associated gaze estimations were excluded from the overall evaluation.

3.3.1.4 Metrics

We discuss our experiment's results in terms of *accuracy*, *precision*, and *availability*. The *per-sample angular offset* was estimated based on a) the Euclidean distance between the ex-

[17] Width: 520 mm. Height: 320 mm. Resolution: 1920x1200 pixels. Screen refresh rate: 60 Hz. Luminance: 0.08 cd/m^2.

[18] ≈ 61.1% of the identified fixations were from a single participant that blinked on 42 and 38 fixations of the trials with and without glasses, respectively.

pected target position and gaze in display coordinates, and b) participant-to-display distance. For a given considered period (e.g., a fixation), *accuracy* and *precision* were evaluated as its samples' angular offset mean and standard deviation, respectively, following [189].

We measure *availability* as the ratio of valid fixations, where a fixation is considered valid if and only if more than a threshold percentile $th_{percentile}$ of its samples are valid. A sample is considered valid if it has a confidence level larger than a certain confidence threshold $th_{confidence}$ and angular offset smaller than th_{angle} – i.e., *true positives* reported by the pupil trackers. Regarding the pupil detection confidence, we employed one of the measures from PuRe as generic confidence metric and, thus, follow the recommendations of the original paper: $th_{confidence} = 0.66$ [6]. As for the angular offset threshold, we set $th_{angle} = 5^{\circ}$ considering typically reported angular offsets for head-mounted eye trackers – e.g., [53], [97], [131], [190]–[192]. The intuition behind this proposed metric is to measure signal availability during fixations, for instance, for human-computer interaction.

3.3.2 Analysis and Results

3.3.2.1 Pupil Confidence Augmentation

The original ElSe and ExCuSe implementations do not offer reliable ways of identifying false pupil detections (although ElSe does offer an unreliable validity threshold parameter) [6]. This leads to a critical nuance when employing their estimates for binocular gaze estimation. Suppose that, from the two eyes, one has a successful pupil detection, whereas the other results in a false positive. Intuitively, if the false positive can be identified as such, it should still be possible to reasonably estimate the gaze based on the valid pupil alone; on the other hand, if both pupils are used, the gaze signal becomes *corrupted*. To investigate this concept, we have evaluated ElSe and ExCuSe in two configurations: a) their *original* implementations, in which the confidence of any detected[19] pupil is always 1 and b) augmented versions that use PuRe's *outline contrast ratio* [6] as a measure of confidence, which we shall henceforth refer to as Confidence Augmented Ellipse Selector (ElSe+) and Confidence Augmented Exclusive Curve Selector (ExCuSe+). We have chosen this confidence method due to its genericity and low overhead, requiring only an estimate of the ellipse outline and the input image. From these inputs, the outline contrast is measured by investigating predefined and equally-strided points laying on the pupil outline, and evaluating the ratio of points in which the inner intensity is darker than the outer intensity; for a more detailed elucidation, we refer the reader to the original paper. On average, one confidence evaluation took $\approx 11.92\,\mu s$ ($\sigma \approx 2.82\,\mu s$), negligible w.r.t. the algorithms' runtime, which is in the order of ms. Furthermore, we developed an additional adaptive gaze estimation[20] method into EyeRecToo that takes advantage of the confidence metric by combining a) a quadvariate polynomial regression if both detected pupils are considered valid and b) a monocular bivariate polynomial regression if only one pupil is considered valid. A pupil confidence validity cut-off threshold of 0.66 was employed following [6]'s recommendations. This method uses the pupil centers to estimate the horizontal and vertical

[19] If a pupil does not pass ElSe's validity threshold, it is simply considered undetected.
[20] Available at www.ti.uni-tuebingen.de/perception

gaze components independently, and the polynomials' coefficients are determined using single value decomposition during calibration. All gaze estimates in the following sections were obtained using this method.

3.3.3 Accuracy and Precision

For each pupil detection algorithm, we have calculated gaze accuracy and precision for *all non-blink evaluation fixations for all participants*, totaling 723 fixations and 123279 samples. The distributions of these aggregations are shown in Fig. 3.26 and Fig. 3.27. Upon visual inspection, a massive difference between the original ElSe and ExCuSe algorithms and their augmented versions (ElSe+ and ExCuSe+) is clearly visible in terms of both accuracy and precision. These results highlight a) the efficacy of the contributed adaptive gaze estimation method implementation, b) the importance of having an associated confidence metric to the pupil tracking algorithm, and c) how pupil tracking algorithms can easily and effectively be augmented with a generic confidence metric. It is worth noticing here that despite the considerable difference in terms of gaze estimation accuracy and precision, the original and augmented versions of ElSe and ExCuSe retain the same *pupil detection rate*, evidencing the need to evaluate pupil tracking algorithms in the full context of eye tracking in opposite to the hitherto approach of focusing solely on detection rates. In fact, ExCuSe+ outperformed ElSe in terms of accuracy and precision despite the former having a lower detection rate than the latter.

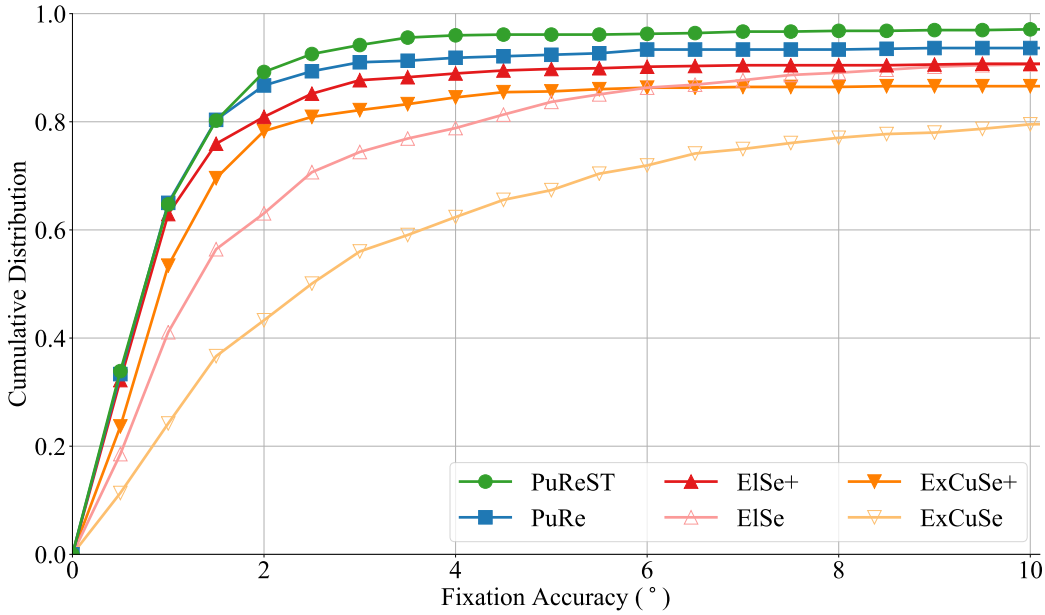


Figure 3.26: Empirical cumulative distribution function for fixation accuracy at different angular offset levels. The distribution is cut off at 10° to improve visualization. The closer to the top left the data point, the better.

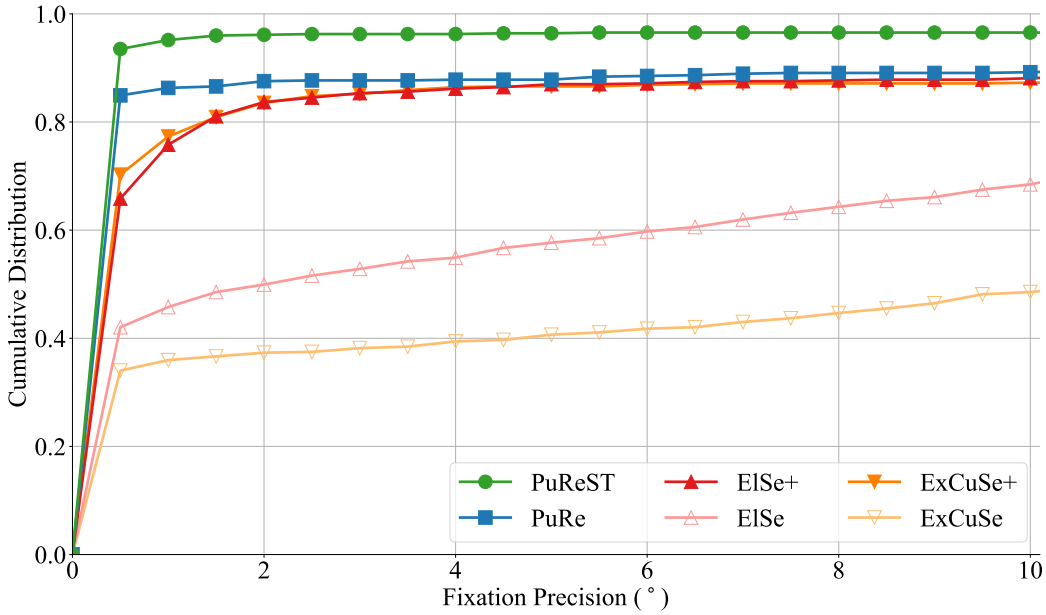


Figure 3.27: Empirical cumulative distribution function for fixation precision at different angular offset levels. The distribution is cut off at 10° to improve visualization. The closer to the top left the data point, the better.

To investigate the performance differences between all the evaluated algorithms more precisely, we opted to conduct *pairwise independent-samples Welch's t-Tests* [193]. Considering all samples, significant differences are easily spotted, as large outlier angular offsets can significantly affect the distribution parameters. In fact, in this case we only *did not* find significant differences ($p < .5$) for the following pairs: Accuracy = { (PuRe,ElSe+), (ElSe+,ExCuSe+), (ElSe+,ElSe), (ExCuSe+,ElSe) } and Precision = { (PuRe,ElSe+) }. Thus, we also conducted the same tests using a cut-off angular offset threshold of 5° to alleviate effects stemming from outlier samples with large angular offsets. We chose this threshold based on the fact that all algorithms with a confidence metric seem to plateau around this range. Per pairwise evaluation, we identified and removed fixation samples that had an angular offset larger than the threshold for at least one of the algorithms – i.e., each algorithm pair was compared using exactly the same set of samples. Results for these tests are reported in Table 3.1. Whereas Fig. 3.26 and Fig. 3.27 suggest that PuReST, the pupil detection-by-tracking representative, outperforms the other (pupil tracking-by-detection) algorithms in terms of both accuracy and precision, the results from Table 3.1 do not show significance in terms of accuracy against PuRe and ElSe. Nevertheless, significance was found when comparing PuReST with all other algorithms in terms of precision, supporting the hypothesis raised by [7] that *detection-by-tracking* produces a stabler signal than *tracking-by-detection*. The results of Table 3.1 also support the aforementioned differences between the original ElSe and ExCuSe algorithms and their confidence-augmented versions ElSe+ and ExCuSe+.

	Method 1	Method 2	*t*	*P*	*Significance*
Accuracy	*PuReST*	*PuRe*	-0.58	2.82e-01	ns
	PuReST	*ElSe*$^+$	-1.51	6.57e-02	ns
	PuReST	*ExCuSe*$^+$	-3.00	1.37e-03	**
	PuReST	*ElSe*	-11.17	1.78e-27	***
	PuReST	*ExCuSe*	-14.35	2.48e-41	***
	PuRe	*ElSe*$^+$	-1.14	1.27e-01	ns
	PuRe	*ExCuSe*$^+$	-2.75	3.02e-03	**
	PuRe	*ElSe*	-10.71	1.30e-25	***
	PuRe	*ExCuSe*	-15.18	3.71e-45	***
	ElSe$^+$	*ExCuSe*$^+$	-1.71	4.39e-02	ns
	ElSe$^+$	*ElSe*	-10.29	6.93e-24	***
	ElSe$^+$	*ExCuSe*	-13.33	8.78e-37	***
	ExCuSe$^+$	*ElSe*	-6.86	5.97e-12	***
	ExCuSe$^+$	*ExCuSe*	-12.76	3.19e-34	***
	ElSe	*ExCuSe*	-5.70	8.02e-09	***
Precision	*PuReST*	*PuRe*	-2.85	2.22e-03	**
	PuReST	*ElSe*$^+$	-11.60	9.25e-29	***
	PuReST	*ExCuSe*$^+$	-10.31	1.58e-23	***
	PuReST	*ElSe*	-10.01	1.26e-21	***
	PuReST	*ExCuSe*	-6.44	2.47e-10	***
	PuRe	*ElSe*$^+$	-9.71	2.40e-21	***
	PuRe	*ExCuSe*$^+$	-7.57	5.18e-14	***
	PuRe	*ElSe*	-10.18	5.47e-22	***
	PuRe	*ExCuSe*	-6.05	2.19e-09	***
	ElSe$^+$	*ExCuSe*$^+$	2.09	1.84e-02	*
	ElSe$^+$	*ElSe*	-7.15	1.61e-12	***
	ElSe$^+$	*ExCuSe*	-3.00	1.46e-03	**
	ExCuSe$^+$	*ElSe*	-6.23	4.95e-10	***
	ExCuSe$^+$	*ExCuSe*	-4.94	6.78e-07	***
	ElSe	*ExCuSe*	1.37	8.54e-02	ns

Table 3.1: Results for one-sided pairwise *Welch's t-Tests* for accuracy and precision between all evaluated algorithms. Significance follows the American Psychological Association (APA) convention: a) ns $\Rightarrow$ $p > .05$, b) * $\Rightarrow$ $p \leq .05$, c) ** $\Rightarrow$ $p \leq .01$, and d) *** $\Rightarrow$ $p \leq .001$.

3.3.4 Validity

As previously mentioned in Section 3.3.1.4, we also evaluated the algorithms in terms of *validity*. Fig. 3.28 shows how the ratio of valid fixations changes depending on application requirements (in terms of percentiles of valid fixation samples). In the context of human-computer interaction, a meaningful threshold for discussion is $th_{percentile} = 95\%$ as this is the percentile suggested by previous work for a robust and smooth gaze interaction [185]. At this level, we found only a small difference in the ratio of fixations considered robust enough for interaction between PuReST (92.7%) and PuRe (91.15%). The remaining algorithms, however, lagged far behind: ElSe+ (85.20%), ExCuSe+ (78.42%) ElSe (77.18%), and ExCuSe (66.53%).

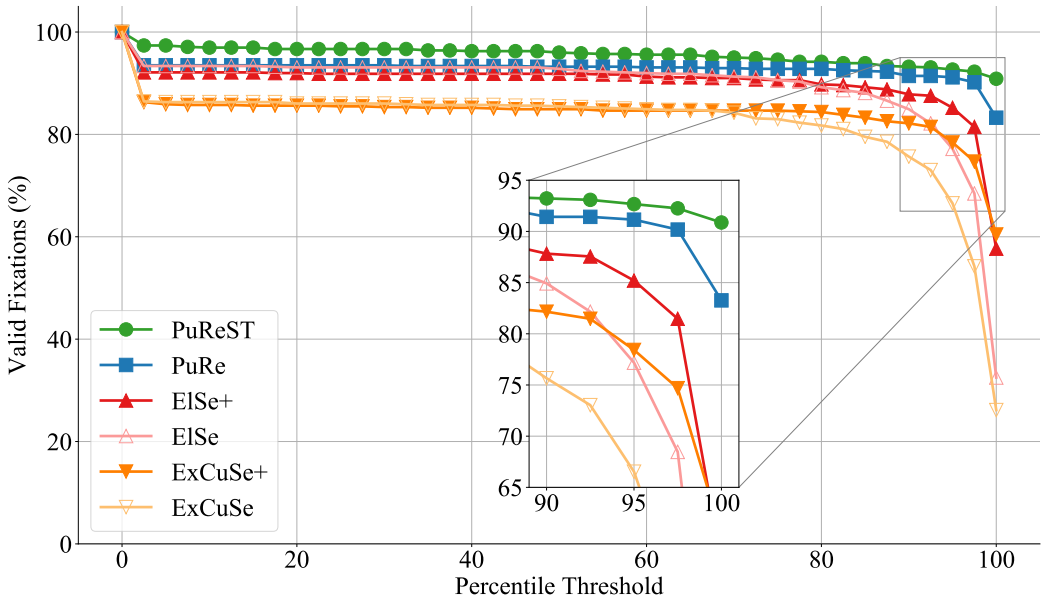


Figure 3.28: Ratio of *valid fixations* (as defined in Section 3.3.1.4) for each pupil tracking algorithm based on different percentile ($th_{percentile}$) requirements.

3.3.5 Conclusion

In this section, we have investigated the influence of four state-of-the-art pupil tracking algorithms on gaze estimation using three distinct metrics: *accuracy*, *precision*, and *availability*. We demonstrated that 1) employing a pupil confidence metric in combination with an adaptive gaze estimation method can appreciably improve binocular eye-tracking performance for all metrics, 2) regular pupil tracking methods can be easily augmented with a generic confidence metric with negligible overhead, and 3) higher detection rates might not directly translate into higher accuracy and precision – e.g., ExCuSe+ vs ElSe. Furthermore, we found supporting evidence that pupil detection-by-tracking approaches (e.g., PuReST) significantly outperforms tracking-by-detection algorithms (e.g., PuRe, ElSe, ElSe+, ExCuSe,

and ExCuSe+) in terms of precision although gains in terms of accuracy and availability are less pronounced.

Currently, we focused our investigation on the real-time algorithms available within EyeRecToo. As machine-learning methods start becoming more feasible for real-time head-mounted eye tracking, we intend to extend this analysis with a broader algorithm selection including such methods. Nevertheless, an important message remains for pupil-tracking researchers developing algorithmic or machine-learning methods alike: *Do not focus solely on pupil detection rates*. In this work, we a) demonstrated that higher detection rates do not imply better fixation accuracy and precision as well as b) provided a data set that can be used to compare pupil detection and tracking methods in this regard.

4 Calibration and Gaze Estimation

> " *A philosopher once asked, "Are we human because we gaze at the stars, or do we gaze at them because we are human?" Pointless, really..."Do the stars gaze back?" Now, that's a question.*"
>
> —Neil Gaiman

4.1 Calibration

A particular aspect of gaze estimation is the calibration step, which is used to produce a function mapping the position of the user's eyes to gaze. High-end state-of-the-art mobile eye tracker systems (e.g., SMI and Tobii glasses [194], [195]) rely on geometry-based gaze estimation approaches, which can provide gaze estimations without calibration. In practice, it is common to have at least an one point calibration to adapt the geometrical model to the user and estimate the angle between visual and optical axis. Additionally, it has been reported that additional points are generally required to achieve satisfactory accuracy [123]. Furthermore, such approaches require specialized hardware (e.g., multiple cameras and glint points), cost in the order of tens of thousands of $USD, and are susceptible to inaccuracies stemming from lens distortions [196]. On the other hand, mobile eye trackers that make use of regression-based gaze-mappings require a calibration step but automatically adapt to distortions and are comparatively low-cost (e.g., a research grade binocular eye tracker from Pupil Labs is available for $2740 EUR [124]). It is worth noticing that similar eye trackers have been demonstrated by mounting one eye and one field camera onto the frames of glasses [25], [197]–[199], yielding even cheaper alternatives for the more tech-savy users.

In its current state, the calibration step presents some disadvantages and has been pointed out as one of the main factors hindering a wider adoption of eye tracking technologies [51]. Popular calibration procedures customarily require the assistance of an individual other than the eye tracker user in order to calibrate (and check the accuracy of) the system. The user and the aide must coordinate so that the aide selects calibration points accordingly to the user's gaze. As a result, current calibration procedures cannot be performed individually and require a considerable amount of time to collect even a small amount of calibration points, impeding their usage for ubiquitous eye tracking. Henceforth we will refer to these methods as N-Points calibrations (N-Points), where N is the amount of calibration points employed.

In this section, we propose a novel approach – dubbed Calibrating with Movements (CalibMe) – that enables users to quickly and independently calibrate the eye tracker based on the movement of *collection markers*. We define *collection markers* as automatically detected markers meant to dynamically collect large arrays of relationship points between a user's eye position and gaze for both calibration and evaluation. These markers can be contrasted with *calibration markers*, which are used as reference points in a more static fashion to collect a small amount of calibration points. We employ a specific ArUco [186] marker selected based on multiple properties that make it an efficient collection marker, compared to custom markers employed as calibration markers in previous work. This allows us to hijack an existing and well established fiducial marker detection method used for augmented reality and to define Areas of Interest (AOIs) to enable CalibMe without incurring additional and costly image processing. Employing a collection marker, users are able to collect a significant amount of eye-gaze relationships for calibration and evaluation in an unsupervised fashion by moving their heads or the marker while fixating the center of the marker. We then propose rationalized outliers removal approaches to automatically eliminate ill-conditioned samples as well as a parameterizable method for the automatic selection of evaluation points. Effectively, these operations enable the users to quickly calibrate and assess gaze estimation quality without the assistance of a supervisor[1], such as in the envisioned use cases illustrated in Fig. 4.1. Moreover, the ramifications of allowing head rotation during calibration are discussed, and different collection movement patterns are proposed and evaluated. Additionally, the efficacy of CalibMe is compared to a typical 9-Points based on a regular twenty five point grid evaluation. A 9-Points calibration was selected for evaluation as it presents a reasonable trade-off between accuracy and calibration time. CalibMe is integrated into EyeRecToo, an open-source data acquisition software for head-mounted eye trackers, and, thus, readily available. We also provide a companion *Android* application that the user can use to display the *collection marker* and collect eye-gaze relationships.

4.1.1 Related Work

Whereas there is a large amount of work investigating how to improve regression-based gaze estimation, previous work has mostly focused on investigating different regression approaches (e.g., polynomial fit [100], projective transformations [110], neural networks [201]) and optimizing their parameters (e.g., polynomial order, number of hidden neurons). Little attention has been given to improving the calibration procedure, which has remained largely unmodified since its inception. In general, reference points are placed as to cover the expected range of visual movements of the subject[2]. Afterwards, a supervisor and the subject cooperate to collect calibration points.[3]. The supervisor is also responsible for checking that eye features (mainly the pupil center) are detected correctly throughout

[1] It is worth noticing that evidence suggests improved gaze estimation accuracy and precision when the participant has control over eye-gaze relationships collection [200].

[2] Natural features can also be used as reference points.

[3] The minimal amount of points is usually determined by mathematical constraints from the regression – e.g., the polynomial order.

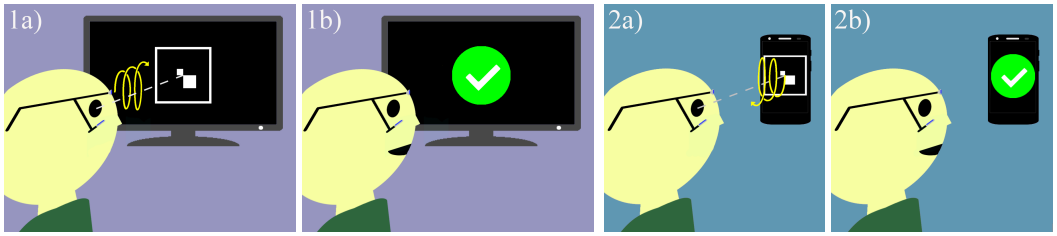


Figure 4.1: Illustrations of two envisioned CalibMe use cases. 1a) The user puts his head-worn eye tracker on, which detects the new user and requests the smart TV to display a collection marker; the user then fixates the marker and moves his head to collect eye-gaze relationships. 2a) The eye tracker detects its calibration became invalid (e.g., based on saliency-gaze overlap) and requests the smartphone to notify the user; the user then moves the smartphone displaying the collection marker to collect eye-gaze relationships. 1b and 2b) the eye tracker notifies the user that the calibration has been performed successfully through other smart devices or visual/haptic/audible feedback, signaling that the system is now ready to use for gaze-based interaction with other devices [24].

the process as well as the gaze estimation accuracy after calibration.

Atypical calibration approaches try to adapt the calibration procedure to their needs. For instance, Pfeuffer et al. [52] propose using objects moving with a known trajectory in a display to calibrate a high-range remote eye tracker placed under a screen, reaching mean accuracies of $\approx 0.6^\circ$ ($\sigma \approx 0.1^\circ$). In a similar stationary scenario, Huang et al. [202] proposed utilizing interactions events between the user and a computing system to collect calibration samples, reporting errors of 2.56°. In a driving scenario, Bernet et al. [203] employed a custom marker (consisting of two nested black squares) that is automatically detected, avoiding, thus, the need for the supervisor and user to coordinate. The user then fixates the custom marker and moves his head in steps of 10 cm exclusively in the horizontal and vertical directions since the proposed approach does not consider depth changes or head rotations. Unfortunately, reported results are based on simulations for their custom made eye tracker, and only the reprojection error for calibration points is given in degrees. In a distinct subsection (5.1), they report best results for evaluation points with a mean error of 2.22 px ($\sigma = 1.42$ px) at a distance of 2.5 m *for noiseless simulated data*; however, there is not enough information in the subsection to infer these values in terms of degrees. More similarly to the pervasive scenario, Evans et al. [191] compared two methods of collecting calibration points in outdoors environments: The first (*moving target*) consists of the user following a partner's thumb with his eyes; the partner then pauses the thumb at five distinct points, which are used for calibration. The second (*head tick*) consists of the users fixating a fixed point and moving their head in $\approx 10^\circ$ steps in an asterisk-like pattern, producing about 25 to 30 calibration points. The collected points were later employed in an offline calibration, and both methods exhibited similar accuracies for central points with a mean error of 0.83° (σ not reported), but the *head tick* approach resulted in better estimations at points in the periphery. The authors also report that the *moving target* method was significantly faster than the head ticks. Pupil Labs employ cocentric circular markers in their *manual marker calibration* in a similar fashion to the *moving target* from [191]; similar

restrictions also apply as mentioned in their website: "*this method is done with an operator and a subject*" [124][4].

It is worth mentioning that some previous works try to counteract calibration degradation through compensation or recalibration. Hornof et al. [204] employs implicit required fixation locations to evaluate the gaze estimation and correct systematic errors. Kolakowski et al. [205] attempt to isolate eye tracker drift based on the corneal reflection gain, which can then be filtered. Sugano and Bulling [136] use gaze input features and saliency maps calculated over the field camera images to (re)-calibrate the eye tracker. Lander et al. [206] perform a recalibration step with a subset of the initial calibration points; afterwards, the updated positions for calibration points not present in the recalibration are extrapolated. Binaee et al. [207] employ a set of ground-truth fiducial positions in a virtual environment to dynamically refine the calibration over time.

4.1.2 On the Selection of Collection Markers

A marker to be employed in the collection of eye-gaze relationships should have the following properties:

1. The user should be able to easily locate and distinguish the reference point to be fixated; for instance, points lying in the intersubsection of lines.

2. Marker detection should be accurate, precise, and require low resources since it must run in real-time in an embedded system alongside other eye tracking related image processing algorithms (e.g., pupil detection [1]).

3. Since the field camera moves w.r.t. the marker, small blurring effects are to be expected. The more robust to blur the marker, the faster the movements allowed, and, thus, the calibration process. However, one should be aware that the user gaze may lag behind the marker at higher velocities due to constrains in human smooth-pursuit capabilities, leading to less accurate gaze-marker relationships.

Hitherto, markers used for calibration follow a similar pattern: They tend to be custom bitonal, nested and cocentric shapes (see Fig. 4.2a to Fig. 4.2d). While such markers meet most of the requirements for a collection marker, they require a unique detection process. In contrast, fiducial markers designed for AOI definition and augmented-reality can meet these requirements and use a single process to detect a large set of markers. Moreover, the detection of such markers is already integrated into most eye tracking software as AOI definition is a common and useful functionality in eye tracking experiments (e.g., ArUco markers are integrated into EyeRecToo and *Pupil Labs' Capture* [67], [124]). Therefore, we propose repurposing one of these markers to be used as collection marker; within the set of fiducial markers, it is likely that one exists, such that the stipulated collection marker requirements are met. In this work, we searched the default predefined ArUco dictionary used in EyeRecToo (*DICT_4X4_250*) for such a marker. Based on our requirements, we

[4] Recently, Pupil Labs added an additional calibration method following the principles of CalibMe; for details, see https://github.com/pupil-labs/pupil-docs/issues/115

found marker `#128` to be particularly suitable for the task. Furthermore, we increased the size of the black marker border from the default size to reduce blurring effects, resulting in the marker used throughout this work (Fig. 4.2e). It is worth noticing that 249 markers remain for regular use in the set, and, if required, larger dictionaries can be generated automatically [208], although an appropriate collection marker from the newly generated set should be chosen following the guidelines from this subsection.

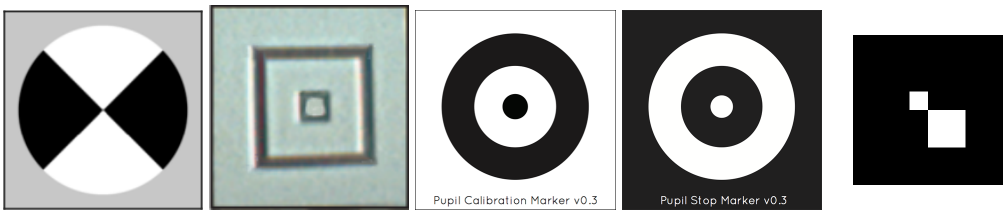


Figure 4.2: Markers used by Evans et al. [191] (a), Bernet et al. [203] (b), Pupil Lab [124] (c,d), and the ArUco marker (`#128`) selected for this work (e) [24].

4.1.3 Rationalized Outliers Removal

Consider a video-based head-mounted eye tracker with one eye camera and one field camera generating data at a predefined rate of r Hz. For each new eye tracker data incoming at timestamp (t), a pupil (p) is detected from the eye image, yielding the pupil center coordinates in the eye image (p_x, p_y) as well as its width (p_w) and height (p_h); similarly, the collection marker (cm) is detected from the field image, yielding its center coordinates in the field image (cm_x, cm_y). Let $D = \{t, p_x, p_y, p_w, p_h, cm_x, cm_y\}$ be the data tuple generated by the eye tracking system every $1/r$ seconds. The goal of the calibration procedure is then to collect data tuples containing pupil (p_x, p_y) and collection marker center (cm_x, cm_y) relationships in order to establish a function mapping pupil to gaze positions – i.e., the point of regard.

Intuitively, wrongly detected values for these variables will perturb the estimation of this function's parameters; thus, one of the supervisor tasks in the regular calibration is to check that the pupil is being detected correctly before association. For an extensive analysis of factors that may influence the correct pupil detection, we refer the user to the work by Fuhl et al. [50]. Additionaly, transient saccading and blinking during collection can be taken into account by collecting a position for a longer interval and taking the median of the samples. However, in an unsupervised calibration where the marker position can be constantly changing w.r.t. the user eye, these procedures are not possible. Thus, alternatives must be found. A common non-domain-specific approach is applying RANSAC to the fitting in order to eliminate outliers (e.g., as done by Bernet et al. [203]). Nonetheless, only outliers that significantly affect the fit are indentified, and, if data is particularly noisy, RANSAC will nonetheless return a fit. Additionally, randomly selecting subsets of points and selecting the best fit may not result in a good transformation function at all. Instead, we propose a

series of rationalized approaches to remove outliers based on domain specific assumptions regarding head-mounted eye tracking setups, data, and algorithms; these outlier removals are described in the sequence, and examples are given in Fig. 4.3.

4.1.3.1 Subsequent Pupil Size Ratio

During calibration, pupil size may change due to physiological factors, or the apparent pupil size in the image may change due to the eye position w.r.t. the camera. Nonetheless, the pupil size for two subsequent data tuples can be expected to remain largely the same due to the difference in magnitudes between the camera frame rate and pupil constriction/dilation speed. Additionally, the apparent size should also remain largely unchanged as the eye pursues the collection marker since no significant eye movement is elucidated. Thus, significant changes in pupil size can be attributed to false pupil detections; an example of outliers detected by this approach are sporadic detections of the iris as pupil – note that the center of the pupil and iris are not necessarilly at the same location [209].

4.1.3.2 Converging Pupil Position Range

Eye cameras are usually placed such that the whole eye, including canthi, is visible. Nonetheless, the pupil position is only expected to fall within a certain range within this image. Certain outliers will evoke pupil detections outside of this range; for instance, when the user blinks, the pupil detection algorithm may sporadically detect glasses frames, make-up, or moles as valid pupils. This outlier removal works by assuming the pupil positions to be normally distributed. Initially, all samples are considered inliers; then, this method computes the mean (μ) and standard deviation (σ) of all inliers, marking as outliers samples falling outside of the range $\mu \pm 2.7\sigma$ (i.e., covering $\approx 99.3\%$ of the distribution). This process is repeated until the amount of inliers converges.

4.1.3.3 Pupil Detection Algorithm Awareness

Some pupil detection algorithms consist of a main and a fallback method. Such fallback methods are typically employed when the pupil is in unfavorable positions for detection and tend to improve the recall of the algorithm (in terms of pupil detections) at the expense of a loss in accuracy of the pupil coordinate, shape, or orientation. For instance, the fallback mechanism of ElSe (which is employed in this work) consists of searching for a point within a dark region and a strong center surround response without providing information about pupil shape and orientation. Thus, this outlier removal depends on the pupil detection method used and consists of requiring that the tuple has a valid pupil size to employ samples solely from the main method.

4.1.4 Automatic Selection of Evaluation Points

One of the main advantages of CalibMe is regarding the high amount of eye-gaze relationships that can be collected in a short period of time. After outliers removal, it is expected that only valid data tuples are left. While the majority of these samples should go towards

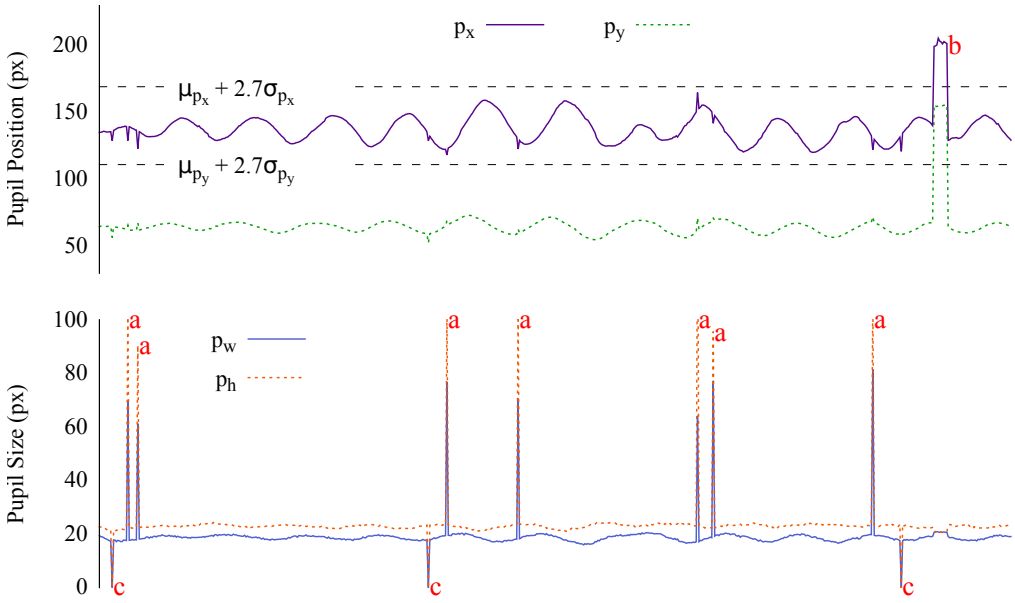


Figure 4.3: Rationalized outlier removal examples during a calibration of ≈ 21 s. Subsequent pupil size ratio outliers are identified by the letter a, converging pupil position range outliers by b, and pupil detection algorithm awareness by c. Notice how the pupil position estimate (p_x, p_y) is significantly corrupted by such outliers [24].

calibration in order to improve the regression, enough data tuples are produced that we can afford to exclude some samples from calibration to evaluate the resulting gaze estimation afterwards. In this subsection, we define a parameterizable method to select these evaluation points automatically.

The proposed selection method is defined by four parameters: the granularity (g), horizontal stride (Δ_x), vertical stride (Δ_y), and evaluation point range factor (rf). According to the first three parameters, a lattice is built around the center (f_x, f_y) of the field image; the lattice is defined by the points $(f_x - g \times \Delta_x, f_y - g \times \Delta_y)$ and $(f_x + g \times \Delta_x, f_y + g \times \Delta_y)$, with points laid down at every (Δ_x, Δ_y) stride inside this region. Afterwards, an elliptical area with a horizontal radius of Δ_x / rf and vertical radius of Δ_y / rf is associated with each lattice point, resulting in a configuration similar to that of Fig. 4.4. After outliers removal, for each lattice point the remaining collected tuples are searched to find the tuple lying inside the associated elliptical region with minimal collection marker center distance from the lattice point. Each tuple found this way is then selected for evaluation. Afterwards, all tuples with a collection marker center matching those selected for evaluation are removed; in this manner, no evaluation point is used in the calibration regression. The resulting evaluation based on these points yields two metrics: 1) the reprojection error for evaluation points, which measure the *calibration accuracy*, and 2) the ratio of lattice points that have an evaluation point assigned to it, which represents the *calibration coverage*.

It is worth noticing that the evaluation points are likely to be spatially close to calibration

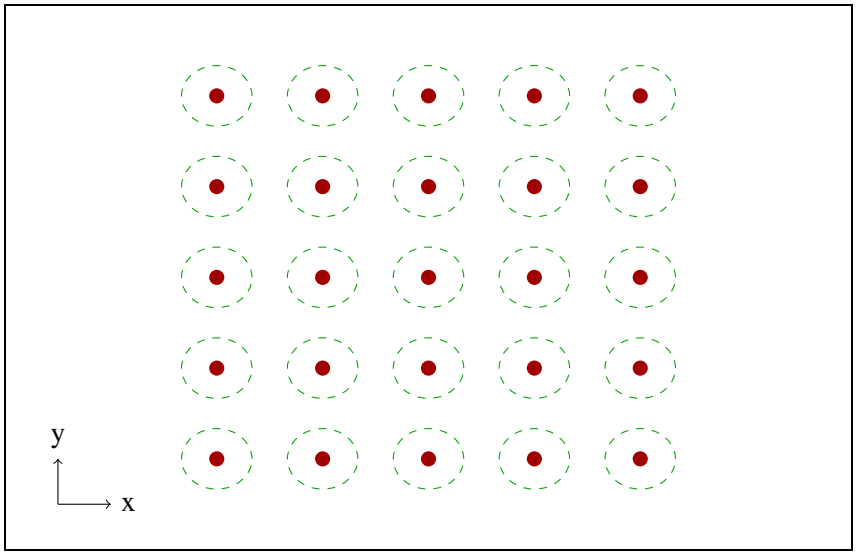


Figure 4.4: Evaluation lattice example ($g = 2$, $\Delta_x = 7°$, $\Delta_y = 6°$, and $rf = 3$) on a field of view of $56° \times 42°$ [24].

points. The closer these points are, the more biased to superior results the evaluation is likely to be since the residuals for calibration points are minimized. Nonetheless, contrary to typical N-Points calibrations, CalibMe collects a large and sparse amount of points for regression; combined with the low order polynomials commonly used in eye tracking, this results in the minimization being spread over the interpolation area instead of concentrated on a few points. Furthermore, even if evaluation points were to be selected independently from the calibration, these points are likely to fall nearby calibration points if the employed calibration pattern covers an adequate range of the field image. Moreover, as shown in the following subsection, collecting evaluation points independently from the calibration process not only requires a second collection process (and, thus, additional time), but these points are unlikely to fall on the calibration surface, therefore biasing the evaluation results to inferior results due to parallax error effects.

4.1.5 On Calibration Movement Patterns

Prior to the analysis of different calibration movement patterns, it is paramount to elucidate 1) how the proposed calibration approach differs from a typical calibration, 2) what is the effect of allowing free head movements, and 3) how this affects the resulting evaluation. To illustrate these concepts, consider Fig. 4.5, which shows the side view of a head-mounted eye tracking setup. Notice that the user's eye and the head-mounted field camera lie at different heights, which tends to be the common case[5]. Typically, the system is calibrated by employing calibration points in a planar surface (i.e., the *calibration plane*); within

[5]For simplicity, we will not include the discrepancies when the camera is unaligned horizontally w.r.t the eye, but analogous effects result.

this plane (e.g., the white dot), the gaze estimation is the most accurate. Whereas on an ideal stationary setup (e.g., a computer screen) this can be expected, it is an unrealistic expectation for pervasive and mobile scenarios since the interactive objects will rarely lie on the calibration plane (e.g., the black dot). When the *object plane* differs from the *calibration plane*, the gaze estimation will produce an inaccuracy proportional to the distance between the calibration plane and object plane because the eye and field camera view the scene from a different angle, resulting in the *parallax* error [47], [191]. Moreover, notice that if the user is only allowed to move his head vertically and horizontally while fixating a stationary target, the resulting calibration is equivalent to that of a plane. On the contrary, if the user is allowed to change depths or rotate his head, the result can be seen as a *calibration surface* (illustrated as a rotation around the center of the camera in Fig. 4.5). From this, three conclusions follow:

1. The underlying regression model should take into account that the calibration surface is not planar. Effectively, this is already the case since most current models employ curved relationships in order to compensate for lens distortions and eye curvature.

2. Relative to the gaze estimation based on a planar calibration, the gaze estimation based on a surface calibration may exhibit a larger or smaller parallax error depending on the object position and surface curvature.

3. Attempting to evaluate a surface calibration on a plane (or vice versa) *will bias the resulting accuracy due to the introduced parallax errors*. In fact, this applies for any two distinct calibration and evaluation surfaces (e.g., two distinct planes).

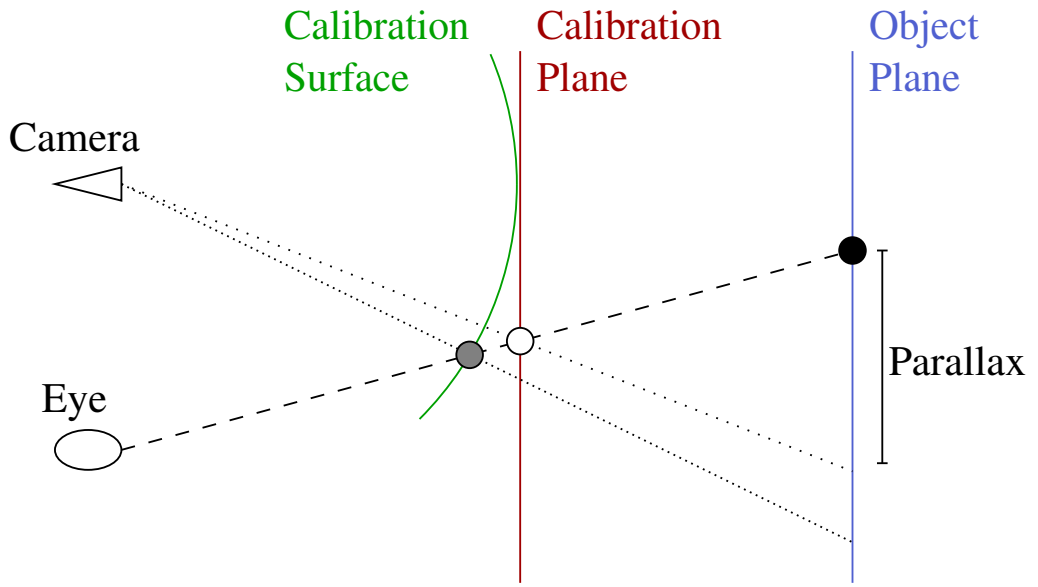


Figure 4.5: Parallax effect illustration between a curved calibration surface, a straight calibration surface (i.e., a plane), and the object plane [24].

As previously mentioned, one of the advantages of collecting evaluation points simultaneously with the calibration is the fact that the calibration and evaluation points will lie roughly on the same surface. If evaluation points were to be collected in a separate step, it is unlikely that the user will be able to reproduce the exact head and eye displacement as the one performed during calibration. Therefore, the gaze estimation is deteriorated due to the resulting parallax errors, producing an underestimation of the calibration quality. Hence, we evaluate the investigated collection movement patterns using the method proposed in the previous subsection.

Initially, we considered several continuous collection movement patterns for evaluation, such as a spiral, star, horizontal path, vertical path (shown in Figures 4.6a-d, respectively) as well as letting the user move freely. However, it quickly became apparent during a pilot study[6] that there are some properties that constitute superior patterns, recalling that the user must "draw" these patterns with the marker using the view of the field camera as "canvas":

1. The pattern should have intuitive parameters: While the pattern itself is usually clear from the illustration, most subjects were confused by parameters – e.g., how many horizontal lines in pattern Fig. 4.6c should be performed. Curiously, no questions were asked regarding the parameters of the spiral pattern (Fig. 4.6a).

2. The initial position of the pattern should preferably be in the center to allow starting in a natural position.

3. The extremeties of the field should be covered without necessitating precise movements from the user.

These criteria eliminated all but the spiral and star patterns; the spiral pattern is original from this work, whereas the star pattern is similar (albeit without pauses) to the *head ticks* pattern employed by Evans et al. [191]. The final form of the new pattern that we proposed for novices is a spiral movement, starting at the center, going outwards, and then spiraling back to the center. This is the spiral used in this work unless explicitly mentioned otherwise.

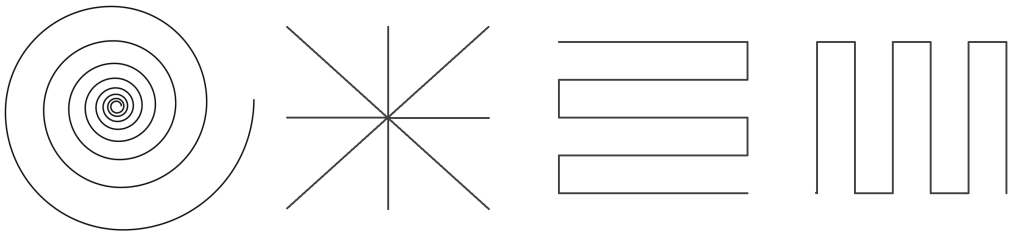
4.1.6 Experimental Evaluation

In this subsection, first the two investigated collection movement patterns are evaluated against each other using the proposed automatic evaluation point selection. Afterwards, the best performing one is compared against a typical 9-Points calibration based on the proposed automatic evaluation point selection and a regular twenty five points grid evaluation.

4.1.6.1 Participants, Apparatus, and Metrics

Experiments were managed by an expert with more than one year of experience in conducting mobile and stationary eye tracking experiments. Five adult subjects participants

[6] During the pilot, the aim of the experiment was explained to five subjects, after which they performed the free-form movement; the users were then shown illustrations of each pattern in a random order and asked to perform those movements.



(a) Spiral **(b)** Star **(c)** Hori. Path **(d)** Vert. Path

Figure 4.6: Examples of movement patterns that can be employed in combination with collection markers [24].

took part in the evaluation (4 male, 1 female), and two of them wore glasses during the experiments. The subjects were briefed about the procedure, including in regard to head velocity and the goal of collecting the marker in multiple locations w.r.t. the field camera. The subjects were asked to verbally communicate when they were finished performing the movement, and *no instructions were given in regard to the collection duration for each pattern* as to not introduce artificial limits to the collection timing.

The experiment was conducted using a Dikablis Pro eye tracker [160]. This device has two eye (@60 Hz) and one field (@30 Hz) cameras; data tuples were sampled based on the frame rate of the field camera. The field camera was equipped with a 1.5× wide turn lens; camera parameters were estimated for use in the marker pose estimation, but the field image was not undistorted. CalibMe was integrated into EyeRecToo, which was used to record and conduct the experiments. Pupil detection was performed using ElSe, and a bivariate second order polynomial regression was employed for gaze estimation in all cases. The poster shown in Fig. 4.7 was used as stimuli and placed at a distance of approximately 1.1 m from the participants. This poster was designed for the reference points to cover about $40^\circ \times 30^\circ$ considering that after this range head movements become a regular feature of gaze shifts [210]. The eight red points in the extremities and the center of marker `#128` were used as calibration points for the 9-Points method. Blue dots and the center of marker `#128` were used for the twenty five points evaluation. CalibMe automatic evaluation point selection configuration was set to match the setup of these evaluation dots ($g = 2$, $\Delta_x = 7^\circ$, $\Delta_y = 6^\circ$, and $rf = 3$). The metrics employed in this evaluation are:

Mean Angular Error (ε): evaluated as the mean of the euclidean distances between the evaluation point coordinates in the field image and the resulting coordinates from the gaze estimation.

Calibration Time (τ): evaluated as the timestamp difference between the last and first collected calibration tuples.

Pattern Coverage (γ): evaluated as the ratio between lattice points with an associated evaluation point and the total count of lattice points (25 in this study); this metric is only meaningful for CalibMe.

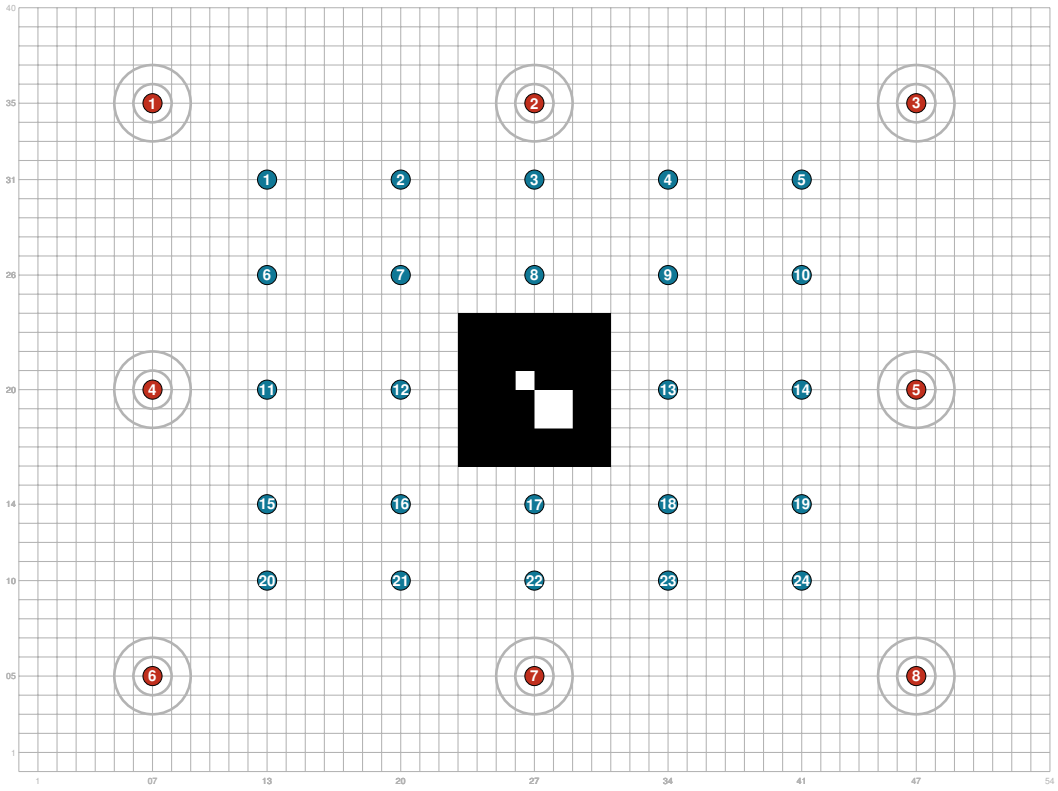


Figure 4.7: The calibration poster used during experiments and placed at 1.1 m away from the subjects. The red points cover an area of $\approx 40^\circ \times 30^\circ$ and are used for the 9-Points calibration together with the center of marker `#128`. Blue points are employed for evaluation together with the center of marker `#128` and lie within the interpolation area of the 9-Points calibration [24].

4.1.6.2 Collection Movement Pattern Comparison

Each participant repeated each pattern three times, and no significant differences between repetitions were found; a visualization of resulting collection marker coordinates on the field image for one of the participants is shown in Fig. 4.8.

For the inter pattern comparison, we have aggregated all repetitions. No significant differences were found between the *Spiral* and the *Star* patterns in terms of calibration time ($F(1,28) = 0.690$, $p = 0.413$). However, the *Spiral* produced significantly larger coverage ($F(1,28) = 34.908$, $p = 0.0000023$). While no significant differences in terms of angular error were found ($F(1,28) = 3.486$, $p = 0.0724$), Fig. 4.9 suggests the *Spiral* to have a small advantage over the *Star* in this regard. Additionally, this figure also exhibits the outcome in case no outliers removal is performed. Particularly interesting in this case is the single angular error outlier above 3.5°. This large error stems not from a bad estimation but from an outlier selected as evaluation point, demonstrating the importance of performing the outlier removal *before* points are automatically selected for evaluation.

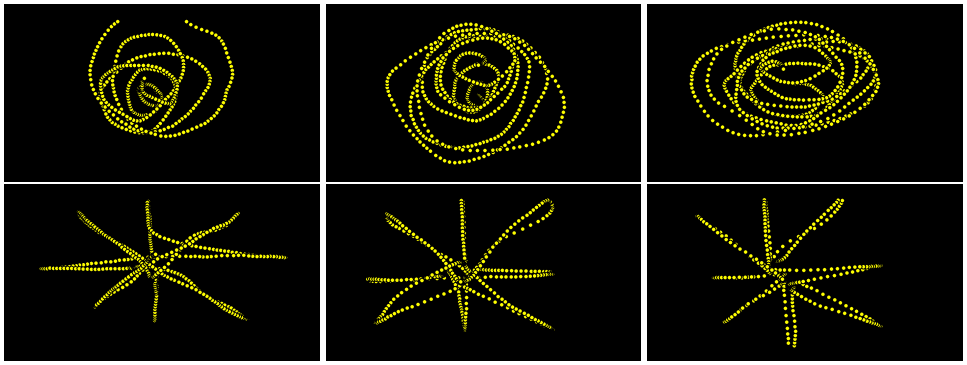


Figure 4.8: Collection marker center coordinates on the field image for one of the subjects when performing the *Spiral* (top) and *Star* (bottom) patterns repetitions [24].

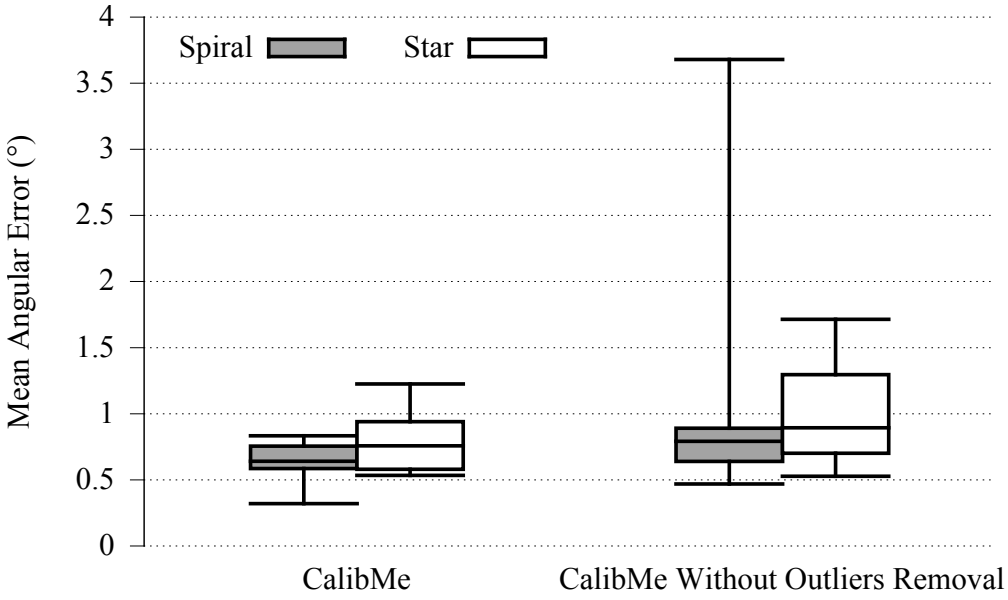


Figure 4.9: Accuracy for the *Spiral* and *Star* movement patterns measured through CalibMe's automatic evaluation point selection with and without outliers removal [24].

4.1.6.3 CalibMe and 9-Points Calibration Juxtaposition

After performing the previously described experiment, each subject participated in a second phase, whose aim is to juxtapose calibrations performed with CalibMe against a typical 9-Points calibration in a setup as similar as possible. Initially, the user rested his head on a chin rest placed at 1.1 m from the calibration poster shown in Fig. 4.7. During this experiment, subjects were instructed not to communicate verbally in order to minimize head movements. The user then performed a 9-Points calibration, followed by a twenty five

points evaluation. The experimenter was responsible for manually selecting the gaze points in the field camera view. After a point was selected, data tuples were collected for 500 ms, and the median of the collected pupil coordinates was associated with the gaze position. Following this step, the system produced an audible feedback so the user knew when to move to the next point, minimizing the amount of communication between experimenter and subject, thus making this calibration procedure faster and less error prone. After both procedures were finished, the experimenter slowly moved the subject's head away from the chin rest, moved the chin rest horizontally out of the way of the subject, and then moved the subject's head to the initial position by aligning it at a distance with the chin rest. With the head now free to move, the subject was instructed to perform the *Spiral* collection movement pattern.

As shown in Fig. 4.10, this procedure results in two distinct surfaces – 1) *Pattern:* a surface formed by the sparse points collected while performing the *Spiral* pattern, and 2) *Poster:* a planar surface along the calibration poster (Fig. 4.7). Therefore, a direct comparison between the regular and the collection method calibrations is *biased*. If we evaluate these solely on one of the surfaces, the accuracy of the other will be underestimated due to parallax errors (and vice-versa). Hence, we juxtapose these by cross-evaluating sets of calibration points from each surface on sets of evaluation points lying on both surfaces instead.

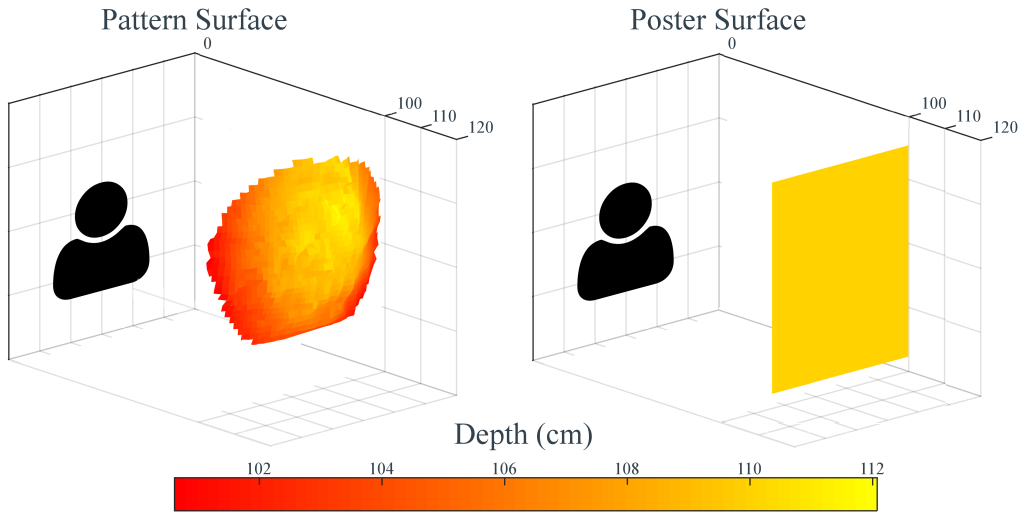


Figure 4.10: Surfaces produced by calibrating using a spiral head movement pattern and a regular 9-Points calibration [24].

The following sets of calibration points were analyzed in this manner:

- CalibMe: this set consists of samples collected during the spiral movement, excluding the tuples selected for evaluation, tuples with collection markers at the same position as evaluation ones, and outliers identified by CalibMe. These samples lie on the *pattern surface*.

- *Outliers*: CalibMe without outliers removal.
- *9 Points:* the eight (red) calibration points plus the marker center; these samples lie on the *poster surface*.

These calibration sets were evaluated on two different evaluation sets:

- *Pattern (Reserved):* samples lying on the *pattern surface* that were automatically reserved for evaluation by CalibMe.
- *Poster (25 Points):* the twenty four (blue) evaluation points that lie on the *poster surface* plus the marker center.

The resulting gaze estimation accuracies from these evaluations are shown in Fig. 4.11. As expected, both methods exhibit better accuracy when evaluated on their respective calibration surface than in a different one. The gaze estimation error due to the parallax effect can be noticed when comparing the same calibration set across different evaluation surfaces, which shows $\approx 0.7^\circ$ of error in all cases. While this is in line with the expected error magnitude given our setup, it is worth noticing that part of these errors are contributed by occasional points that fall outside of the calibration interpolation range, for which the gaze estimation is expected to have an inferior accuracy. When comparing CalibMe against the 9-Points calibration evaluated on their respective surfaces, no significant difference was found in terms of accuracy ($F(1,8) = 3.372$, $p = 0.104$). Nonetheless, this figure suggests that CalibMe produces slightly better results, yielding a mean angular error averaged over all participants of 0.59° ($\sigma = 0.23^\circ$) in contrast to 0.82° ($\sigma = 0.15^\circ$) for the 9-Points calibration, attesting for the efficacy of the proposed approach.

While no significant differences were found when comparing CalibMe and its outliers inclusive counterpart (*Outliers*), when visually inspecting the figure an anomaly (with a mean error larger than 4.5°) is clearly visible in the *25 Points* evaluation. By virtue of CalibMe's automatic evaluation point selection, we quickly traced this anomaly as a result of a low coverage and a single pupil detection outlier in the right eye during a blink. This particular subject had a coverage ratio of 68%, whereas all other subjects had coverages of at least 80% ($\mu = 87\%$). As shown in Fig. 4.12, this subject did not cover the left part of the 25 points evaluation area, resulting not only in a subpar accuracy, but also in no polynomial regression constraints in that area. The aforementioned outlier happens to lie in this unconstrained area, thus significantly skewing the gaze estimation at those points and artificially amplifying the mean angular error.

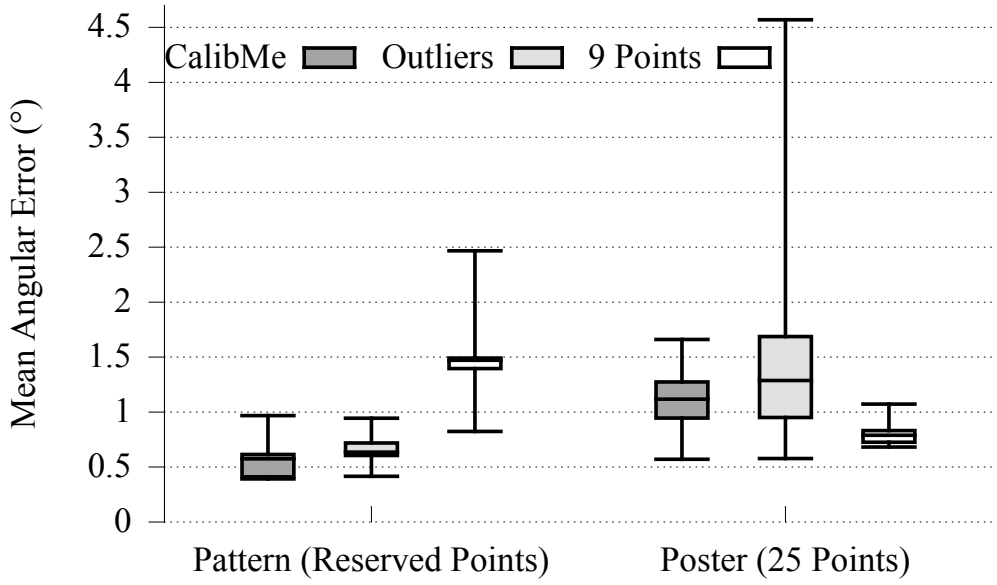


Figure 4.11: Mean angular error evaluated on points lying on the pattern and poster surfaces [24].

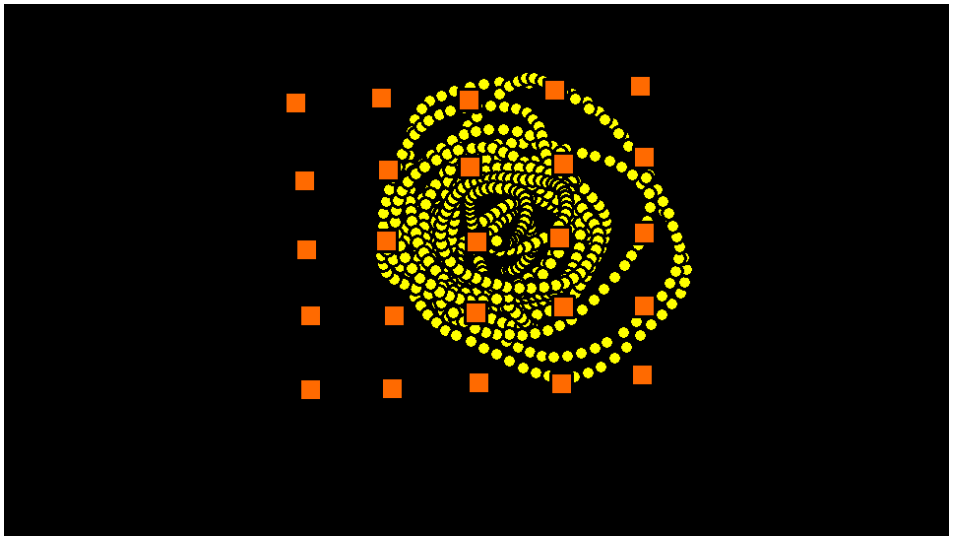


Figure 4.12: Collected tuples (yellow circles) and the 25 evaluation points (orange squares) for the subject with anomalous mean angular error when outliers are considered. Notice how the left side of the evaluation points are not covered by collected tuples [24].

4.1.6.4 A Note on Calibration Time

Assuming a qualified and experienced supervisor, the 9-Points calibration time is rather constant if nothing disrupts the collection (e.g., in case one position must be repeated, the calibration flow is broken, and calibration time increases significantly). We can model the optimal time required for such calibrations by summing up 1) the sampling time for each point, 2) the time for the subject to react to the audible feedback and saccade to the next target, and 3) the time for the supervisor to react to the saccade and start the next sampling. Empirically, we found that the time to collect a single points in our setup is ≈ 1.67 s – i.e., one can expect about 15 s and 42 s to collect nine and twenty five points, respectively.

As previously mentioned, we intentionally did not impose time limitations on the subjects when performing the pattern calibration as to not bias results. Intuitively, the pattern calibration time depends on the spatial distribution and amount of collected tuples. These can be mainly determined by two factors: First, the user's skills and internal mental model of the system; for instance, the better a user can abstract the marker position w.r.t the camera, the more efficiently he can utilize his collection time. Second, the speed of the marker relative to the camera/subject; the faster the marker moves, the faster the calibration. However, several elements influence the latter factor, such as the camera resolution, frame rate, shutter type, marker detection blur robustness, and human smooth-pursuits physiological limitations. These elements are further discussed in Section 4.1.7.

Since estimating user's skills is rather subjective, and finding users of different skill levels is unpractical, we approach this analysis from an alternative perspective. By downsampling the data from the previous experiment, we can study the effect of the amount of tuples on the method's accuracy. Here, downsampling is performed by using only one tuple in each *down sampling factor* (*DF*) tuples so that spatial distribution is preserved. It is worth noticing that a reduced amount of samples can compromise the coverage ratio and, thus, limit the information provided by the automatic evaluation. Therefore, we perform evaluations against the twenty five points instead. Fig. 4.13 shows the effect of this downsampling in the mean angular error. It is clear that subjects collected an amount of tuples ($\mu = 549.8, \sigma =$

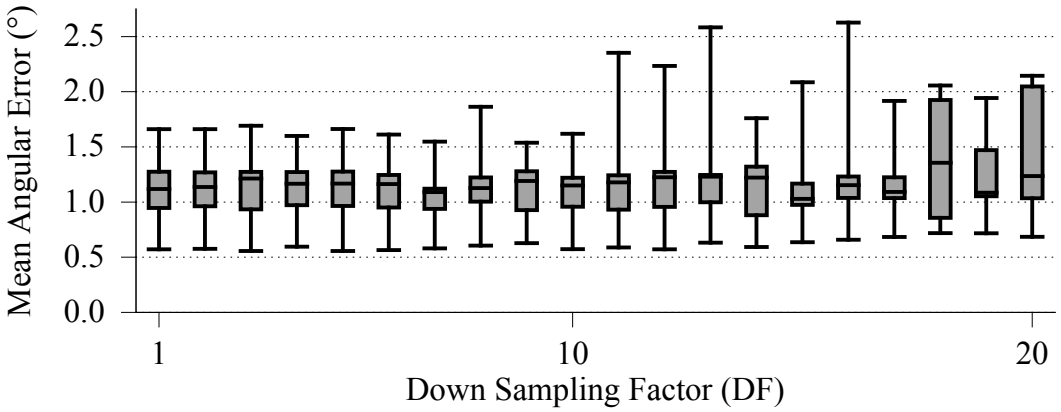


Figure 4.13: Downsampled CalibMe evaluated on the twenty five point grid, showing that downsampling by small factors retains accuracy as long as the spatial distribution is preserved [24].

159.11) much larger than required to reach satisfactory accuracy. Thus, the limiting factor becomes the spatial distribution of the collected tuples and, consequently, the speed of the marker relative to the camera/subject. Assuming a conservative *DF* of two, the proposed approach already becomes significantly faster than the 9-Points calibration, requiring on average 10.12 s as shown in Fig. 4.14.

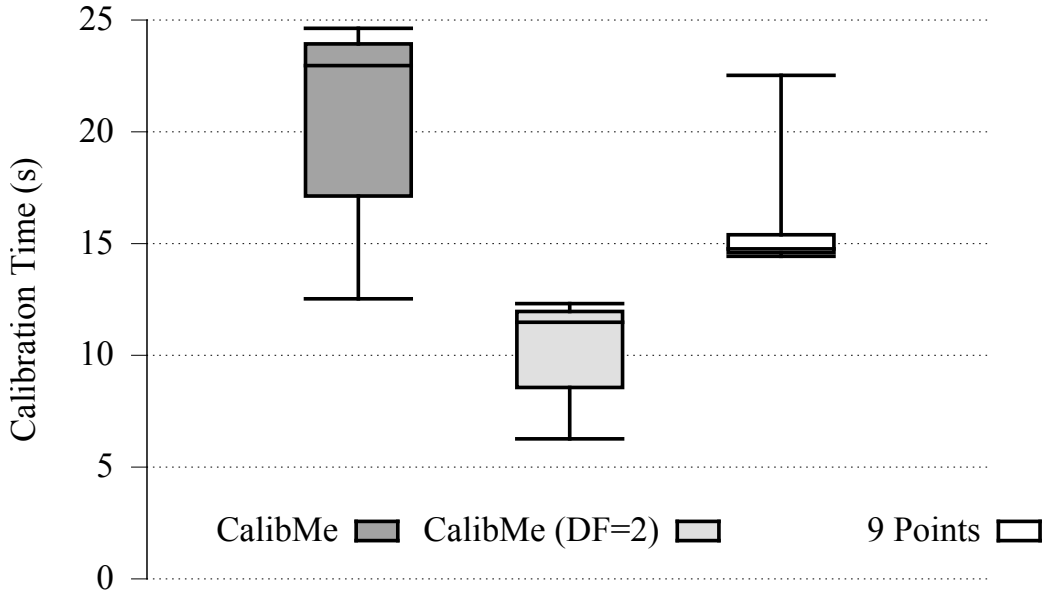
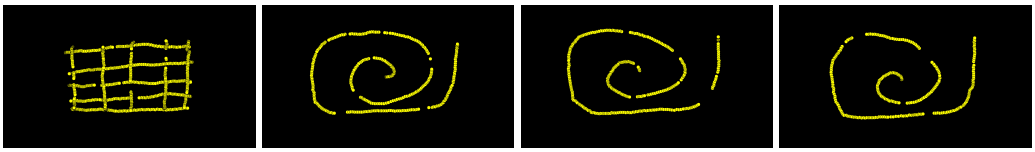


Figure 4.14: Calibration time for CalibMe, CalibMe downsampled by a factor of two, and the 9-Points calibration [24].

In order to corroborate this time estimate, we conducted a separate experiment with a user considered experienced enough to have a good mental model of the marker movement relative to the camera: one of the CalibMe developers. As previously mentioned, a small amount of collected tuples may compromise the coverage ratio, limiting the amount of information provided by CalibMe's automatic evaluation. Therefore, the user first collected eye-gaze relationships to be used for evaluation by staring into the center of a collection marker displayed in a cellphone screen and moving the cellphone in a grid pattern, as shown in Fig. 4.15a. This collection took ≈ 43 s, and tuples identified as outliers were removed from the evaluation set, resulting in 816 evaluation tuples. Afterwards, the user conducted ten independent CalibMe calibrations by moving the cellphone in spiral patterns, which were evaluated against the aforementioned evaluation tuples. It is worth noting that the user opted to perform a single outward spiral instead of the outward-inward pattern recommended for novices; instances of the resulting patterns can be seem in Figures 4.15b-4.15d. The average calibration time for these calibrations was 10.68 s ($\sigma = 0.86$ s), reaching an average angular error of 0.69° ($\sigma = 0.044°$). The individual values for each collection are shown in Fig. 4.16. The result from this experiment is in line with both the expected

angular error and calibration time, thus endorsing the time estimate of the downsampling analysis.



(a) (b) (c) (d)

Figure 4.15: The points used for evaluation (4.15a) and points from three (out of ten) distinct calibrations performed by the user (4.15b-4.15d) [24].

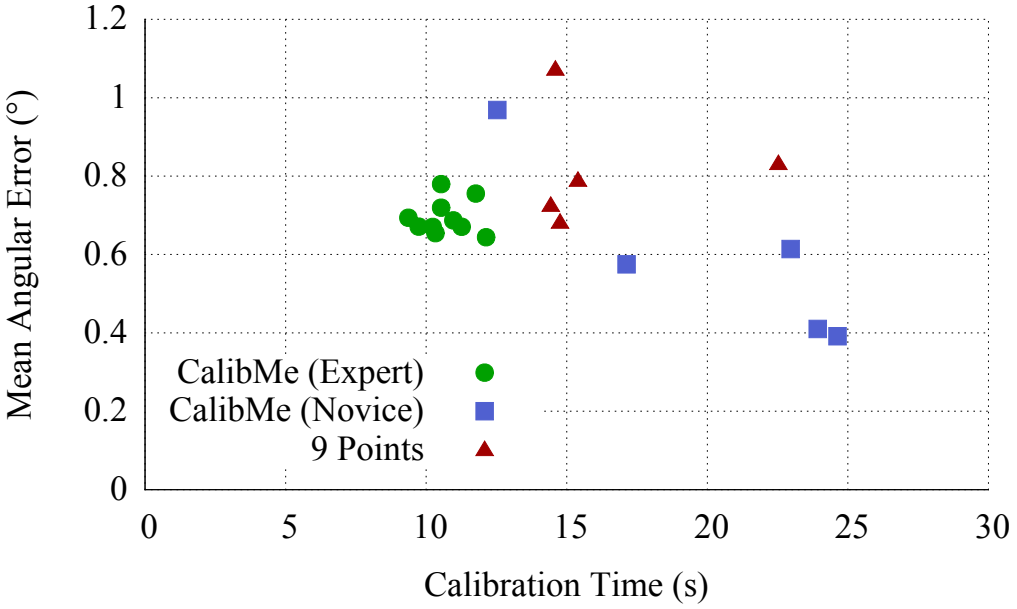


Figure 4.16: Calibration time and resulting mean angular error for the ten calibrations performed by the user; also shown, is the calibration time and mean angular error for the reprojection of the evaluation points [24].

4.1.7 Limitations

Naturally, the key limitation for CalibMe is in regard of marker and pupil detection. These tasks face similar challenges such as occlusion, highly skewed viewing angles, poor or irregular illumination, and motion blur [1], [211]. Nonetheless, there is significant research for these tasks in pervasive and challenging scenarios (see [50], [211]), and spurious incorrect detections during calibration can be eliminated through the proposed rationalized outliers removal methods. In particular, motion blur could be greatly alleviated by the usage of global-shutter sensors.

Aside from these detection challenges, human smooth-pursuit physiology imposes a lower bound on the calibration time. As marker speed increases, the smooth-pursuit gain (i.e., the ratio between the marker and smooth-pursuit speed) starts deviating significantly from one [212]. As a result, the marker center may not correspond exactly to the true gaze position, specially at higher speeds. These periods could potentially be identified based on real-time eye movement detection algorithms such as *I-BDT* [66].

4.1.8 Conclusion

In this section, we have introduced CalibMe, consisting of a collection marker that does not incur any additional processing to common eye tracking systems and a set of techniques to 1) quickly collect eye-gaze relationships for calibration, 2) remove ill-constrained relationship outliers, and 3) automatically reserve tuples for evaluation. As a result, CalibMe allows eye tracker users to quickly calibrate the system anywhere without supervision, giving feedback not only in terms of gaze estimation accuracy, but also on the calibration area coverage relative to the field camera. The proposed method reached accuracies ($\mu = 0.59°$,$\sigma = 0.23°$) commensurable to a 9-Points calibration ($\mu = 0.82°$,$\sigma = 0.15°$), well within physiological values, and calibration can be performed in ≈ 10 s, thus increasing gaze-based HCI usability. Future work includes developing and integrating methods to assess gaze estimation validity, and extending the proposed approach to 3D gaze estimations based, for instance, on vergence information [213].

4.2 Gaze Estimation

In contrast to their remote counterparts, video-based head-mounted eye trackers[7] provide a singular perspective from the users' egocentric point of view, allowing for plural gaze contingency virtually anywhere. We envision a future in which individuals wear miniaturized versions of such devices on a daily basis not only for digitally intermediated interaction with all kinds of devices [28]–[34], but also for other added benefits such as preventive health monitoring [36], self quantization [37], [38], daily and life logging [39], [40], advanced driving assistance [41], and alternative forms of veillance [45]. In fact, given the current pace of head-mounted eye-tracking system modularization and miniaturization, the integration of head-mounted eye-tracking into head-worn hardware – e.g., smart, augmented-reality, and prescription glasses – seems evident.

Given the aforementioned integration and recent advances in robust pupil tracking in pervasive scenarios (c.f. [7], [50]), the major remaining challenge hindering a wider adoption of ubiquitous eye-tracking seems to be device *slippage*. *Slippage* is characterized by changes in eye tracker pose (translations or rotations) w.r.t. the calibration pose, thus corrupting the learned mapping function from the eye-camera feature space to gaze in field camera coordinates: A phenomenon also known as *calibration drift* [21], [22], [49]. In this work, we propose Grip, a glint-free and slippage-robust gaze estimation method for video-based head-mounted eye trackers. The method was evaluated using previously collected

[7] Head-worn devices capturing images of at least one of the user's eyes and part of his / her field of view.

data from a large scale pervasive eye-tracking study, achieving a decrease in average participant median angular offset by more than 43% w.r.t. a regular polynomial gaze regression method.

4.2.1 Background and Related Work

Let us consider *a)* a rigid-body eye tracker consisting of three cameras (left eye, right eye, and field), *b)* that all cameras' intrinsic and extrinsic parameters are known a priori, and *c)* that the eyes are perfect spheres. Our goal, then, is to estimate a 2D *gaze point* in the field camera's image based on the eye images at a particular instant. Geometrically, this estimation can be achieved as follows:

Monocular gaze estimation: Given that a) the eye *center* and *visual axis* can be estimated w.r.t. its corresponding eye-camera, and b) the transformation from that eye-camera to field-camera is known, one can derive the user's 3D *gaze vector* (*origin* and *direction*) in field-camera coordinates. Projecting this 3D gaze vector to the field camera's image plane, yields a 2D *line* in which the desired 2D gaze estimate lies. However, without gaze depth information, we cannot infer at which point along this line exactly[8].

Binocular gaze estimation: Similarly to the monocular version, we can estimate a 3D *gaze vector* for each eye. An approximation of the 3D *gaze point* in field camera coordinates is given by these vectors' intersection point. Projecting this 3D gaze point to the field camera's image plane, yields the desired 2D gaze estimate.

In theory, thus, even if the eye tracker *slips*, *there is no associated calibration drift*: The geometry still works out. In fact, eye-tracking systems operating on this geometrical principle *should* be robust to device slippage. This is the case, for example, of the glint-based Tobii Pro Glasses 2, ÖÖGA [119], [214], and OMG! [215] eye trackers. However, these systems typically have several other drawbacks, usually requiring a highly controlled and calibrated setup that can be unattainable in many head-mounted eye-tracking scenarios [111], [216]. They also make use of multiple glints, whose detection is a critical challenge, particularly in outdoor environments [118]. Moreover, the fixed inter-camera transformations assumptions mean the systems cameras cannot be adjusted to users' idiosyncrasies nor to capture distinct fields of view. The glints and rigid body also hinder users from wearing the devices in conjunction with glasses.

So where does the *slippage-associated calibration drift* come from? Alas, the geometrical solution simplifies the system considerably, restricting attainable system accuracy. For instance, the eyes are not spheres, seldom the 3D gaze vectors intersect in practice[9], and a calibration[10] is still required to estimate the *optical-to-visual axes offset*. Moreover,

[8] The exception being if the eye and field camera are aligned, in which case the 3D vector projection resolves to a 2D point.

[9] One can use the *mean point* of the *shortest line segment* connecting the two *3D vectors* instead.

[10] One calibration point suffices, but more are generally required to achieve satisfactory accuracy [123].

even if glasses fit under the device and glints / pupil were perfectly detected, lenses refraction can lead to significant errors [196]. Thus, many eye tracker system designers opt to trade-off slippage robustness for increased accuracy and flexibility by employing alternative gaze estimation methods, *which oftentimes are not robust to device slippage*. In such cases, multiple approaches have been proposed to mitigate calibration drift, for instance: *a) The classical pupil-glint vector*, which is also affected by drift but to a lesser extent [205], *b)* Determining camera translation, for example using eyelid templates [217], [218], eye corner tracking [219], gain values differences [205], and tracking automatically selected landmarks [220], *c)* Using saliency maps to detect and correct shifts [136], *d)* Fast and unsupervised calibration [24], recalibration [206], as well as auto-calibration [202] schemes, and *e)* The simplest (and probably most used) approach: Designing the eye tracker to reduce slippage – e.g., through head straps, cloth clips, helmets, or even facial plastic molds [192], [221], [222].

4.2.1.1 Glint-Free Geometry

A promising alternative to the aforementioned gaze estimation methods is to learn the system geometry and intricacies implicitly and map the gaze point directly from the eyes appearance (e.g., [107]). Moreover, such approaches have the potential to allow for the replacement of regular image sensors by less power-hungry ones (e.g., multiple low resolution image sensors [105] or photosensors [88]). However these systems are still in its infancy [216], and its not clear whether they are able to operate adequately under continuously changing conditions such as driving or outdoors.

Stereo setups with multiple cameras can be used to estimate the pupil pose [113] at the cost of added hardware and processing; Tobii Pro Glasses 2 actually use a complex model based on multiple glints and two cameras per eye. Alternatively, temporal information can also be exploited to infer eye and pupil poses. An estimation of eye parameters from *iris* observations (assuming weak perspective) for model-based gaze estimation is given in [112]. Assuming full perspective, [111] proposed a novel temporal glint-free approach to estimate and optimize the eye (as a 3D sphere) from a set of 2D *pupil* observations detected in the eye-camera image plane. Afterwards, 3D pupil estimations are derived as the 3D circle resulting from intersecting the 2D-to-3D pupil unprojection with the eye sphere. This 3D circle's normal vector – i.e., the vector passing through the 3D eye and pupil centers – is used to approximate the *optical axis*. The Pupil Labs 3D gaze estimation bases their eye model on this approach, and "using temporal constraints and competing eye models we can detect and compensate for slippage events when 2d pupil evidence is strong" [223]. Whereas this approach might work in particular cases, there are three key outstanding issues.

1. What are the requirements on the observation set to approximate a *sufficient* eye model? E.g., how many distinct observations are needed? Is there a minimum angular range required? If an insufficient model is used during calibration or gaze estimation, the resulting gaze estimates will be *offset*.

2. Even if we can determine these *sufficiency* requirements based on thorough simu-

lations – e.g., using the works of [108], [109] – how likely is human natural gaze behavior to conform with them? E.g., while you read this text, your eyes are likely to move only around the tiny fraction of your field of view that encompasses the text.

3. Assuming that the previous issues are solved, *compensation still is not instantaneous* and only valid after the model is replaced by a new sufficient one. This is a significant issue for cases in which the eye tracker can be expected to move often, such as during sports [224], and when used by children [225] or by glasses wearers who tend to periodically adjust their glasses' position.

4.2.2 Proposed Method: Grip

Our main goal in this work is to provide a glint-free and slippage-robust gaze estimation method for video-based head-mounted eye trackers that is able to compensate for eye tracker movement instantaneously. The proposed method does not require any camera parameter to be known a priori, thus being compatible with commercial and do-it-yourself eye trackers alike. Our main assumption is that the eye tracker remains a rigid body *after calibration*. Nonetheless, the eye tracker cameras can be adjusted as needed prior to calibration, allowing for the flexibility required to adapt to users' idiosyncrasies and distinct usage scenarios.

The method consists of two main stages: 1) estimating the instantaneous optical axis direction in eye-camera coordinates, and 2) mapping this optical axis to a 2D gaze point in the field camera image plane. We have named our method Gaze regression: instantaneous and pervasive (Grip) alluring to its capability of compensating for slippage instantaneously and that the mapping function is regressed during calibration. Grip is a hybrid method, combining a geometrical approach to derive a slippage-robust input feature for a non-geometrical gaze mapping function.

4.2.2.1 Instantaneous Optical Axis Direction

Our method starts by estimating the *optical axis* direction w.r.t. the eye camera. Following the approach of [111], we unproject the 2D pupil detected on the eye camera image plane to 3D using the circular intersection method of [226]. By considering only solutions that point towards the camera – since the pupil can only be detected when it is facing the eye camera – this procedure yields two circles parametrized by their center position ($\vec{p}$), normal vector ($\vec{n}$) and radius (r):

$$(\vec{p}_1, \vec{n}_1, r), (\vec{p}_2, \vec{n}_2, r), \tag{4.1}$$

as illustrated in Fig. 4.17. This ambiguity can be easily resolved by using temporal information under the assumption that the eye tracker does not move w.r.t. the eye within a brief window containing at least two distinct fixations[11]. For instance, we can disambiguate between these solutions similarly to Świrski's model initialization (c.f. [111] Equation (8)). In practice, this corresponds to a least-squares intersection of lines following the minor radius of each 2D detected pupil ellipses in the set of samples inside the time window, followed

[11] The actual window size depends on the camera frame rate, but usually hundreds of milliseconds should suffice for any modern eye tracker.

by unprojecting this 2D intersection point at a fixed distance, and picking the solution that points *away* from the resulting 3D point (which can be seen as a very rough eye center estimate). Furthermore, since we are only interested in the *optical axis* direction, we do not need to solve the distance-size $(\vec{p}, r)$ ambiguity – i.e., whether the 3D circle is large and far away or small and close to the camera. Thus, this step yields the disambiguated $\vec{n}$ as an *instantaneous optical axis direction estimate*, discarding $\vec{p}$ and r.

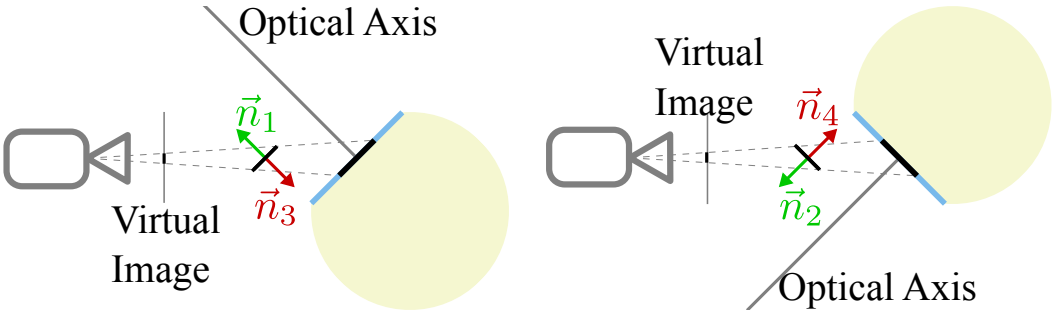


Figure 4.17: The detected pupil (shown in the camera's *virtual image*) is unprojected to 3D, resulting in four solutions. Solutions pointing away from the camera ($\vec{n}_3$ and $\vec{n}_4$) are discarded, but the remaining two solutions ($\vec{n}_1$ and $\vec{n}_2$) are ambiguous because both eye positions generate the same 2D pupil. Further information is required to resolve this ambiguity such as an approximate eye center location [26].

4.2.2.2 Gaze Mapping

Since we do not know the eye positions, nor pupil positions, nor cameras (intrinsic and extrinsic) parameters, we cannot *directly* perform a geometric gaze mapping such as the 3D-to-3D mapping from [97]. Instead, we assume the eye center to be located at the eye-camera center, which introduces an intrinsic error. In practice, head-mounted eye tracking eye cameras are within a radius of a few centimeters from the eye center, whereas gaze points lay at much larger distances. Consequently, the angular offset between our estimated and the real optical axes is small and can be implicitly learned during calibration, similarly to the optical-to-visual axis offset.

We then normalize and parameterize the optical axis with origin at the eye camera center in spherical coordinates such that

$$\vec{n} = (\phi, \theta, 1), \tag{4.2}$$

resulting in a *slippage-robust* feature. Unlike the *slippage-invariant* pure geometrical solution, the proposed approach can introduce an error up to

$$\arctan\left(\frac{d}{D}\right), \tag{4.3}$$

where d is the slippage magnitude and D the gaze point distance, with the introduced error diminishing the closer the fixated point direction is to the slippage direction. Nonetheless, given that slippage magnitude tends to be rather small, $d \ll D$, and this error can be consid-

ered negligible for most cases. Therefore, this feature can be used with typical regression gaze-estimation methods even under the assumption of device slippage.

In this work, we chose to employ a set of two second order polynomial regression methods to perform the mapping to 2D gaze points considering that typical head-mounted eye tracking field cameras contain some level of radial distortion. The first (bivariate) polynomial

$$\mathcal{M} = \begin{bmatrix} 1 & \phi & \theta & \phi\theta & \phi^2 & \theta^2 & \phi^2\theta^2 \end{bmatrix}^T \tag{4.4}$$

is employed for monocular gaze estimation – e.g., if only a single eye camera is available, whereas the second (quadvariate) one

$$\mathcal{B} = \begin{bmatrix} 1 & \phi_L & \theta_L & \phi_L\theta_L & \phi_L^2 & \theta_L^2 & \phi_L^2\theta_L^2 & \phi_R & \theta_R & \phi_R\theta_R & \phi_R^2 & \theta_R^2 & \phi_R^2\theta_R^2 \end{bmatrix}^T \tag{4.5}$$

is employed for binocular gaze estimation, where $\phi_L, \theta_L, \phi_R, \theta_R$ are the optical axis parameters for the left and right eye, respectively. Let C_x and C_y be the polynomial coefficients estimated during calibration per gaze estimation configuration – i.e., left eye monocular, right eye monocular, and binocular. Thus, the 2D gaze point (g) horizontal (g_x) and vertical (g_y) components can be estimated as

$$\begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} C_x \\ C_y \end{bmatrix} \mathcal{P}, \tag{4.6}$$

where $\mathcal{P} = \mathcal{M}$ and $\mathcal{P} = \mathcal{B}$ for the monocular and binocular cases, respectively.

The rationale behind employing $\mathcal{B}$ instead of simply averaging the left and right eye monocular estimates (as suggested by [47]) is that it provides a more complex model that allows us to better compensate for *parallax* errors through eye vergence. The drawback, naturally, is that it requires significantly more calibration points (13, as compared to 7 for the monocular case) to regress C_x and C_y. Whereas this might be a usability issue for screen based calibrations, it can be achieved under ten seconds virtually anywhere by unsupervised users through the CalibMe calibration technique [24]. Moreover, due to the slippage robustness, calibration should only be required whenever the rigid body assumption is broken – e.g., if the cameras are moved independently.

4.2.3 Evaluation

To demonstrate the proposed method's applicability to pervasive real-world scenarios, we applied it to previously collected data from a large scale unconstrained pervasive eye-tracking study investigating the gaze behavior of museum visitors [53].

4.2.3.1 Data Collection

Data was collected in parallel throughout five days using four sets of *Pupil Labs* eye trackers and *Microsoft Surface* tablets (carried in a backpack). EyeRecToo was used to drive the eye trackers, with eye (Pupil Cam2) and field cameras (Pupil Cam1) capturing images of 400 px^2 at 60 fps and *720p* at 30 fps, respectively. Ten researchers *without previous*

head-mounted eye-tracking experience conducted the experiment after receiving a thirty minutes briefing from an expert with more than three years of experience. Visitors arriving at the top of the *Grand Staircase* of the Austrian Gallery Belvedere were invited to join the experiment only restricted by their consent age (18 or older) and language (English or German) – i.e., *no selection of participants suited for eye tracking was performed.* In total 109 subjects (63 females, 46 males) took part in the experiment, averaging 34.86 years of age ($\sigma = 14.62$).

4.2.3.2 Procedure and Stimuli

After donning the backpack and eye tracker assisted by a researcher, each participant stood on a floor mark approximately 1.16 m from a CalibMe collection marker. This marker was used to collect ground-truth gaze positions and stood at about the same height as the exhibition pieces. The researcher then adjusted the scene camera to center the collection marker and eye cameras to a suitable position. Subsequently, the participant was instructed on how to perform the *ground-truth* gaze collection: Keep gaze fixed at the center of the collection marker while moving their head in a spiral fashion smoothly and slowly. The researcher then started the recording, controlling when gaze collection started and stopped. This first gaze collection (henceforth *Collection 1*) was performed soon after recording started. *Collection 1* was used to calibrate the eye tracker in real time, after which the participant was asked to gaze at four Post-its® about 25° away from the center of the calibration marker for the researcher to assess gaze estimation quality. If the gaze estimation was considered subpar, *Collection 1* was repeated and the eye tracker recalibrated. For reference, one of the participants during *Collection 1* is shown in Fig. 4.18.

The participant was then instructed to freely roam through four rooms (containing more than 30 distinct paintings and sculptures) as he/she wished, and that the researcher would meet him at the end of the last room. During their visit, participants were allowed to interact with other visitors.

At the end of the visit, the researcher and participant proceeded to a separate room where: a) a second gaze collection (henceforth *Collection 2*) was performed following the same procedure as *Collection 1*, b) the subject was interviewed and performed a remembrance mapping task, and c) the participant answered a questionnaire containing museum-visit and eye-tracking related questions. After the visit, participants were rewarded with a small souvenir. Originally, *Collection 2* served as a measure of data quality at the end of the visit. The average interval between the end of *Collection 1* and the start of *Collection 2* was 18.80 min ($\sigma = 8.83$). Given the unconstrained nature and long duration of the experiment, a large part of participants exhibited errors of varying magnitudes during *Collection 2*, many of which can be attributed mostly to eye tracker slippage. Besides the magnitude and systematic nature of the error for some participants, this attribution can also be inferred from the field camera videos (e.g., visible active adjustment by the user), gaze signal (e.g., abrupt systematic changes), and eye cameras (e.g., by using the eye corners as reference points for a particular gaze direction).

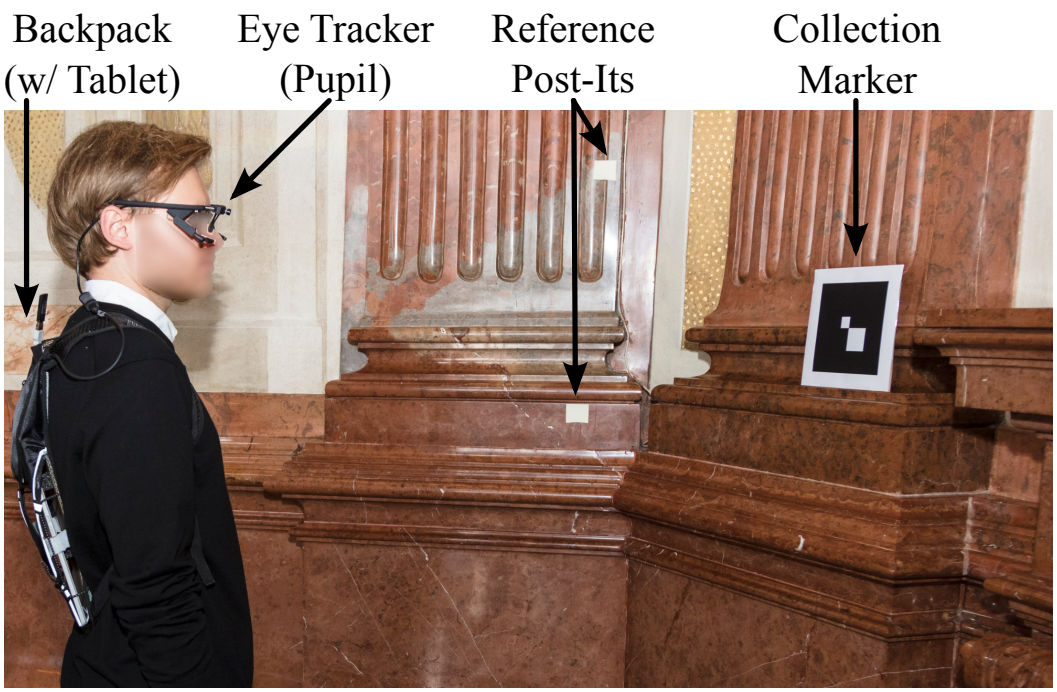


Figure 4.18: A (taller than average) participant during *Collection 1*. Two of the gaze-quality control Post-Its (to the right side of the collection marker) do not appear in the photo [26].

4.2.3.3 Evaluated Methods

As baseline, we employ a typical binocular polynomial fit (BPF) based on the 2D *pupil center* using the same configuration as described in Section 4.2.2.2. In other words, instead of the spherical coordinates ϕ and θ, the pupil center x and y coordinates are used. This is a similar approach to the original gaze estimation method employed during data collection and gives us a reference for a *non-slippage-robust* method. Both the baseline (BPF) and proposed method (Grip) used the same pupil input automatically detected using PuReST with default parameters. In case one of the pupils had a reported confidence below *0.66*, the methods automatically switched from binocular to monocular mode; this threshold was selected based on the recommendations from [6]. The methods were calibrated using data from *Collection 1* and evaluated on data from *Collection 2*.

Additionally, as a reference for a *temporal approach*, we ran the data through the 3D processing pipeline of the *Pupil Player* [124], which includes pupil tracking, calibration, and gaze estimation. The automatically detected collection marker center positions were loaded as *natural features* only for the world frame index in which they were detected – in contrast to the default [-5,5) index range. The calibration method was set to *natural features*, and the *calibration and mapping trim marks* were set to the start and end of *Collection 1* and *Collection 2*, respectively. For this evaluation, we employed *Pupil Player* version *1.9.7* and default parameters. Pupil also provides a confidence based on the detected pupils. We considered only gaze estimates with confidence above a certain threshold. For this

evaluation, we considered two distinct thresholds based on their documentation ($\text{Pupil}_{th=0}$ and $\text{Pupil}_{th=0.6}$): "In our experience useful data carries a confidence value greater than ≈0.6. A confidence of exactly 0 means that we don't know anything. So you should ignore the position data" [124].

4.2.3.4 Results

We discarded 31 subjects for various reasons: a) missing recording for one of the collections (11), b) nystagmus (7) or bright pupil effects (4) during gaze collections, and c) improper eye camera positioning (9), usually due to difficulties to place the eye tracker over glasses or adjust the cameras properly for participants with small faces or epicanthic folds. Average collection time for the remaining 78 participants was 20.92 s ($\sigma = 8.56$ s). Fig. 4.19 and Fig. 4.20 shows the spatial distribution of the collection marker center for *Collection 1* (calibration) and *Collection 2* (evaluation), respectively. It is worth reiterating here that this marker center was used as ground-truth for gaze position, consequently introducing some erroneous data in the evaluation – e.g., during blinks or distractions when the user was *not* gazing at the marker. Unlike typical eye tracking evaluations that are performed at a few predefined points, our evaluation encompasses a wide and well distributed range of points w.r.t. the field camera field of view, providing a realistic accuracy estimation across most of the device's operating range.

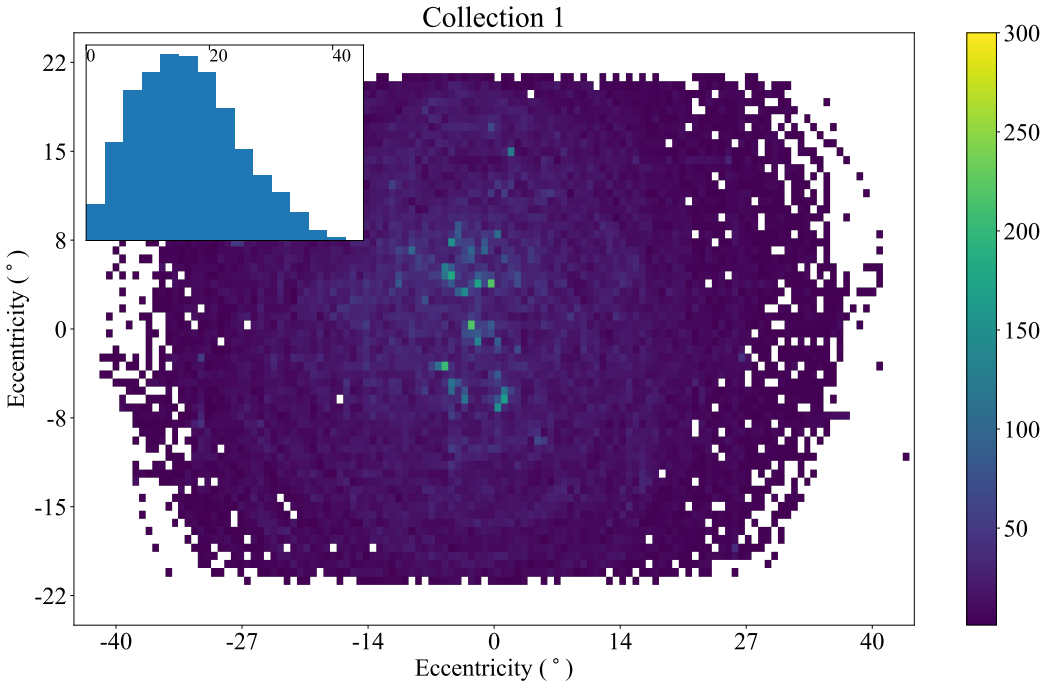


Figure 4.19: Calibration points spatial distribution as a 2D histogram w.r.t. the field camera field of view for all evaluated participants. The small 1D histogram at the top left shows the absolute distance-to-center distribution ($\mu = 16.41°$, $\sigma = 8.19°$) [26].

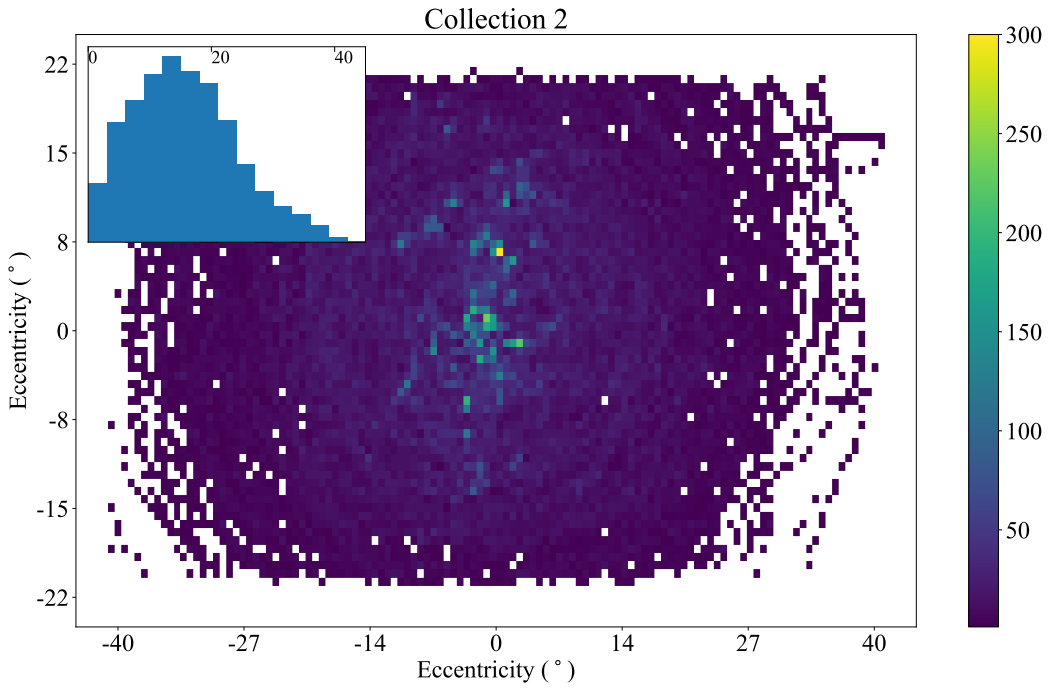


Figure 4.20: Evaluation points spatial distribution as a 2D histogram w.r.t. the field camera field of view for all evaluated participants. The small 1D histogram at the top left shows the absolute distance-to-center distribution ($\mu = 15.73°$, $\sigma = 8.43°$) [26].

We report our results in terms of *angular offset* between the automatically detected collection marker center and each method's 2D gaze estimate. During collection, blinks, distractions, and extrapolated[12] gaze points are expected, which can significantly impair average metrics. Thus, to give a better overview of the resulting gaze estimation quality, we report results not only in terms of participant average (*Mean*) but also quartiles (*Q1*, *Q2*, *Q3*). If calibration failed for a participant, these metrics were set to 90° to retain the same amount of participants across methods without perturbing the lower end of these metrics' distribution; this was the case only for 3 participants with Pupil.

Fig. 4.21 shows the distribution of these four metrics for all participants per evaluated method. This figure evidences Grip's superiority across participants, presenting significant improvements on average and for all quartiles. Relative to the non-slippage-robust gaze estimation (BPF), Grip exhibited a reduction of the average participant median (Q2) angular offset by 43.82% (5.66° vs 3.18°).

For approximately the first half of participants with smaller angular offsets, the difference between Grip and BPF is not that significant: These are likely participants for which the eye tracker position during *Collection 1* and *Collection 2* was very similar, thus BPF remains accurate. It is worth noticing that this *does not* imply that for such participants BPF is as accurate as Grip throughout the whole museum visit though. For instance, the eye tracker

[12] I.e., gaze points outside of the calibration region, which must be extrapolated instead of interpolated.

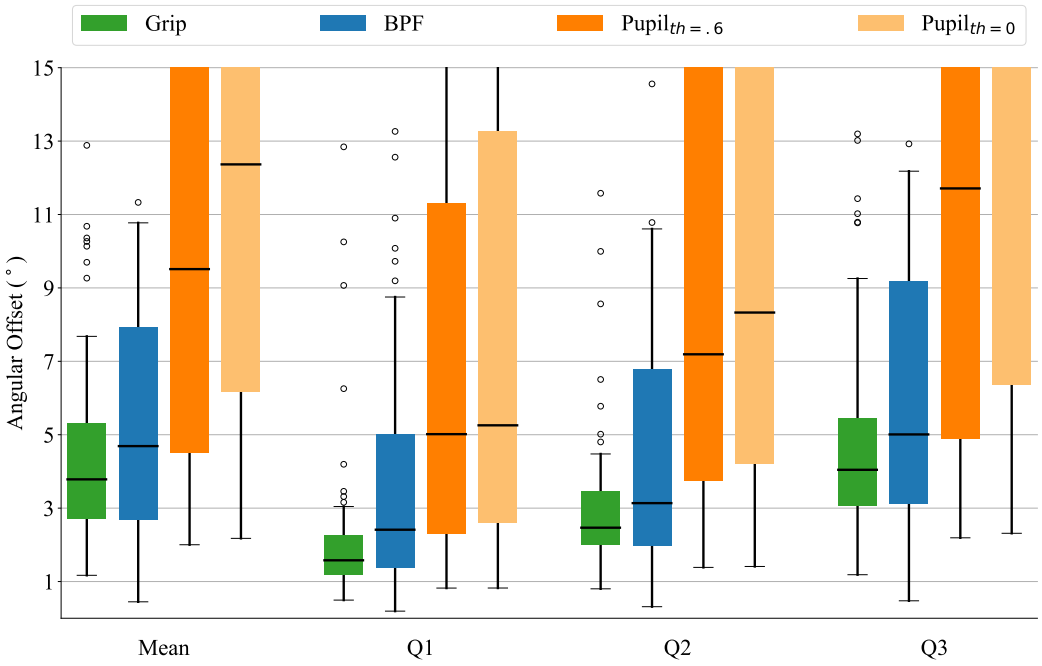


Figure 4.21: Distribution of per-participant gaze angular offset *Mean*, *Q1*, *Q2*, and *Q3*. Calibration was performed using *Collection 1* and evaluation on *Collection 2*. Whiskers subtend the $[0, 90]$ percentile range. E.g., considering the 25% samples with smallest offsets (Q1), 90% of the participants are within an accuracy of 3° using the proposed method (Grip), in contrast to 9° for the regular binocular polynomial fit (BPF) [26].

might have slipped soon after *Collection 1* but slipped back to a similar position just before *Collection 2*. The other half of participants (with larger angular offsets) showcases Grip's capability in cases where BPF is no longer reliable. For example, considering Q1 samples, 90% of participants are within an accuracy of 3° with Grip, in contrast to 9° for BPF. Similarly, at the more stringent Q3 requirement, Grip retains about 75% of participants within an accuracy of 5°, whereas BPF only retains about 50% of participants at the same accuracy requirement.

To give the reader an idea regarding camera positioning and image quality, examples of eye images for the 78 evaluated participants are shown in Fig. 4.22. Before including the results from Pupil in the discussion, it is important to note that its pupil tracking method has been shown to be significantly outperformed by the one we used (PuReST) [7][13]. Despite the data being recorded with Pupil hardware, we observed that for many participants Pupil's pupil detection did not work properly with default parameters[14], hence resulting in large

[13] In fact, it has been shown to be outperformed not only by PuReST, but also by other state-of-the-art pupil tracking algorithms [2], [7] such as Ellipse Selection by Candidate Filtering [2] (ESCaF), PuRe, ElSe, ExCuSe, and Świrski.

[14] Attempts at producing a better set of pupil detection parameters using *Pupil Capture*'s interface yielded unsatisfactory results.

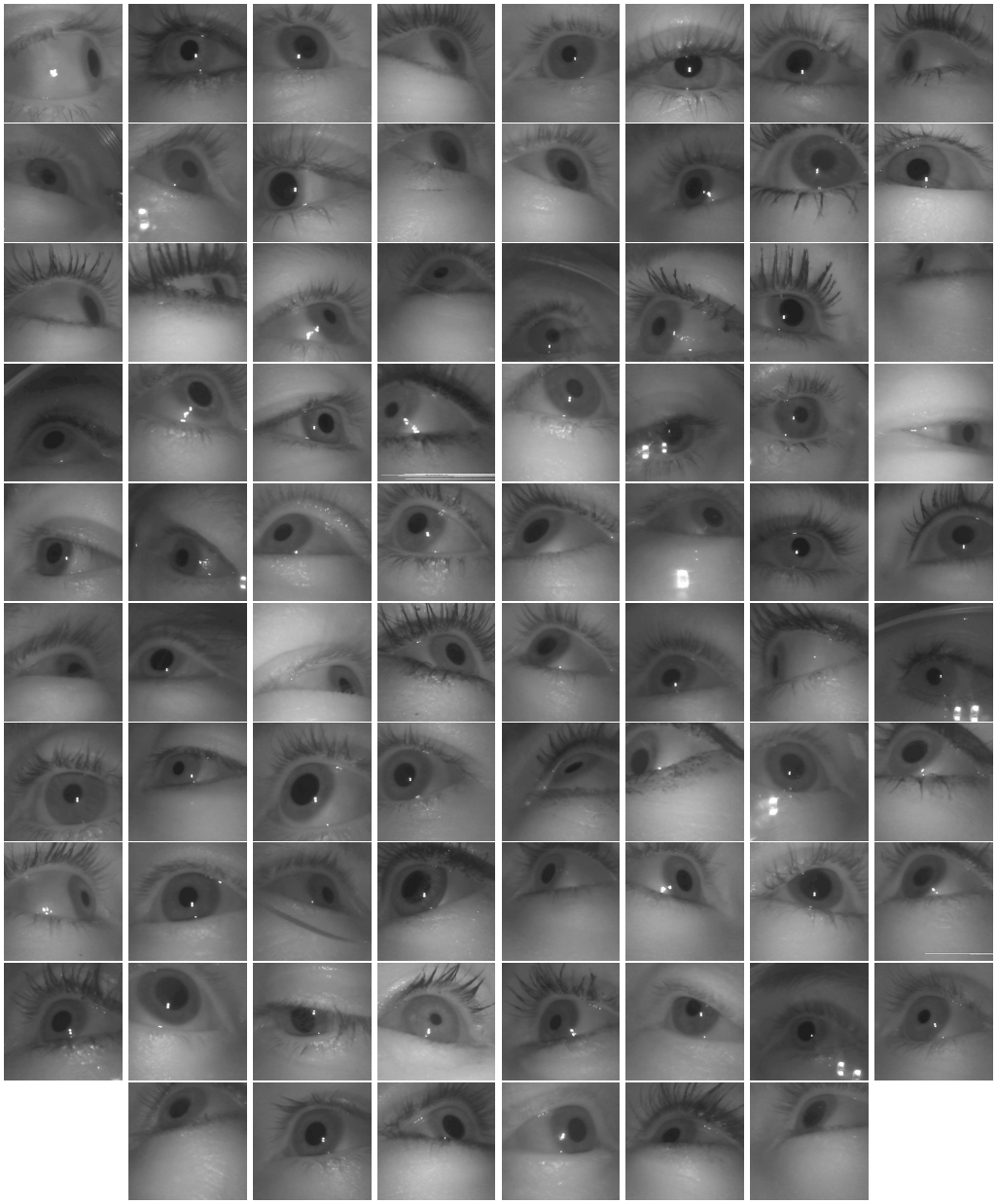


Figure 4.22: Example of eye images for the 78 evaluated participants. For each participant, we randomly selected the left or right eye recording and then randomly picked a frame from that entire eye recording. If the pupil was not visible in the image (e.g., because of a blink), another random frame was taken until the pupil was visible [26].

offsets for a large part of the participants. Therefore, we focus the following discussion only on the 25% of participants with smallest angular offsets for each metric, considering these as the participants for which Pupil best performed. As Pupil exhibited better results than

Pupil$_{th=0}$, we consider only the former in the following discussion. In this case, Grip still outperformed Pupil by large margins as shown in Table 4.1: On average, the angular offset was reduced by ≈1.8°, whereas the percentage of participants within the same accuracy as the 25% Pupil angular offset reaches ≈69%. In terms of computation time, Grip incurs negligible overhead: only a small fraction than the average computation time for pupil detection. In contrast, the temporal approach (Pupil employs non-linear optimizations when constructing the eye model, which might incur significant overhead depending on the number of samples used. Whereas this construction can be done asynchronously and is only required when the device slips, it could represent an issue for cases when the device is expected to move often and increases slippage compensation latency.

Metric	**25% Participants Angular Offset (°)**			**25% Pupil Equivalent (%)**	
	Grip	**BPF**	**Pupil**	**Grip**	**BPF**
Mean	2.71	2.69	4.51	69.23	48.72
Q1	1.19	1.36	2.31	78.21	47.44
Q2	1.99	1.96	3.75	85.90	53.85
Q3	3.05	3.11	4.90	65.38	48.72

Table 4.1: Angular offset for the best 25% participants for evaluated methods (columns 1-3), and percentage of participants covered by Grip and BPF (columns 4-5) considering a maximum angular offset equivalent to Pupil 25% (i.e., column 3). For example, considering the *mean*, 25% of participants with Pupil had angular offsets below 4.51°; in contrast, with the same angular offset threshold, Grip retained 69.23% of participants.

It is worth noticing the results for the participant with smallest angular offset (i.e., the lowest whisker), which can be considered the *best-case scenario* for each method. In this case, whereas Grip still outperforms Pupil it is outperformed by BPF. This highlights one of Grip's drawbacks: If the eye tracker remains stationary w.r.t. the eye center, Grip is excepted to be *less accurate* than BPF because the optical axis direction estimation introduces *additional noise* due to the 2D-to-3D unprojection (see Section 4.2.4).

Regarding overall accuracy, oftentimes eye-tracking studies only report on the accuracy of the gaze signal by reiterating a manufacturer provided number – usually below one degree – that does not correspond to the actual accuracy during real usage. For instance, real reported offsets are as large as 2° [47] even for tower-mounted eye trackers. As detailed in Section 4.2.4, head-mounted eye trackers have multiple sources of errors, which in practice reduce overall accuracy, especially when a thorough evaluation across most of the device's operating range is considered. It is worth noticing that some studies report average offsets of 3.6° even for high-grade eye trackers with multiple glint sources and reflection filters [227], whereas other authors consider accuracies within 5° as acceptable for single-glint eye trackers [131]. With Grip, our main goal is to provide slippage robustness while retaining realistic and usable accuracy. Accuracy requirements highly depend on the intended use of the collected gaze data. Grip managed to retain ≈75% of participants with an average and Q3 accuracy within ≈5° despite all the challenges involved. This level of

offsets suffice for many practical applications such as activity recognition [228], attention analysis [229], gaze-contingent audio guides [53], and assistance mode discrimination in human-robot shared manipulation [55]. As a reference for the reader, Fig. 4.23 illustrates various accuracy ranges w.r.t. the field camera view.

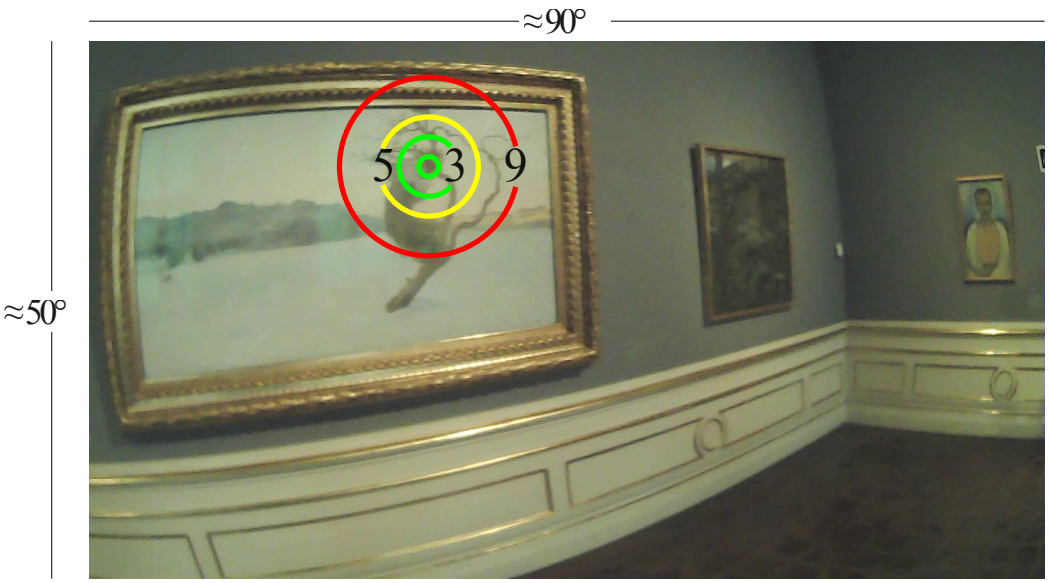


Figure 4.23: Error range overlaid on top of a field camera image (encompassing $\approx 90° \times 50°$) for reference. The circles' radii represent approximately 1°, 3°, 5°, and 9°. Figure best viewed in digital form [26].

4.2.4 Limitations

Despite promising results over a large array of participants in a pervasive and challenging scenario, Grip has some known limitations, which we discuss in this section.

First, Grip relies on the pupil outline. Thus, its performance is closely tied to the performance of the pupil tracker employed. This means that, if too much of the pupil outline is occluded, Grip might not be able to cope – e.g., for participants with droopy eye lids or epicanthic folds. In cases where the outline estimate does not match the real outline, the estimated optical axis (and thus gaze estimate) will be shifted. Moreover, subpar pupil outline contrast (e.g., due to thick lenses or bad camera focus) might result in a significant loss in precision: Although the pupil center might remain reliable, the pupil outline is unstable, causing the optical axis estimate to be volatile. This volatility could be lessened through signal filtering at the expense of partial loss of the non-fixational dynamics – e.g., using specialized Kalman filters [230] or general-purpose filters [185]. Furthermore, the outline requirement hinders its usage paired with pupil trackers that track only the pupil center – e.g., *DeepEye* [178] and *PupilNet* [64], [65].

Second, as previously mentioned, the unprojection of the 2D detected pupil ellipse to a

3D circle also adds noise to the optical axis estimate. Based on the naïve unprojection and simulations from [111], this appears to introduce a median error of ≈2.5° to the optical axis estimate. Moreover, the optical axis estimate is also affected by refractions from glasses [196] and the cornea [216]. Nonetheless, the resulting deviations are intrinsically learned during the calibration to some extent. Thus, contrasting these measurements with our results is not straightforward due to the non-geometric gaze mapping we employed and other sources of errors stemming from the realistic and out-of-the-lab scenario. For instance, head-mounted devices tend to exhibit angular offsets due to other factors such as *parallax error* [231], and changes in *pupil dilation*, which can result in inaccuracies up to 1.5° [47]. In particular for the data reported in Section 4.2.3, *Collection 1* (calibration) and *Collection 2* (evaluation) were performed in distinct rooms with very distinct illuminations, which can cause large discrepancies in pupil size.

Third, the pupil unprojection also introduces a further complication: When the optical axis points towards the camera center such that the projection of the 3D pupil approximates a 2D circle in the image plane, the optical axis estimate is ill defined [113], [232]. By construction, this is a gaze position that rarely occurs in practice (unless the user is gazing close to the eye camera). Nevertheless, we recommend placing the eye cameras eccentric to the eye primary position [233]. This complication might also present an incompatibility with eye trackers that make use of hot mirrors to remove parallax (e.g., [234]), as the primary eye position might often correspond with the ill-defined unprojection region.

4.2.5 Conclusion

In this section, we have proposed and evaluated Grip, a slippage-robust and glint-free gaze estimation method for head-mounted eye trackers. The proposed method combines a geometrical approach to derive a slippage-robust input feature for a non-geometrical gaze mapping function, thus making it a *hybrid* method. Grip was evaluated with data from 78 participants of a pervasive and unconstrained eye tracking study, showing significant improvements in multiple metrics when compared to a regular binocular polynomial fit (BPF) and a temporal eye-model-based gaze estimation approach (Pupil). Despite the challenging scenario, Grip achieved sufficient accuracy for multiple practical applications, retaining 90% of participants with a median gaze offset below 4.57°. Relative to the non-slippage-robust gaze estimation (BPF), Grip exhibited a reduction of the average participant median angular offset by 43.82% (5.66° vs 3.18°).

In order for the proposed method to be able to instantaneously compensate for slippage, we have relaxed the eye-center estimation requirement. Consequently, we did not explore the application of more complex models designed to improve optical axis estimate accuracy – e.g., relying on temporal constraints to improve gaze estimation when the detected pupil is not accurate [111] or compensating for corneal refraction [216]. To take advantage of such methods, we intend to explore adaptive hybrid approaches that combine instantaneous and temporal gaze estimation approaches in the future. We envision an approach that is able to detect instabilities in the eye pose and automatically adapt by switching between the instantaneous and temporal approaches. In order to realize this vision, we also intend on investigating the temporal eye model construction design space and elucidate the issues

raised in Section 4.2.1.

Finally, we intend to investigate alternative methods for the estimation of the optical axis that are potentially more robust to occlusions of the pupil outline – such as machine-learning approaches [235]. These approaches can then be employed as a simple drop-in replacement for the current ellipse unprojection method (described in Section 4.2.2.1) integrated into *Grip*.

5 Eye Movement Identification

> "*We see in order to move; we move in order to see.*"
>
> —William Gibson

The human visual perception involves mainly six types of eye movements: fixations, saccades, smooth pursuits, optokinetic reflex, vestibulo-ocular reflex, and vergence [236]. The automatic and correct identification of these eye movements based on the raw eye-position signal is critical for several applications, such as driver's activity recognition [41] or detection of hazard perception during driving [237]), marketing applications, and HCI [238].

Initially, eye-tracking research restrained head movements and employed *static stimuli*, such as images and text. In this scenario, the only relevant movements considered were *fixations* (in which the eyes are relatively still) and *saccades* (rapid transitions from one fixation point to another); thus, early algorithms for the automatic classification of eye movements focused on segregating only between these two movements. Nowadays, there is an increasing interest in using *dynamic stimuli* (e.g., video clips) [239], where an object of interest moves through the subject's field of view and is kept on the fovea, producing a fluent eye motion – which we denominate a *smooth pursuit*. It is worth noticing that during these pursuits, minor eye movements such as tremors and microssacades exist, albeit these minor movements do not show on low-resolution eye-tracking data.

The presence of smooth pursuits disturbs the performance of established event classification algorithms since these pursuits end up spread over the two classification classes. Moreover, they also provide valuable information on subject's health and behavior; for instance, smooth pursuit impairment and dysfunction have been linked to mental illnesses, such as schizophrenia [240] and Alzheimer's disease [241]. Thus, an automatic and efficient algorithm to distinguish between fixations, saccades, and smooth pursuits is paramount for eye-tracking research involving dynamic stimuli. Furthermore, some of the possible applications must be in the form of embedded systems (e.g., driving assistance) and impose real-time, processing, and energy consumption constrains on the eye-tracking system. To meet these constraints, typically eye trackers with a lower sample rate are used. Consequently, such an algorithm must not only work in real-time, but also be able to deal with the low resolution arising from such eye trackers.

In this chapter, we propose a novel algorithm for ternary classification of oculomotor events. Our main contributions are:

- We propose the Bayesian Decision Theory Identification (I-BDT) algorithm to identify fixations, saccades, and smooth pursuits in real-time for low-resolution eye trackers. Additionally, the algorithm operates directly on the eye-position signal and, thus, requires no calibration.

- The proposed algorithm is evaluated relative to manual annotation by a domain expert, and performance is measured in terms of recall, precision, specificity, and accuracy; on average, the proposed algorithm scores above 90% on all metrics.
- I-BDT's performance is compared to that of a state-of-the-art algorithm (Velocitiy and Dispersion Threshold Identification (I-VDT)), showing a significant improvement in terms of average score and variability.
- Additionally, we openly provide a *MATLAB* implementation for the I-BDT algorithm as well as the annotated datasets used for evaluation at `www.ti.uni-tuebingen.de/perception`.

5.1 Related Work

In 1991, Sauter et al. [242] proposed using a Kalman filter coupled with a χ^2-test to separate saccades from other eye movements. This approach was later extended as the Attention Focus Kalman Filter (AFKF) by Komogortsev and Khan [243], using velocity and temporal thresholds to separate fixations from smooth pursuits. Similarly, several methods use a simple *velocity threshold* to isolate saccades, followed by a second step to distinguish between fixations and smooth pursuits. Such algorithms are typically identified by a name following the pattern *I-V**. Komogortsev and Karpov [244] proposed to distinguish between the remaining movements through a *second velocity* threshold (Velocitiy and Velocity Threshold Identification (I-VVT)) and through a *dispersion* threshold combined with a temporal window (Velocitiy and Dispersion Threshold Identification (I-VDT)). Berg et at. [245] proposed analyzing the ratio between first and second principal components to identify smooth pursuits (Principal Component Analysis Identification (I-PCA)) on the intuition that fixations would have a ratio close to one. Lopez [246] started a subgroup that uses the *movement pattern* to identify smooth pursuits, hence the shared I-VMP prefix; in [246], Lopez used the standard deviation of the movement directions in a time window to isolate fixations (Velocity Movement Pattern Standard Deviation Identification (I-VMPStd)). Larsson [247] used a Rayleigh test to identify smooth pursuits by rejecting the hypothesis of uniformity of inter-sample vectors around the unit circle (Velocity Movement Pattern Rayleigh Identification (I-VMPRay)); more recently, this algorithm was extended with four different spatial features (dispersion, consistent direction, positional displacement, and spatial range) in [239].

Tafaj et al. [248] used a Bayesian Mixture Model based on the Euclidean distance between sequential points to discern fixations from saccades, which was later extended in [133] with a principal component analysis similar to I-PCA to identify smooth pursuits. This method is called the Bayesian Mixture Model Identification (I-BMM). Vidal et al. [249] defined a set of shape features, whose expected range is derived from training data. A k-nearest neighbors classifier ($k = 3$) is then used to isolate smooth pursuits from other movements.

As illustrated in Fig. 5.1, the above methods fall mainly into two classes: *threshold-based* and *probabilistic* methods. While threshold-based algorithms tend to be simpler to

implement, their major drawback is that they usually depend on the eye movements being clearly discernible from each other. On the other hand, probabilistic methods work based on softer decision rules in the form of probabilities, making them more flexible. Hybrid methods combine insights from physiological limits to define clear thresholds (e.g., only during saccades the eyes reach velocities above 100 °/s [212]) with a probabilistic approach in other cases. I-BDT, the method proposed in this chapter, falls into the probabilistic group.

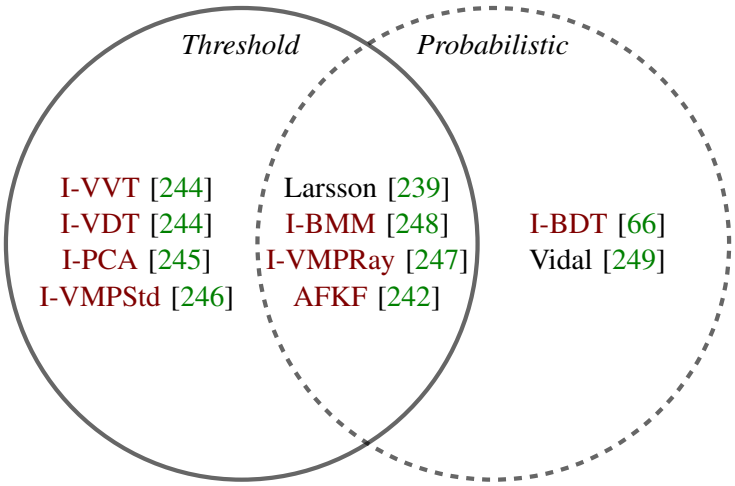


Figure 5.1: Algorithms for the automatic identification of smooth pursuits according to a broad classification based on their underlying mechanisms. The algorithm proposed on this work (I-BDT) falls within the probabilistic group [66].

Additionally to the previously mentioned algorithms, recently several authors have applied machine-learning methods for eye movement identification. For instance: Zemblys [250] investigated ten different machine learning methods applied to eye movement detection, Hoppe and Bulling [251] proposed a shallow CNN for ternary eye movement identification, Zemblys et al. [252] investigated the performance of random-forests for binary event detection, Bellet et al. [253] achieved human-level saccade detection using deep learning, and Zemblys et al. [254] proposes a CNN to augment hand-coded data to create large training data sets to train a second CNN, which is able to classify raw eye-tracking data into fixations, saccades, and post-saccadic oscillations.

It is worth noticing that most previous work has focused on eye trackers with high sampling rates (i.e., above 250 Hz). However, in dynamic scenarios where a non-intrusive head-mounted eye tracker is required (e.g., driving assistance), such high sampling rates are not available. Currently, mostly head-mounted eye trackers present an upper limit of 60 Hz for binocular tracking – e.g., Dikablis Pro, SMI Glasses 2, ASL H7 Optics, Tobii Pro Glasses 2. The exception is SR Research's EyeLink II, which has a binocular sampling rate of 500 Hz. Despite its clear advantage in temporal resolution, this eye tracker is rather intrusive, occupying a large part of the subject's field of view; for comparison, EyeLink II's eye cameras measure each approximately 5 cm × 5 cm × 1 cm whereas Dikablis Pro's eye cam-

eras measure approximately only 2.5 cm × 2 cm × 1 cm, resulting in a volume difference of five times. Furthermore, higher sampling rates also incur in higher power consumption, which might hinder their usage in embedded scenarios.

5.2 Bayesian Decision Theory Identification

5.2.1 Problem Statement

Let $S = \{s_i | 1 \leq i \leq N\}$ be a set of N temporally ordered tuples, each containing two-dimensional pupil position estimates (x_i, y_i) and a timestamp (t_i) generated by an eye tracker (i.e., an eye-tracker protocol). The problem, thus, is to classify all periods between two subsequent tuples according to the set of possible events $E = \{fix, sac, pur\}$, where fix, *sac*, and *pur* stand respectively for fixation, saccade, and smooth pursuit.

5.2.2 Model

In this work, we propose a Bayesian decision theory approach to solve the stated problem based on a pair of features derived from S. In other words, given some data D, we are interested in defining the *likelihoods* $p(D|e)$ and *priors* $p(e)$ for each event $e \in E$ in order to calculate the *posteriors* $p(e|D)$ of these events. Following the naming convention from [244] and [255], we will hereby refer to this method as the Bayesian Decision Theory Identification (I-BDT) algorithm.

The first feature derived from S is the estimated eye speed (v_i) between two subsequent tuples, defined as

$$v_i = \frac{\sqrt{\Delta x_i{}^2 + \Delta y_i{}^2}}{\Delta t_i}, \tag{5.1}$$

where $\Delta x_i = x_i - x_{i-1}$, $\Delta y_i = y_i - y_{i-1}$, and $\Delta t_i = t_i - t_{i-1}$.

The second derived feature is the movement ratio r_i over the window $W_i = \{v_j | i - N_w < j \leq i\}$ of the latest N_w tuples. For simplicity, we define it as the amount of non-zero eye speed estimates relative to the window size, conveying the idea that the more movement in the window, the more likely a smooth pursuit is; thus,

$$r_i = \frac{1}{N_w} \sum_{v_j \in W_i} [v_j > 0] = \frac{1}{N_w} \sum([W_i > 0]), \tag{5.2}$$

where $[...]$ is the Iverson bracket notation [256] given by

$$[X] = \begin{cases} 1 & \text{if X is true;} \\ 0 & \text{otherwise.} \end{cases}$$

It is paramount to note that this feature's definition is heavily dependent on the eye tracker used to record the data and its temporal and spatial resolution; zero speed may not be an appropriate representation for fixations. Nevertheless, the intuition behind this feature is that fixations exhibit little continuous movement, saccades are brief and usually separated

by fixations, and smooth pursuits tend to exhibit continuous movement during larger periods of time (see Fig. 5.2). Therefore, r_i should be a good smooth pursuit indicator if an

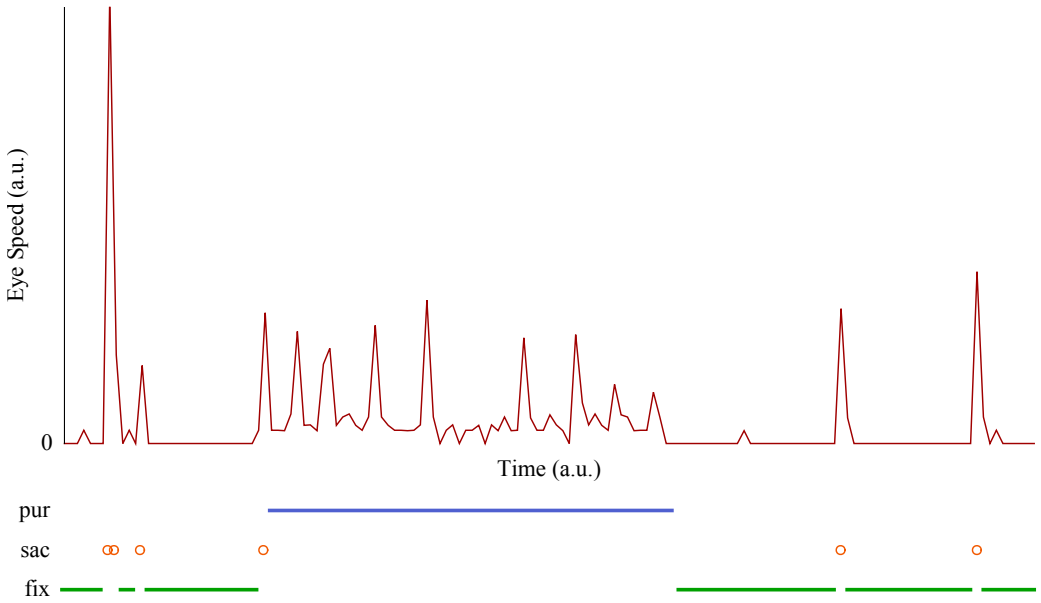


Figure 5.2: Eye speed compared to manual classification by a domain expert. Fixations (fix) tend to be mostly still, with only few deviations due to micro eye movements and measurement noise, whereas saccades (sac) result in brief spikes in the eye speed signal. On the contrary, smooth pursuits (pur) show a distinct speed pattern during a longer period of time [66].

adequate window size is chosen; this time window should be large enough to encompass the maximum saccade duration, otherwise misclassification of saccades as pursuits may be exacerbated. In our model, we use this feature directly as the smooth pursuit likelihood, i.e.,

$$p(r_i|pur) = r_i. \tag{5.3}$$

Once a smooth pursuit has started, it tends to continue for an arbitrary period; thus, this should be reflected in one's belief before any evidence is taken into account. For this reason, we model the smooth pursuit prior as the mean of previous smooth pursuit likelihoods (i.e., the set $L_i = \{p(r_j|pur)|i - N_w < j < i\}$) such that

$$p(pur) = \frac{1}{N_w - 1} \sum_{p(r_j|pur) \in L_i} p(r_j|pur). \tag{5.4}$$

Naturally, the joint probability of priors must sum to one. With no further evidence, we do not have reason to believe either fixations or saccades are more probable, and, thus, we divide the remaining joint prior probability equally between these movements such that

$$p(fix) = p(sac) = \frac{1 - p(pur)}{2}. \tag{5.5}$$

It is worth noticing that if information on the task being performed by the subject is available, one could improve these priors based on the duration of the current event. For instance, imagine a task characterized by fixations with a relatively constant duration: after a first fixation is found, the following events are likely to be fixations until the average fixation duration is reached. At this point the next event becomes less and less probable to be a fixation. Such behaviour could be taken into account by adjusting the priors.

The fixational and saccadic likelihoods are deemed to be dependent only on the current eye speed (v_i) feature. This feature can be used to reliably separate high-speed saccades from other events as it has been shown that no other event can reach a velocity higher than V_{sac}, estimated to be around $100\,^\circ/\mathrm{s}$ [212]. However, the speed spectra of different eye movements overlap for lower velocities. Nonetheless, it is intuitive that velocities closer to zero are more likely to stem from fixations whereas velocities closer to V_{sac} are more likely to stem from saccades. In fact, [248] have shown that saccades and fixations can be represented by a mixture model of two Gaussian distributions based on the distance between sequential points – one Gaussian generating fixations, and another one generating saccades. Therefore, we assume the eye speed feature to also be generated by two such Gaussian distributions. Intuitively, saccade likelihood should be at its maximum for speeds above V_{sac}. Ideally, fixations would exhibit zero speed; however, as they typically include small movements, such as microsaccades and tremors, there is a threshold speed V_{fix} that encompasses these combination of movements. Thus, fixation likelihood should be at its maximum for speeds below V_{fix}. In the interval between these thresholds, we assume the likelihood to be generated by two Gaussian[1] distributions, one centered around V_{fix} and the other around V_{sac} (see Fig. 5.3). Thus,

$$p(v_i|fix) = \begin{cases} N(V_{fix}|V_{fix},\sigma_{fix}) & \text{if } v_i < V_{fix} \\ N(v_i \quad |V_{fix},\sigma_{fix}) & \text{if } v_i \geq V_{fix} \end{cases}, \tag{5.6}$$

and

$$p(v_i|sac) = \begin{cases} N(v_i \quad |V_{sac},\sigma_{sac}) & \text{if } v_i < V_{sac} \\ N(V_{sac}|V_{sac},\sigma_{sac}) & \text{if } v_i \geq V_{sac} \end{cases}. \tag{5.7}$$

Having defined the priors and likelihoods for all events, we can calculate the posterior for each event $e \in E$ given the data $D = \{v_i, r_i\}$ using Bayes' Theorem; thus,

$$p(e|D) = \frac{p(e)p(D|e)}{p(D)}, \tag{5.8}$$

and the period is classified as the event with highest posterior probability. Here, $p(D)$ is merely a scaling factor that guarantees that the sum of the posterior probabilities sum to one. It is worth noticing that this model can be extended to include other eye movements in the future by determining their priors and likelihoods, and taking these into account when computing $p(D)$.

[1] Denoted as $N(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 / 2\sigma^2}$

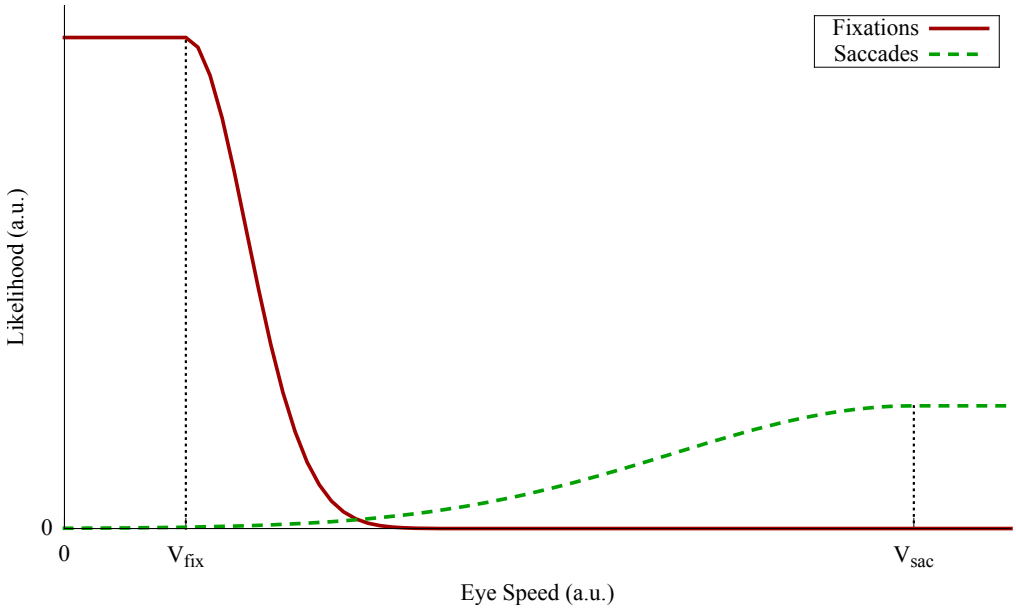


Figure 5.3: Resulting fixational and saccadic likelihoods based on the eye speed feature (v_i) [66].

5.3 Experimental Setup

5.3.1 Dataset

To evaluate the proposed algorithm, we designed an experiment to cover a wide range of induced as well as natural eye movements. The induced movements are characterized in Table 5.1.

Movement	**Amplitude**[a]**/Radius**[b] (°)	**Velocity** (°/s)
Saccade[a]	6, 11, 14	—
Straight Pursuit[a]	6, 12, 22, 28	10, 20, 30
Circular Pursuit[b]	6, 8, 14	18, 25, 44

Table 5.1: Induced movements used within the experiment. Degrees are expressed in terms of visual angle. Straight pursuit amplitudes and velocities were combined such that their durations were within 0.4 and 2 seconds to account for subject latency while keeping pursuit duration realistic. Circular pursuits were conducted at a constant angular velocity of 180°/s. Pursuits were separated from other movements by one second fixations. Saccades were separated from each other by fixations of 0.75 seconds. The directions of the movements were chosen randomly and differ per subject [66].

Prior to the recording, each user was shown a tutorial with detailed on-screen instructions and examples of movements for each class in Table 5.1. Four datasets were recorded per subject, and all datasets had a common beginning: first, four dots were shown at 15° of visual angle diagonally from the screen center for five seconds (Fig. 5.4a); subjects were

instructed to look at these stimuli at will. During this period natural saccades and fixations are collected; saccades of ≈ 20 and 30° of visual angle were expected, separated by fixations of arbitrary duration. Afterwards, a single dot appeared at the screen center for two seconds (Fig. 5.4b); subjects were instructed to focus on and follow this target. The subsequent movements differ per dataset and are listed in Table 5.2.

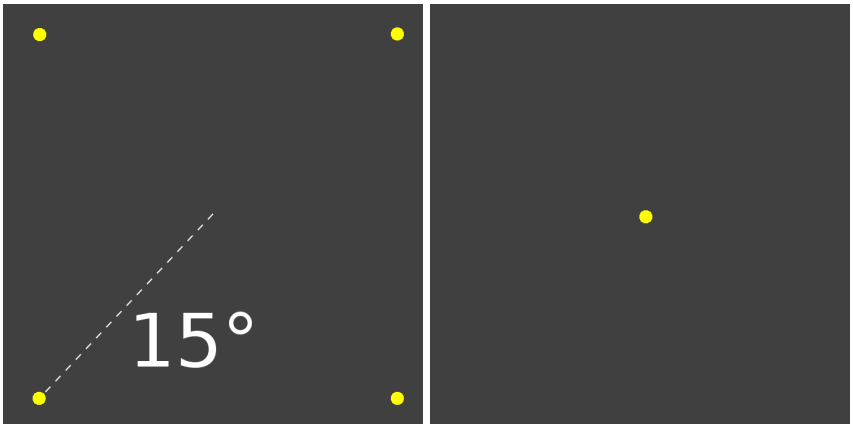


Figure 5.4: Common stimuli at the beginning of each dataset. In this figure, the color of the targets was changed from red to yellow to facilitate visualization [66].

Dataset	Movements
I	Fixations, saccades, and all possible straight pursuits.
II	Fixations and saccades. No pursuits.
III	Fixations, saccades, and all circular pursuits.
IV	Fixations, saccades, straight and circular pursuits.

Table 5.2: Movements distribution per dataset [66].

Targets were red dots (with a width of 1° of visual angle) on a dark gray background displayed using *MATLAB* (r2013a) and the *Psychtoolbox* (3.0.12) [257] on a Windows 64-bit machine. Subjects' heads were supported by a chin rest at a distance of 300 mm from a *Samsung SyncMaster 2443BW*[2] color display unit. Ocular dominance was determined using the Miles test, and data was collected only from the dominant eye using a *Dikablis Pro* eye tracker (eye images of 384x288 pixels with a 30 Hz sampling rate) and *EyeRec* [70] (1.2.2) running the ExCuSe [3] pupil detection algorithm on a distinct Windows 64-bit machine. To avoid gaze estimation noise and calibration requirements, we use the pupil position signal as input; as such, no calibration step was performed. An unjittering function was applied to this input prior to processing to remove obvious jitter artifacts (e.g., one sample

[2] Width: 520 mm. Height: 320 mm. Resolution: 1920x1200 pixels. Screen refresh rate: 60 Hz. Luminance: 0.08 cd/m^2.

spikes [258]). Six adult subjects (age: $\mu = 31.50$, $\sigma = 2.59$ years; 4 males, 2 females) took part in the experiment. Eye location relative to the eye tracker varied greatly between subjects to exacerbate differences in the input signal and stress the algorithm (see Fig. 5.5). Two of the subjects wore corrective glasses for myopia (-13 dpt and 1.5 dpt).

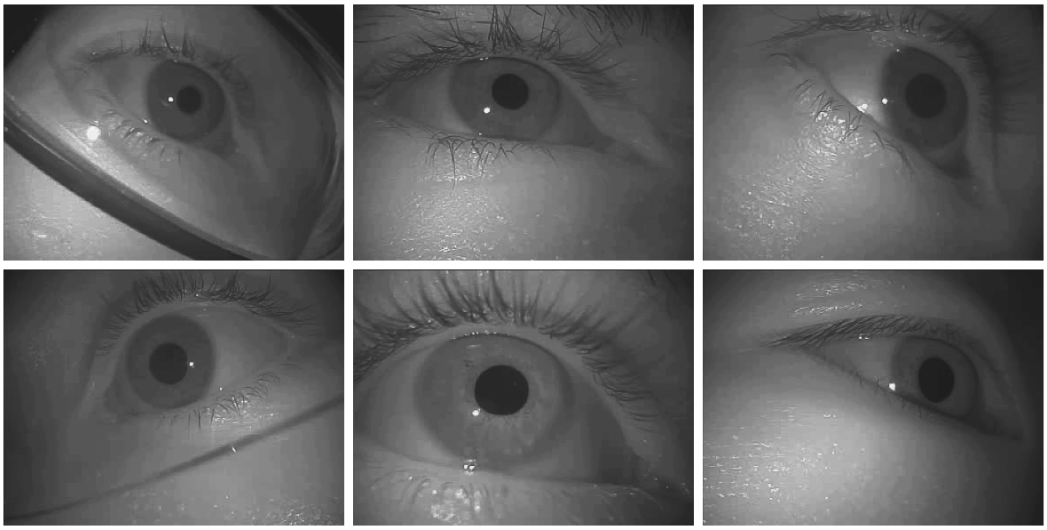


Figure 5.5: Example of eye location relative to the eye tracker during experiments. Note the distinct proximities, positions, and rotations [66].

5.3.2 Baseline and Metrics

The collected data was manually classified by one domain expert in order to identify data that is not coherent with the stimulus information, e.g., because the subject did not follow the stimulus as instructed. This manual classification was used as the ground truth.

Nonetheless, it is worth noticing that the manual classification is a subjective task, especially for data with a temporal resolution where a measurement period may contain a mixture of the end of a saccade and the beginning of a fixation. For this reason, we provide our annotated dataset openly to allow for review and potential improvements. Initial corrective saccades during pursuit onset were classified as saccades, whereas catch-up saccades during pursuit were classified as smooth pursuits. Fixation classifications encompass small eye tracker noise, drift and microsaccades. Blinks, partial pupil occlusions, and pupil detection failures were marked as noise and are ignored for performance evaluation; these represent $\approx 1.76\%$ of samples.

Overall, 18,682 fixations, 1,296 saccades, and 4,143 smooth pursuits were classified. Performance is measured through four metrics per movement class, namely: recall $\left(\frac{TP}{TP+FN}\right)$, precision $\left(\frac{TP}{TP+FP}\right)$, specificity $\left(\frac{TN}{TN+FP}\right)$, and accuracy $\left(\frac{TP+TN}{TP+FP+TN+FN}\right)$, where TP, FP, TN, and FN stand for True Positive, False Positive, True Negative, and False Negative, respectively. Moreover, we compare the performance of the proposed algorithm to that of the I-VDT algorithm as implemented by [244], [259]; I-VDT was chosen as it can be easily

adapted to perform online classification on low-resolution eye trackers, and because it has been shown to exhibit a competitive performance with smaller variability relative to other algorithms [244], [260]. Additionally, we also provide *Cohen's Kappa* [261] values for the overall classification agreement between the algorithms and the domain expert to account for agreement merely due to chance.

5.3.3 Algorithm's Parameters

I-BDT: We have chosen a window size to fit 1.5 times the maximum saccade duration (80 ms [47]). This value was chosen to fill the minimum size requirement while keeping the window size to a minimum, thus minimizing the duration of the pursuit detection onset. For each subject-dataset pair, the Gaussian distributions parameters are derived from an approximately 15 s of data to demonstrate an online training procedure. Initially, the Expectation-Maximization algorithm was used to derive a mixture of two Gaussian distributions based on speed samples from this period (with the smallest positive scalar supported by the platform added to the estimated covariance matrices to ensure they were positive definite). The parameters of the Gaussian distribution with the highest mean are used as parameters for saccades in Equation (5.7). However, due to the low resolution of the eye tracker, the Gaussian distribution with the smaller mean is heavily biased towards zero and does not describe fixations adequately; we chose instead to derive the parameters for Equation (5.6) based on the inherent eye tracker resolution: the minimum dispersion between two samples larger than zero divided by the inter-sample period was taken as V_{fix}, and σ_{fix} was set to $\frac{2}{3}V_{fix}$ such that $\approx 99.7\%$ of the distribution values lie within the interval $[0, 2V_{fix}]$. Furthermore, this low resolution also leads to speed samples with null value during slow smooth pursuits; thus, we have redefined Equation (5.2) as

$$r_i = \frac{1}{N_w}\sum(smooth([0 < W_i < V_{sac}])) \tag{5.9}$$

where the *smooth* function applies the following logical substitutions over the entire temporal window

$$\begin{cases} 1x1 \rightarrow 111 & \text{always} \\ 1xx1 \rightarrow 1111 & \text{if sample } i-1 \text{ was classified as a smooth pursuit} \end{cases},$$

with x representing a *don't care* term. In other words, r_i tolerates a single isolated null speed sample if not currently in a smooth pursuit; otherwise, it is more lenient and tolerates up to two isolated null speed samples. This redefinition implies the temporal window must include at least four samples.

I-VDT: In order to get I-VDT's optimal performance, we give it an advantage by defining pareto-optimal thresholds that maximize Z_1 scores based on the ground truth. First, the Z_1 score for saccade classification is evaluated for all the inter-sample velocities that can be derived from the eye-tracker protocol; the velocity that maximizes this score is chosen as the *velocity threshold*. Second, the minimum fixation duration is derived from the ground truth and is used as the *temporal window size threshold* (generally around 100 ms). Lastly,

fixing the previously defined thresholds, the Z_1 score for pursuit classification is evaluated for all the inter-sample dispersions that can be derived from the eye-tracker protocol; the dispersion that maximizes this score is chosen as the *dispersion threshold*. If the ground truth contains no pursuits, the Z_1 score for fixation classification is used instead.

5.4 Experimental Results

First we look at an overview that encompasses all datasets and eye movements to show the overall performance of the proposed algorithm. Afterwards, we analyze our results for separate movements and datasets to provide a comprehensive understanding on the I-BDT behavior. Results are reported using *boxplots* (a box is drawn between the first and third quartiles, a horizontal line represents the median value, and whiskers extend to the minimum and maximum values). Ideally, the value for all metrics should be as close to one as possible. The method introduces no delays, and the average time required to classify a new sample was 0.44 ms, thus attesting for the real-time capabilities of the proposed approach.

5.4.1 Overall Results

Table 5.3 and Fig. 5.6 indicate the high performance of the I-BDT algorithm. It is clear that not only I-BDT presents better scores throughout all metrics relative to I-VDT, it also exhibits less variability. Moreover, the high Cohen's kappa score indicates that the inter-rater agreement between expert and algorithm was not due to chance. Note that, in its current form, the algorithm seems to favor precision instead of recall; this is true for smooth pursuits (which can sometimes be misclassified as fixations, specially during onset) and saccades (which are rarely misclassified as smooth pursuits); however, fixations are very seldom misclassified but tend to encompass other movements in its class more often.

	I-BDT	**I-VDT**
Recall	$\mu = 91.42\%, \sigma = 9.52\%$	$\mu = 87.67\%, \sigma = 14.73\%$
Precision	$\mu = 95.60\%, \sigma = 5.29\%$	$\mu = 89.57\%, \sigma = 8.05\%$
Specificity	$\mu = 95.41\%, \sigma = 7.02\%$	$\mu = 92.10\%, \sigma = 11.21\%$
Accuracy	$\mu = 96.95\%, \sigma = 2.54\%$	$\mu = 94.65\%, \sigma = 4.50\%$

Table 5.3: Average algorithm performance per dataset per subject per movement class ($n = 3 \times 6 \times 3 + 1 \times 6 \times 2 = 66$) [66].

5.4.2 In-depth Analysis

We start our in-depth analysis by looking at the algorithms performance per dataset for fixations. Fig. 5.7 shows that the algorithm scores highly for the recall and precision metrics for this class, consistently above 90%, and generally above 95%. However, since fixations are the prevalent class in all datasets, false positives are drowned in the larger number of

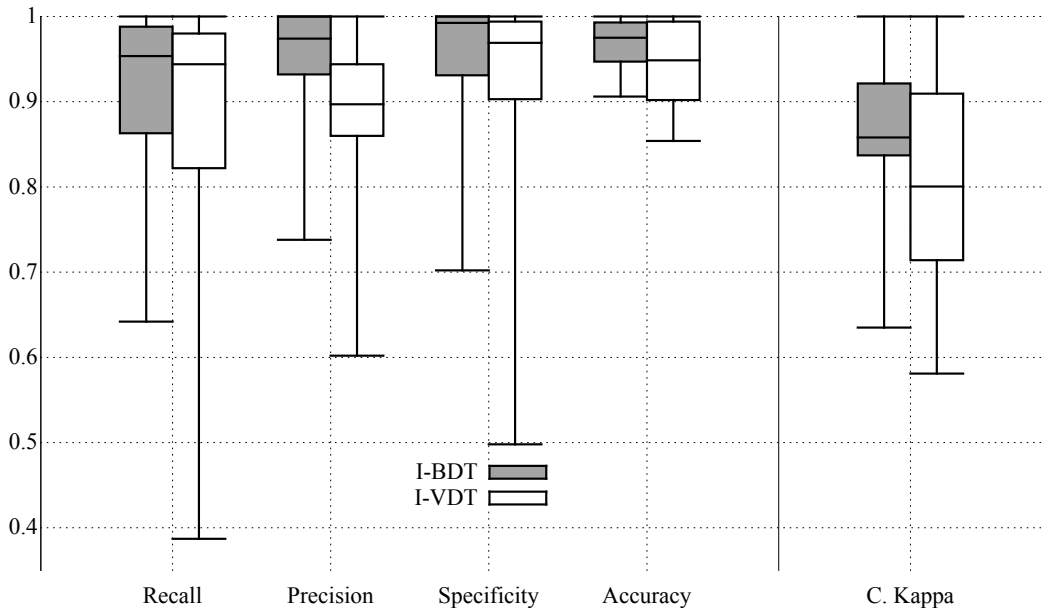


Figure 5.6: Overall algorithm performance. Recall, precision, specificity, and accuracy per dataset per subject per movement class ($n = 3 \times 6 \times 3 + 1 \times 6 \times 2 = 66$). Cohen's kappa per dataset per subject ($n = 4 \times 6 = 24$) [66].

true positives; as a result, it is of great importance to look at the specificity when evaluating fixation classification performance. In this case, I-BDT scored above 80% reliably. It is plain that the specificity for dataset II is well above the others, which suggests that the false positives are mostly misclassified smooth pursuits. This is supported by evidence that slow smooth pursuits are the ones being misclassified; specificity for dataset I is almost consistently lower than for dataset III and IV, presumably due to dataset I always including the slowest smooth pursuits. Likewise, specificity for dataset IV is only sometimes lower than that of dataset III because dataset IV only randomly includes the slowest smooth pursuits.

As can be seen in Fig. 5.8 and Fig. 5.9, specificity for both saccades and smooth pursuits classification is persistently high ($>$ 95%). However, similarly to how precision can be misleading for the performance evaluation of fixation classification, specificity can be deceptive for saccades and smooth pursuits classification as false positives get masked by the larger amount of true negatives. Thus, we analyze saccade and smooth pursuit classification through the recall and precision metrics.

Fig. 5.8 shows that saccade classification is very precise ($>$ 90%) in the majority of cases. While the proposed algorithm also displayed a good recall (mostly above 80%), it is clear that some saccades are being misclassified; these are usually saccades surrounded by noise, which the algorithm ends up interpreting as a high movement ratio and, thus, classifying as smooth pursuits. This effect also leads to the I-VDT algorithm outperforming I-BDT for saccade recall for dataset II. Since this dataset contains no smooth pursuits, there

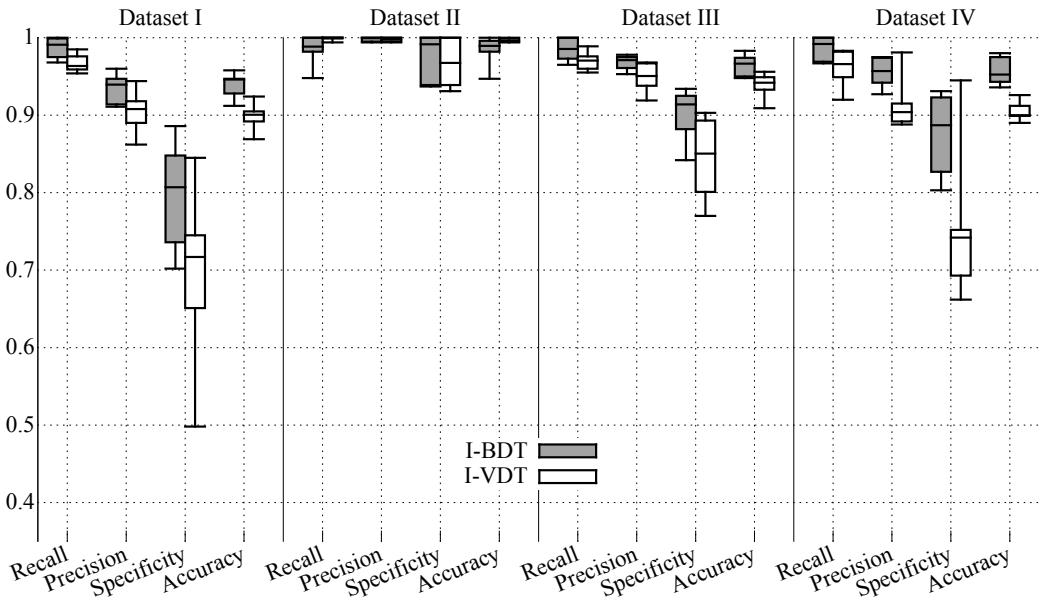


Figure 5.7: Performance metrics per dataset for fixations [66].

is a clear velocity threshold separating the remaining movements, and, thus, I-VDT can clearly distinguish between them. I-BDT, however, is still affected by saccades surrounded by noise, on average classifying 2.18% of the samples as smooth pursuits. In contrast, dataset III exposes one of the I-VDT weaknesses as it contains smooth pursuits with higher speeds (i.e., $44\,^{\circ}/\mathrm{s}$); as a result, smooth pursuit and saccade speeds overlap, yielding the misclassification of some high-speed pursuits and decreasing saccade classification precision.

Regarding smooth pursuit classification performance, Fig. 5.9 highlights the consistent good precision ($>$ 80%) through all datasets, scoring above 90% in the great majority of cases. I-BDT exhibits good recall ($>$ 85%) for datasets III and IV. As mentioned previously, for dataset I there is a struggle to classify slow smooth pursuits, resulting in the smaller recall for this dataset. Furthermore, it is worth noticing that I-BDT cannot reach maximum recall by design; since the algorithm relies on a temporal window to consider smooth pursuits, there is an onset period after the smooth pursuit has started until I-BDT starts classifying samples as such.

Fig. 5.10 illustrates I-BDT's smooth pursuit classification relative to that of a domain expert. Notice how the algorithm detects a false short smooth pursuit sequence at the beginning due to a saccade surrounded by noise. In an offline version, such misclassifications could be eliminated, for example, by using a minimum duration threshold for smooth pursuits; the one in question, has a duration of approximately only 100 ms. Moreover, it is possible to perceive the onset period for the smooth pursuit detection at the beginning of each smooth pursuit; this onset period could also be dealt with in an offline version by employing a similar detection technique but reversing the order of the samples. Further-

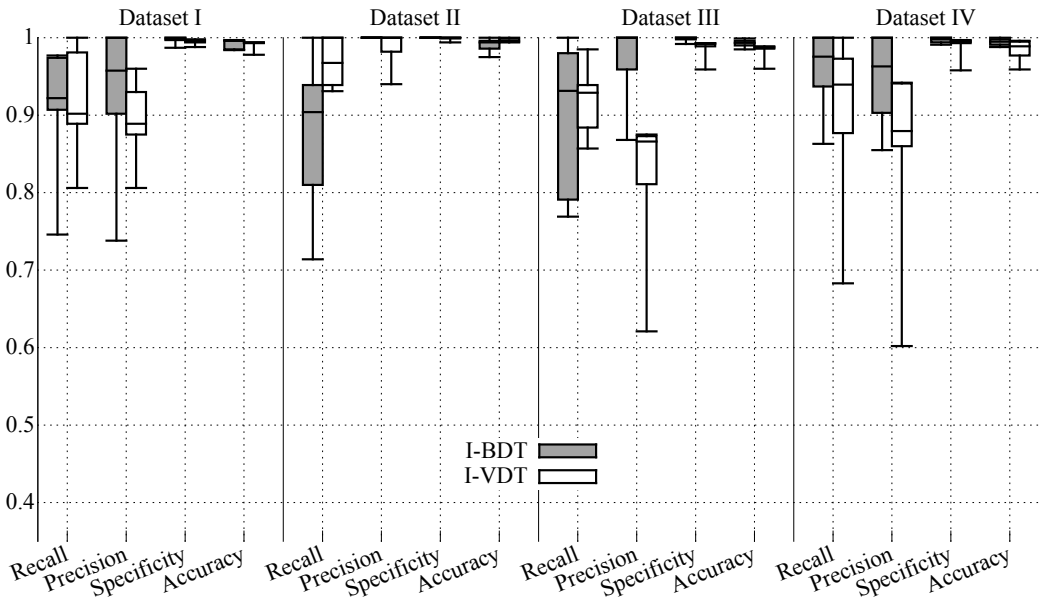


Figure 5.8: Performance metrics per dataset for saccades [66].

more, notice that during the second smooth pursuit the eye speed quickly switches between zero and close to zero values, misleading the algorithm, which does not detect the whole slow pursuit successfully. Thus, we do not advise the usage of I-BDT as is for very slow smooth pursuits when using low-resolution eye trackers; higher resolutions should alleviated this problem, but further investigation is required. It is worth noticing that, despite this weakness, low-resolution eye trackers are more appealing for embedded use in dynamic scenarios because these systems are cheaper, less computationally intensive, and consume less power than their high-resolution counterparts.

Comparing our results to those of related work is relatively complicated, mainly due to the lack of openness regarding algorithms and datasets, and due to differences in eye-tracking systems and metrics used for evaluation. Regarding dataset, eye-tracking system, and online constraints, our work is most similar to [249]. Our dataset design was heavily influenced by the dataset used in [262] and [263]; the main differences are 1) smooth pursuits in this work are not restricted to horizontal and vertical directions, and 2) we chose not to include short smooth pursuits (e.g., amplitude of 2° and velocity of 30 °/s) as their durations are smaller than an acceptable latency for the subject to start tracking the target.

In their work, [249] report an accuracy for smooth pursuit detection up to 92% whereas, in this work, I-BDT reached an average accuracy of 94.98%, ranging from 90.57% to 98.19%. It is worth noticing that accuracy alone does not allow us to completely evaluate algorithm performance [264]. Unfortunately, the machine learning-based classifier presented in [249] is not available for evaluation on our dataset, nor is their dataset available for evaluation with other algorithms. Thus, a direct comparison of both methods could not be performed.

[239] use a subset of the dataset from [263]; however, their algorithm is designed for

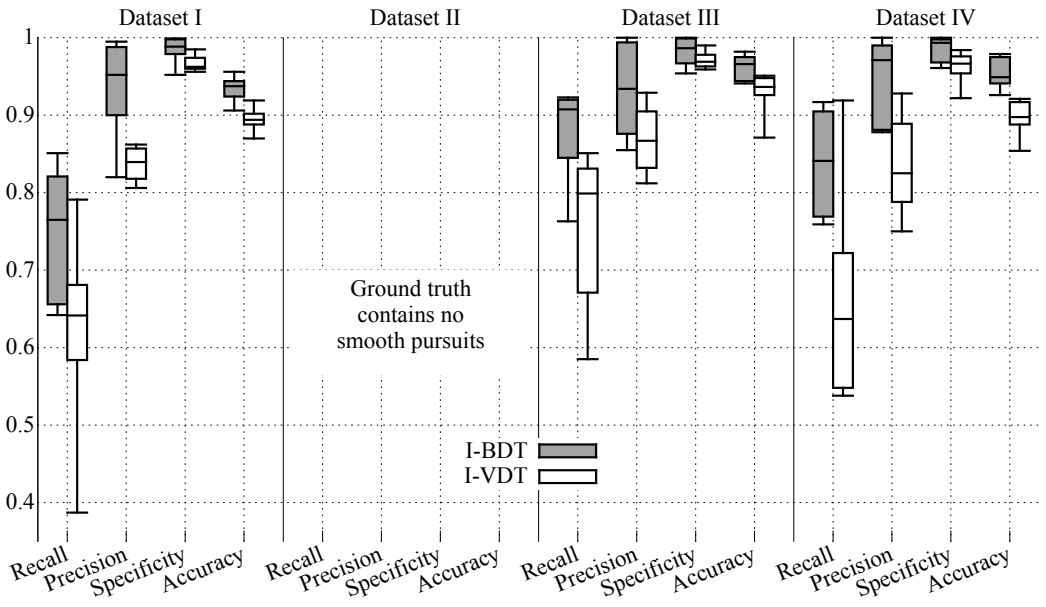


Figure 5.9: Performance metrics per dataset for smooth pursuits. Dataset II contains no smooth pursuits in the ground truth; thus, the resulting performance metrics are irrelevant and not reported [66].

offline analysis of high-resolution eye-tracking data. Thus, it cannot be applied to low-resolution eye trackers such as the one used in this work – mainly due to the preliminary segmentation stage relying on hypothesis testing, which would require a long time interval from the low-resolution eye tracker to be statistically significant (363 ms compared to the 22 ms used in their work). Nonetheless, their static *image* dataset can to some extent be compared to dataset II (in the sense that both do not contain smooth pursuits inducing elements). Similarly, their *video* and *moving dot* datasets can be compared to datasets I, III, and IV. Since their algorithm uses the same mechanism as I-VDT to separate saccades from other eye movements, their algorithm performance in this regard is clear. Thus, we briefly draw a parallel between their results for smooth pursuit and fixation classification and our results. Table 5.4 reports recall and specificity values from I-BDT mean results from this work, as well as best case results from [239] – to pick a best case scenario, we utilize the maximum value independent from which expert (1 or 2) was used as ground truth. Although I-BDT seems to provide better performance despite working under harder constraints, a fair and valid conclusion could only be drawn from similar experiments. Nonetheless, it is worth noticing that such an experiment is possible as I-BDT could be applied to the datasets from [239] (e.g., by coalizing the data into a lower resolution or applying I-BDT with adapted parameters). Unfortunately, neither dataset nor algorithm implementation from [239] were available at the time of writing.

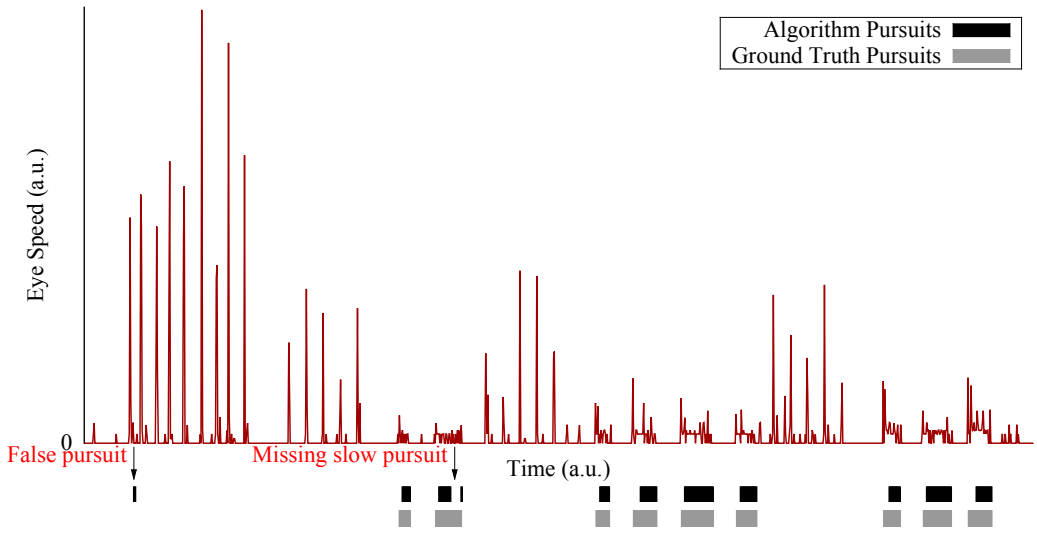


Figure 5.10: I-BDT smooth pursuit classification compared to that of a domain expert, accompanied by the eye-speed signal. A wrongly detected smooth pursuit and a partially detected slow smooth pursuit are highlighted. Moreover, notice the onset period required by the algorithm to classify the smooth pursuits [66].

		Recall		Specificity	
		I-BDT	**Larsson**	**I-BDT**	**Larsson**
Static	Fixation	0.985	≈0.93	0.977	≈0.98
	Pursuit	N/A	≈0.75	N/A	≈0.97
Dynamic	Fixation	0.986	≈0.90	0.859	≈0.85
	Pursuit	0.822	≈0.80	0.984	≈0.95

Table 5.4: Performance comparison between I-BDT and [239]. Static represents the average performance for dataset II compared to the best performance for the images dataset. Dynamic represents the average performance for datasets I, III, and IV compared to the best performance for the videos/moving dot datasets [66].

5.5 Conclusion

In this chapter, we have proposed and evaluated a novel algorithm for the real-time identification of fixations, saccades, and smooth pursuits. Since the algorithm operates directly on the eye-position signal, it requires no calibration step. The proposed algorithm displayed higher and more consistent performance than a state-of-the-art algorithm, demonstrating the capability of I-BDT to provide meaningful ternary classification. Moreover, an open-source *MATLAB* implementation of the algorithm is provided.

One of the main difficulties during evaluation, was the lack of open annotated datasets. The manual coding of eye movements is a subjective, laborious, and time-consuming task; thus, having to create one from scratch is far from ideal. In an effort to allow for review and to kick-start an open-access benchmark for the evaluation of eye movement identification algorithms, we provide our annotated datasets openly at `www.ti.uni-tuebingen.de/perception`.

For future work, we are interested in analyzing additional features for I-BDT to further improve its performance as well as evaluating the algorithm with higher-resolution eye trackers. Moreover, an important step to enable the fully automation of eye movements classification is a reliable detection of blinks, which the proposed algorithm does not take into account at the moment. Furthermore, we are intent on developing solutions to account for head movements in order to reliably distinguish smooth pursuits from vestibulo-ocular reflexes.

6 Apotheosis: The EyeRec Project

> "*The whole is greater than the sum of its parts.*"
>
> —Aristotle

On one hand, a platform is as good as the methods that it integrates. On the other, regardless of how extraordinary a method's performance might be, its usefulness is quite limited if users cannot *easily* take advantage of it. This symbiotic relationship is the key driver behind the EyeRec project, whose aim is to provide a *high-usability*, *transparent*, and *hardware agnostic* platform for *real-time pervasive* head-mounted eye tracking, offering *effortless integration* for new eye-tracking methods.

High-usability in the sense that it is straightforward and requires minimum interaction for non-expert and non-technical users to use. This means we strive to provide robust and parameterless[1] methods. **Transparent** in terms of being open source so users can know and choose which methods are used. **Hardware agnostic** meaning that EyeRec development is not guided by the requirements of a particular eye tracker, supporting a whole plethora of both specialized and Do-It-Yourself (DIY) devices instead. **Real-time** operation so users can utilize it for online gaze-based applications, such as HCI. **Pervasive** in a manner that it can be utilized in *out-of-the-lab* scenarios with an unsupervised and quick calibration procedure. Finally, **effortless integration** to the extent that we try and offer a clear and simple interface for common eye tracking tasks, allowing developers to simply encapsulate their methods within these interfaces and take advantage of the rest of the infrastructure and methods offered by the platform.

The aim of this chapter is to contextualize and give a brief overview of the EyeRec project. First a bit of history behind its origins and related work are introduced in Section 6.1. Henceforth, we use the term EyeRec to refer to the set of individual interfaces and method implementations that can easily be reusable within any eye-tracking software, whereas EyeRecToo refers to the software distribution integrating these methods and interfaces together with other required components to construct a fully functional eye tracking platform. Section 6.2 briefly dwells into the software architecture of EyeRecToo; afterwards, it introduces the EyeRec interfaces and how these connect into EyeRecToo. For more in-depth information, we refer the reader to the main repository hosted by the University of Tübingen (`https://atreus.informatik.uni-tuebingen.de/santini/EyeRecToo`) or the author's personal repository (`https://github.com/tcsantini/EyeRecToo`).

[1] Technically speaking, methods that do not require the user to tune parameters.

6.1 Origins

" EyeRec is a data acquisition software for head-mounted eye trackers. Its main raison d'être is to provide an open platform to replace the data acquisition functionality from eye-tracker vendors software, which typically are expensive and closed-source. Thus, if something is not working properly, you can fix it yourself instead of relying on the vendor to fix it; for instance, if the pupil detection algorithm does not suit your needs. "

—Excerpt from the first EyeRec distribution README

The vast majority of eye tracking users employ the accompanying software to their hardware. At first glance, this is a natural and well motivated decision: Theoretically, vendors should known their hardware *exactly* and, thus, know which constrains and assumptions are valid, allowing them to fully optimize their system. In practice, however, this is not always the case. For instance, companies: *a)* have a limited amount of resources and must choose carefully where to invest these resources, *b)* do not usually design and manufacture all their components – e.g., cameras are usually COTS, *c)* must take hardware costs into account, and *d)* are sometimes divided into separate entities with different priorities, making it hard to tend to user feedback adequately[2]. One should also be careful not to over-estimate the scientific competence of the vendors [47]. Additionally, commercial software presents other issues such as:

High cost: vendors usually must have a (justifiably) high markup on both software and hardware because of the small volume of sales in the eye-tracking market. Whereas most research users from academia and industry are able to afford these systems, it is a significant hindrance for small companies, third-world universities, and regular users. With the recent interest in eye tracking for accessibility [265], gaming [266], as well as virtual and augmented reality [267], the eye-tracking industry has seen a surge in interest that might decrease associated costs as evidenced by the release of low-cost remote eye trackers such as the *EyeTribe* and *Tobii 4C*.

Lack of transparency: most vendors consider their methods a vital component of their systems and, thus, keep these behind closed doors. This is particularly pertinent for eye movement research, where the system might actually hide (or even produce) events of interest, making some methods improper for certain research questions [121], [268]. Many vendors also filter the gaze signal, which can significantly alter results, but do not provide any information on *how* the filter works [47].

Lack of robustness and third-party access: given their limited resources, vendors often follow the *Pareto* principle[3], covering only a small set of possibilities that cover a

[2] A good example here is Tobii, which, despite offering the best eye-tracking solutions in the opinion of this author, has a big flaw in the fact that the user facing branch *(Tobii Pro)* has little influence over the branch that develops the technology *(Tobii Tech)*.

[3] A misnomer for the *"vital few and trivial many"* concept [269], [270].

majority of the users. This results in a lack of robustness, evidenced by the lack of eye-tracking systems that actually work with glasses and outdoors. From a business perspective, this is understandable but excludes a significant part of the population and applications nonetheless [53]. While some companies might be willing to implement a customer's property or functionality [47], this is often on a case-by-case basis and largely dependent on the customer size and effort required to implement the request. A possible solution then would be to allow users to plug-in their own methods so that researchers and other interested parties could improve such aspects; this, unfortunately, does not seem to happen in practice.

Aside from commercial software solutions, open-source solutions also exist [91], [165], [197]–[199], [214], [215], [271]–[277], mostly stemming from academia. Some notable mentions are: **a) The Haytham Gaze Tracker** [277], an open-source video-based eye tracker suited for head-mounted or remote setups that is currently in *beta* and offers some truly interesting functionality for interaction with computer screens, including head-gesture and blink recognition. **b) The ITU Gaze Tracker** [199], [273], a framework for open-source eye tracking using off-the-shelf components, such as web and video cameras, supporting both head-mounted and remote setups; some of its developers later went on to create the EyeTribe [278], which was later acquired by *Oculus VR* [127]. **c) Pupil** [124], [165], which started as a master thesis at the MIT [279] and developed into a commercially-driven (and most successful open-source solution to date) framework for open-source eye tracking under Pupil Labs GmbH [124], offering a handy plugin system[4]. For a more in-depth list of available eye tracking solutions, we refer the reader to the COGAIN's list [280]. With existing open-source solutions available, the reader might be wondering if developing yet another one is not just another case of Not Invented Here Syndrome (NIHS) or *standards proliferation* – see Fig. 6.1 for a quirky explanation of the latter.

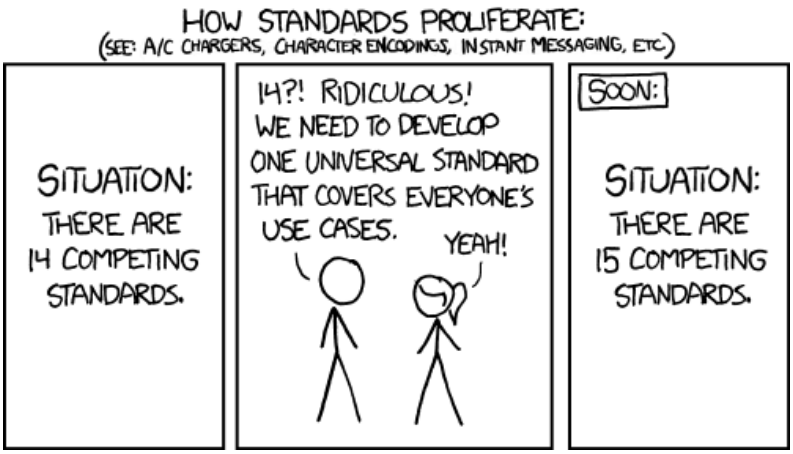


Figure 6.1: An XKCD Comics' illustration of *standards proliferation*, courtesy of Randall Munroe [281].

[4] https://docs.pupil-labs.com/#plugin-guide

In fact, one of this author's main usability complains with Pupil is exactly in this regard: For instance, developing in-house Graphical User Interfaces (GUIs) frameworks and device interface protocol specification[5], instead of relying on well-established components. Existing solutions also exhibit other problems such as guiding development towards a particular device, general licensing disagreements, use of language-specific containers, all of which contribute to non-adoption. EyeRec does not aim to be a *one-size-fit-all* solution; that would be a fallacy. Instead, EyeRec focus on head-mounted devices, trying to meet the requirements highlighted in the beginning of this chapter by:

1. using a camera-agnostic interface for handling different eye trackers,
2. providing a set of readily available methods for eye tracking, allowing for a straightforward method replacement (e.g., if a pupil detection method does not perform well for a participant) as well as method comparison for research or benchmarking purposes,
3. employing language-agnostic data formats, allowing for immediate data access regardless of which language and operating system the user favors, and
4. encapsulating functionality in a modern and cross-platform application framework (Qt), which provides portability, increased usability, and allows developers to focus resources on the eye-tracking aspect instead of already solved GUI and cross-platform development issues.

6.1.1 A Trace of the Past

Aside from technical aspects, it is worth preserving a bit of EyeRec's history to understand how it got to the its current state and where it is going. Let us go back to EyeRec's beginning back in 2015. At that time, we were working on improving pupil detection for an industrial cooperation project. While the focus was on pupil detection, we quickly realized one thing: *How could we actually integrate this new method into the eye tracker's vendor software for real-time operation?* As it turned out, there was no viable option, so we started looking into open-source alternatives, hoping to find a project *a)* in which to integrate our newly developed methods, and *b)* that we could adapt to our industry partner's requirements. Much to our dismay, none of the available alternatives could even access the eye tracker we were interested in, nor were our usability expectations met. There was, however, one internal project in our research group – developed by Thomas Kübler and which would later blossom into Smart Ocular Motility Analysis [9] (SOMA) – whose implementation was able to access the eye tracker's cameras [6]. This project's code base served as foundation for the development of the very first EyeRec incarnation[7] [70]. This name was suggested by Wolfgang Fuhl, who also contributed the first pupil detection algorithm (ExCuSe). Many of

[5] E.g., see https://github.com/pupil-labs/pyglui and https://github.com/pupil-labs/pyndsi

[6] Through the VideoMan [282] library

[7] The original repository has been archived but is still accessible at https://atreus.informatik.uni-tuebingen.de/santini/EyeRec/.

the design decisions were taken to satisfy the industrial partner's requirements, leading this version to little flexibility – for instance, only monocular eye trackers were supported, and selecting an eye tracker was clumsily done through editing a text file. Nevertheless, many lessons were learned from the development of the original version. These lessons translated into a much more flexible and streamlined second-generation: EyeRecToo[8] [67]. For this version, major architectural changes happened, and the whole infrastructure was rewritten from scratch[9] – with only the integrated pupil detection and gaze estimation methods withstanding this revamp. Some of the major improvements included *a)* the replacement of the old camera backend replacement by the *QMediaService*[10] infrastructure, and *b)* the integration of CalibMe. The former lead to a streamlined and uniform method to add support for new devices: Simply implementing a *QMediaServiceProviderPlugin*[11], giving device access not only to EyeRecToo, but also to other Qt-based applications. This also included the implementation of the *uvcengine*[12], which enables access to UVC-compliant devices, including the Pupil eye trackers. Since then, many small improvements (e.g., better video containers) as well as larger additions (e.g., the development and integration of PuRe, PuReST, and Grip) have contributed to turning EyeRecToo into a real alternative to users who are unsatisfied with the performance of existing eye-tracking platforms.

6.2 Architecture and Interfaces

EyeRecToo is built around widgets that provide functionality and configurability to the system. It was designed to work from a single camera configuration (e.g., a single eye camera to study eye movements or single field camera for egocentric video studies) to a large array of cameras (e.g., stereo eye cameras and multiple field cameras). Whereas it focus on mono or binocular head-mounted eye trackers (with a field camera), it also foresees the existence of other data input devices in the future – e.g., an Inertial Measurement Unit (IMU) for head-movement estimation [283]. This is achieved by assuming there is no hardware synchronization between input devices, employing a built-in software *synchronizer* instead.

Each input device is associated with an *input widget*. *Input widgets* register with the *synchronizer*, read data from the associated device, timestamp the incoming data according to a global monotonic reference clock, possibly process the data to extend it (e.g., pupil detection and marker detection), and save the resulting extended data, which is also sent to the synchronizer. The *synchronizer*'s task is then to store the latest data incoming from the *input widgets* (according to a time window specified by the user) and, at predefined intervals, generate a *DataTuple* with timestamp t containing the data from each registered *input widget* with timestamp closest in time to t, thus synchronizing the input devices data[13]. The resulting *DataTuple* is then forwarded to the *calibration/gaze estimation*, which

[8] Available at https://atreus.informatik.uni-tuebingen.de/santini/EyeRecToo/.

[9] We chose to add the *Too* suffix to indicate to users that this is a completely different program, rather than just an updated version of the original.

[10] http://doc.qt.io/qt-5/qmediaservice.html

[11] http://doc.qt.io/qt-5/qmediaserviceproviderplugin.html

[12] https://atreus.informatik.uni-tuebingen.de/santini/uvcengine

[13] In case the input devices are hardware-synchronized, one can use a delayed trigger based on any of the input

complements the tuple with gaze data. The complete tuple is then stored (*data journaling*), broadcasted through UDP (*data streaming*), and exhibited to the user (*GUI update*). This results in a modular and easily extensible design that allows one to later reconstruct events as they occurred during run time; this architecture is depicted in Fig. 6.2.

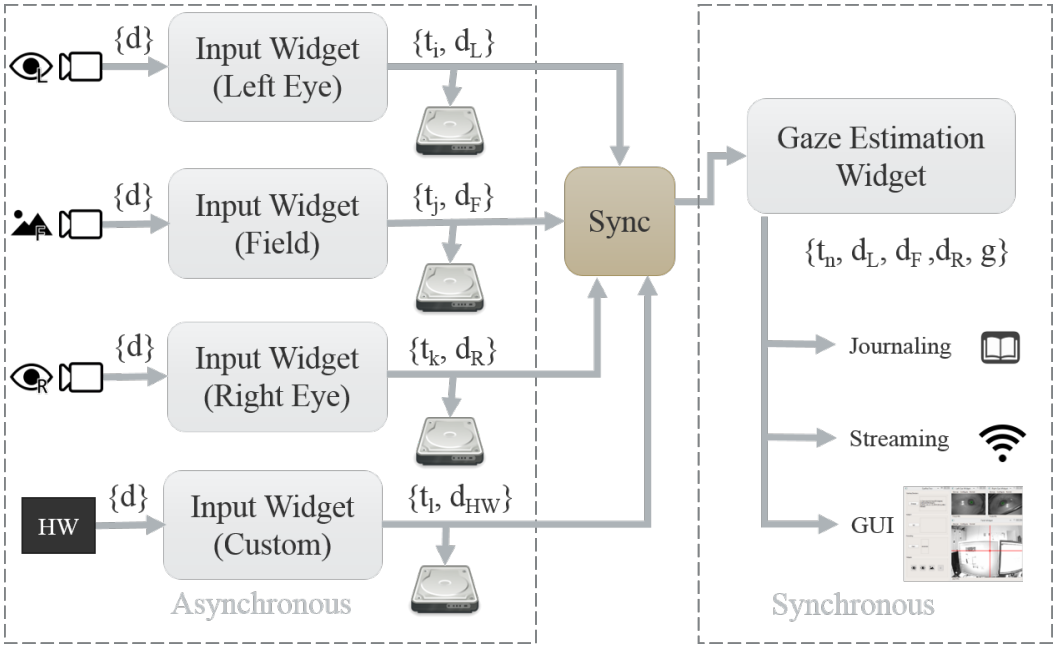


Figure 6.2: In the asynchronous section of EyeRecToo, input devices generate data (d), which are processed independently by the input widgets. These widgets then augmented the input stream based on the input widget type (e.g., with the detected pupil location), resulting in the *extended data* (d_L, d_R, d_F). These are are timestamped (t_i, t_j, t_k), stored to disk, and forwarded to the *synchronizer* (*Sync*) by the respective input widget. The synchronizer generates *data tuples* of synchronized data based on the timestamps, which the *gaze estimation widget* uses to derive a gaze estimation (g). The gaze estimate is then added to the *data tuple* and forwarded to the journaling, streaming and GUI update stages. Adding new input devices to the system is achieved through adding a *custom input widget*.

6.2.1 Input Widgets

Currently two input widgets are implemented in the system: the eye camera input widget, and the field camera input widget. These run in individual threads with highest priority in the system. The eye camera input widget is designed to receive close-up eye images, allowing the user to select during run time the input device, region of interest (ROI) in which eye feature detection is performed, image flipping, as well as pupil detection and tarcking algorithms. The field camera input widget is designed to capture the user's egocentric view, allowing the user to select during run time the input device, image undistortion,

devices to preserve synchronization.

image flipping and fiducial marker detection. Additionally, camera intrinsic and extrinsic parameter estimation is built-in. Currently, ArUco markers are supported. Both widgets store the video (as fed to their respective image processing algorithms) as well as their respective detection algorithm data and can be used independently from other parts of the system (e.g., to detect pupils in eye videos in an offline fashion).

6.2.2 Gaze Estimation Widget

This widget provides advanced calibration and gaze estimation methods, including two methods for calibration (*supervised* / *unsupervised*). The *supervised* calibration is a typical eye tracker calibration, often referred to as *natural features* calibration, which requires a human supervisor to coordinate with the user and select points of regard that the user gazes during calibration. The *unsupervised* calibration method is a implementation of the CalibMe method described in Chapter 4. Moreover, functionalities to save and load data tuples (both for calibration and evaluation) are implemented, which allows developers to easily prototype new calibration and gaze estimation methods based on existing data.

6.2.3 Interfaces

Throughout this section, we will omit the functions' signatures and simply refer to them solely by name a) for the sake of readability, and b) since these signatures might change in the near future. To further improve readability, functions shall be highlighted as `foo()` and classes as `Foo`.

EyeRec currently supports interfaces that allow users to easily extend *pupil detection*, *pupil tracking*, and *gaze estimation* functionality. This is achieved through C++ *abstract superclasses* containing *pure virtual* functions related to the desired functionality (e.g., pupil detection) as well as functions providing shared functionality (e.g., the generic confidence measure introduced in Section 3.1.2). Naturally, to introduce a new method to the architecture, one simply creates a *subclass* inheriting from the appropriate superclass, implementing all pure virtual functions. Similarly, each method is identifiable by requiring the developer to implement a `description()` pure virtual function. These interfaces are provided by the `PupilDetectionMethod`, `PupilTrackingMethod`, and `GazeEstimationMethod` abstract superclasses.

In the context of EyeRecToo, classes deriving from these methods can be registered in the system within `EyeImageProcessor` (pupil detection and tracking) and `GazeEstimation` (gaze estimation). `EyeImageProcessor` lives inside of the eye camera input widget, whereas `GazeEstimation` lives inside of the gaze estimation widget. After registering new classes, EyeRecToo automatically handles attaching the method to the GUI's list of available methods, calls the appropriate class' methods if so selected by the user, and integrates the results into the rest of the framework.

Pupil detection is accessible through the `detect()` and `detectWithConfidence()` functions. The former interface provides access to the as-is implementation, whereas the latter augments the result with a generic pupil confidence metric. Developers

wishing to integrate new methods provide the implementation encapsulated in the `implDetect()` function. EyeRec includes implementations for ExCuSe, ElSe, and PuRe.

Pupil tracking is accessible through the `detectAndTrack()` function, which automatically switches between pupil detection and tracking based on the pupil confidence metric. Users can choose any mix of pupil detection and tracking algorithms. Developers wishing to integrate new methods provide the implementation encapsulated in the `track()` function. If their method also provides a indispensable detection stage, they can override the virtual `detect()` function instead of allowing the user to pick any pupil detection algorithm. Unless the detection stage is indispensable, we highly recommend developers to integrate the detection stage as a pupil detection method instead. For finer control over the switching between detection and tracking, `detectAndTrack()` is marked virtual, allowing developers to override it if so desired. EyeRec includes a implementation for PuReST.

Gaze estimation functionality is accessible through two functions: `calibrate()` and `estimate()`. Developers wishing to integrate new methods provide the implementation encapsulated in the `calibrate()` and `implEstimate2d()` functions. A barebones implementation produces a gaze estimate for each eye camera independently. The interface then takes care of deriving a binocular gaze estimate and determining the appropriate gaze estimate (i.e., monocular left eye, monocular right eye, or binocular) based on user preference and input validity (e.g., pupil confidence and camera availability). Developers can obtain finer control over binocular gaze estimation by overriding the virtual `estimateBinocular2d()` function. EyeRec includes implementations for Grip, *homographies*, as well as monocular and binocular versions of *bivariate and quadvariate polynomial regressions* of multiple polynomials.

7 Case Study

> "*In theory, there is no difference between theory and practice. But, in practice, there is.*"
>
> —Walter J. Savitch

In this chapter, we demonstrate the applicability and effectiveness of EyeRecToo and the methods introduced in this thesis through a large-scale fully-unconstrained mobile eye tracking study. In particular, it is worth highlighting the following aspects:

- Real-time performance using a mobile and light-weight processing unit, including data acquisition, processing, and storage.
- Broad participant coverage, allowing for over one hundred participants from over thirty distinct nationalities between 18 and 66 years old without any a) selection of participants suited for eye tracking nor b) parameter adjustment.
- High usability, enabling data collection by multiple unexperienced experimenters.
- Low cost, allowing us to collect data with four systems in parallel at a fraction of the price of typical commercial eye tracking systems.

Here, we focus on system design and usability. For results regarding gaze estimation performance, please refer to Section 4.2.3.

7.1 Motivation

Pervasive mobile eye tracking provides a rich data source to investigate human natural behavior [284], [285], which provides a higher degree of ecological validity than traditional *in-the-lab* controlled experiments, specially when combined with natural environments, In the context of museums, there is strong evidence that ecologically valid testing in natural conditions (e.g., in museum conditions) is paramount for experimental aesthetics [286], and object authenticity (e.g., photographs vs real objects) has a positive impact on subjects' attention [287]. Moreover, eye tracking provides significantly more detailed insights than traditional timing-and-tracking or external observer approaches [287] – e.g., see Fig. 7.1 and Fig. 7.2. Previous works using mobile eye trackers in museums have investigated the difference in eye movements between children and adults [286] and how expertise influences viewing behavior of domestic textiles [288], as well as performed exploratory studies [285], [287], [289], [290]. However, many of these works are either constrained

(e.g., subjects observe paintings only from a fixed position), have very short durations, or do not discuss the accuracy of the eye trackers, which is known to still suffer from multiple issues such as drift and parallax errors [47]. In this Section, we present in detail the eye tracking system used in a large scale *fully-unconstrained* study in the Austrian Gallery Belvedere, providing useful information for system designers. This study is described in detail in Section 7.3 and is part of a larger experiment, whose aim is to investigate how visitors perceive artworks in relation to distinct museological settings[1].

Figure 7.1: Eye-tracking enabled insights. Eye tracking reveals the trajectory of fixations (i.e., the *scanpath*) as the visitor attends to Giovanni Segantini's *The Evil Mothers*. These *scanpaths* provide key insights into the human cognitive process and are also useful for distinguishing a subject's expertise level [291]. Figure best visualized in digital form [53].

[1] Combinations of artworks, spacial placement, additional textual information, etc.

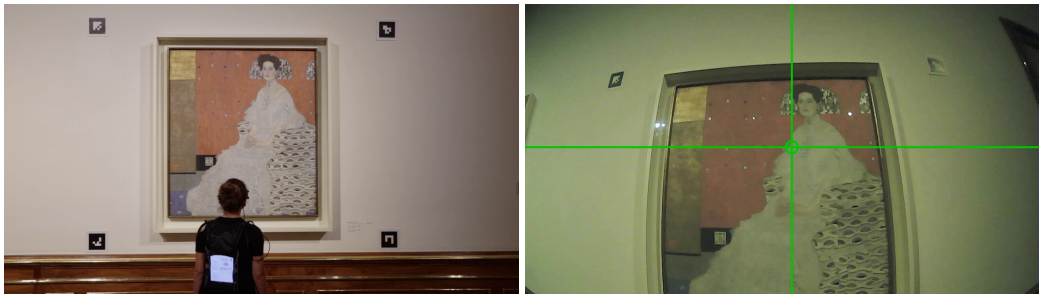


Figure 7.2: Eye-tracking enabled insights. Whereas an external observer might be tempted to consider that the visitor is gazing at the face from Gustav Klimt's portrait of *"Amalie Zuckerkandl"* because of its saliency, eye tracking reveals the visitor's true fixation position, suggesting an analysis of the painting details along Amalie's dress. Figure best visualized in digital form [53].

7.2 Eye-Tracking System

When designing the eye-tracking system, one of our goals was to make the system as general as possible while maintaining *real-time* and *pervasive in-device* eye tracking functionality. This includes being accessible for users with glasses, which represent about 30 percent of the young adult population [292], without the need for them to remove their glasses – e.g., as required by Tobii and SMI glasses. This is particularly critical with the ever growing myopia epidemic; for instance, in East and Southeast Asia, myopia prevalence reaches over 80% of the young adult population [293]. A direct result from these requirements is that the resulting system can be easily calibrated and used in real-time in an individual fashion, enabling future gaze-based human-computer interaction applications [22] – e.g., gaze-activated audio guides. Research wise, the developed system also allows an experimenter to check pupil detection and gaze estimation accuracy in real time without the need of additional devices and available network, allowing for significant improvement in data quality as well as a much simpler and cheaper experimental setup. In comparison, other mobile eye-tracking systems commonly require a separate recording unit and a separate control device for the experimenter – e.g., Dikablis, SMI, and Tobii glasses [160], [194], [195].

7.2.1 Hardware

We employed a binocular Pupil [124], [165] (Pupil) head-mounted eye tracker (2x *Pupil Cam2* eye cameras; 1x *Pupil Cam1* scene camera). This device was paired with a *Microsoft Surface Pro 4*[2] tablet running Windows 10.1, a USB 3.0 hub, and a slide presenter. Through this slide presenter, experimenters could toggle the recording, calibration, and view of all cameras, allowing for a seamless remote-control experience without direct tablet interaction. Since the subjects were free to roam, it was necessary to find a comfortable and lightweight method for carrying the hardware components. Our initial candidate was a backpack designed for this tablet and provided as part of the Dikablis Wireless system [160]. However,

[2]Configured with a Intel® Core™ i5-6300U

Figure 7.3: Subject contemplating Wilhelm Bernatzik's *Pond* while wearing the eye tracking system during the experiment. Notice the remote-control receiver and the eye tracker cable near the left and right shoulders, respectively [53].

aside from the steep price (350 EUR each), the protective mesh hampered air flow and made it impossible to use the tablet's touch screen without opening the backpack. Thus, we built our own solution out of a biking hydration backpack, each costing less than 9 EUR [294]. This was easily achieved by removing a cutout from the backpack canvas as well as parts of the side fabric; the tablet was then inserted into the backpack with a cardboard piece and two Styrofoam bars separating the tablet from the backpack side that stays in contact with the user's back. This solution allowed for a decent amount of airflow and tablet interaction, while keeping the warmth of the tablet away from the subject. Furthermore, two apertures (originally designed for the water reservoir straws) located at the seams of the backpack straps provided ideal places to run the eye tracking cable and place the remote-control receiver. The complete system (see Fig. 7.3) weighs less than 1 kg with a total cost of about

3197 EUR – contributed mostly by the eye tracker (2150 EUR) and tablet (1015 EUR); user feedback is reported in Section 7.4. Thanks to this small cost, we were able to conduct the experiment with four systems simultaneously; in contrast, solutions provided by vendors typically cost *more than our four systems combined*. An additional benefit enabled by multiple systems is the ability of running simultaneous measurements within groups, allowing for the exploration of collaborative learning and social engagement [290]. It is worth noting that this social aspect adds to the ecological validity of museum-related experiments as approximately 75 to 95 percent of visits are done in groups.

7.2.2 Software

Although the Pupil eye tracker manufacturer provides a software solution (Pupil [124] (Pupil Capture) and Pupil [124] (Pupil Player)), we opted to use EyeRecToo instead. One of the main reasons for this decision was the difficulty in calibrating outside of the lab in natural environments with Pupil Capture and its *manual marker calibration*[3]. Additionally, the pupil detection we experienced seems to be bimodal, either working relatively well or not at all. EyeRecToo on the other hand provides a robust marker detection (through ArUco [186] (ArUco)) and an advanced calibration method called CalibMe, which uses a target moving w.r.t. the scene camera for calibration (see Section 7.3). Furthermore, EyeRecToo also provides PuRe and PuReST, which significantly outperforms the methods available in Pupil (as shown in Chapter 3).

7.2.3 Advice for System Designers

Pervasive robust video-based eye tracking remains not only challenging, but also computationally expensive. Since embedded devices are required to allow for mobility, handling three[4] camera streams, image processing, and data recording is not trivial. In particular, one of the biggest challenges we found was **performance throttling mechanisms** – e.g., to stay within thermal constraints [295]. Such mechanisms are present virtually in every powerful modern system on chip. For instance, the *Intel Extreme Tuning Utility* [296] reports *four* different throttling mechanisms for the device employed in this work: *Thermal*, *power limit*, *current limit*, and *motherboard VR thermal* throttling. While some of these mechanisms are activated to keep the device within thermal constraints or because the battery can no longer supply enough current, the end result is more or less the same: *A significant performance drop*. Practitioners should be aware of such behavior and *test* during longer periods of operation to assure that performance requirements are within the expected range. Failing to do so might result in *severe* frame dropping during field experiments; it is worth noting that EyeRecToo provides a useful performance monitoring widget, which allows one to monitor frame dropping. Furthermore, when developing mobile systems such as ours, it is important to **minimize single points of hardware failure**, such as USB connections.

[3] Pupil seems to be aware of this issue since modifications to the marker were made recently; see: https://github.com/pupil-labs/pupil/releases/tag/v1.2

[4] The actual number of cameras might change depending on the eye tracker. For instance, a monocular one has two cameras, whereas Tobii glasses has five.

During our initial in-the-lab tests, we noticed that the connection between the USB hub and the eye tracker cable would spuriously stop working because of cable movement; a simple solution with Velcro was promptly arranged to make sure the connection remained tight even during harsh movements. In contrast, we experienced connection issues with the Pupil USB clip only a single time, during the field experiment. Finally, it is also important to **be aware of operating system specific requirements and caveats**. For Windows 10.1, we found the following worth of mention[5]: 1) disabling the *Connected Standby* feature[6], which allows for a more extensive customization of advanced power settings, 2) disable sleeping/hibernation options, 3) disable *USB selective suspend setting*, 4) disable *Turn off hard disk after*, 5) uninstall *Microsoft OneDrive*, 6) disable the *Windows search index service*, 7) disable the *Windows updater service* **after every reboot**. The latter three items are necessary to prevent Windows from running CPU intensive tasks during experiments.

7.3 Data Collection

7.3.1 Participants

Visitors arriving at the top of the *Grand Staircase* of the Austrian Gallery Belvedere were invited to join the experiment. The only requirements for a participant to take part was their consent (thus requiring them to be 18 or older) and ability to speak either English or German. The experiment took place from the 22nd to 28th of January 2018, with data being collected during five of these days. In total 109 subjects (63 females) took part in our experiment averaging 34.86 years of age ($\sigma = 14.62$).

7.3.2 Procedure

Upon accepting to take part in the experiment, the subject received and read a consent form containing experiment instructions. Afterwards, an experimenter assisted the subject to don the eye-tracking system (see Section 7.2 for details). The subject was then asked to stand on a floor mark approximately 1.16 m away[7] from a CalibMe collection marker that stood at about the same height as the paintings in the gallery. The experimenter adjusted the scene camera to center the collection marker, and then proceeded to adjust the eye cameras to a suitable position. Subsequently, the user was instructed on how to perform the calibration by always gazing at the central intersection of the collection marker while moving his head in a spiral fashion smoothly and slowly. The experimenter then started the recording and the subject calibration procedure, controlling when the eye tracker calibration started and stopped. After calibration, the subject was asked to gaze at four Post-its® about 25° away from the marker center so the experimenter could check the gaze estimation quality. It

[5] Please note that most of these informations came from extensive interactions with *Microsoft Support*. **These changes are provided as part of a detailed documentation of our system without warranty of any kind. In no event shall the authors be liable for any claim, damages or other liability.**

[6] registry: *HKEY_LOCAL_MACHINE\System\CurrentControlSet\Power\CsEnabled=0*

[7] Calibration distance was selected to minimize parallax error for the expected viewing range from 0.5 m to 3 m.

is worth noting that the experiment was conducted by *ten experimenters without previous experience with head-mounted eye trackers*, which prior to the experiment had a half hour instruction session with an eye-tracker expert with about three years of experience. The participant was then instructed to freely roam through four rooms (containing more than 30 distinct paintings and sculptures) as he/she wished, and that the experimenter would meet him at the end of the last room. At the end of the visit, the experimenter and subject proceeded to a separate room where: 1) a second calibration was performed, 2) the subject was interviewed and performed a remembrance mapping task, and 3) the subject answered a questionnaire containing museum-visit and eye-tracking related questions. After the experiment, participants were rewarded with a small souvenir.

7.4 Usability Results

For the usability analysis, we have included all participants. Since the museum-visit-related questionnaire and interview already lasted a considerable amount of time, we opted to include only seven eye-tracking-related questions as to not overload subjects. Participants were asked *"Please indicate how much you agree with the following statement:"* for each of the statements listed in Fig. 7.4. This figure also shows the distribution of the subjects' feedback[8]. *Q1* addresses one of the main issues with eye tracking: The calibration, which has been considered one of the main factors hindering eye tracking adoption [51] and is often considered of poor usability, experienced as difficult, and described as tedious [52]. In this regard, CalibMe was perceived by $\approx 92\%$ of users as easy to follow, even when instructed by *non-experienced* experimenters. *Q2*, *Q3*, and *Q4* address comfort and obtrusiveness [290], [298]. Feedback from these questions show that 1) the small-footprint 3D-printed eye tracker produced by Pupil has a good comfort rating (agreed by $> 84\%$ participants) without limiting visibility – agreed by $> 86\%$ of participants, and 2) the whole system is perceived as comfortable while maintaining mobility, real-time gaze estimation capability, and good autonomy ($\approx$ one and a half hour of recording and processing). *Q5* and *Q6* probe the implications of head-mounted eye tracking for the ecological validity of data collected with such devices: Even with the unobtrusiveness of the system, 31.5% of participants feeling *moderately* observed and 6.48% observed. Nonetheless, 76.1% of the participants felt that the eye tracking equipment *did not* interfere with their art perception.

[8] Originally the answers were formatted in a *"Not at all"* to *"Very much"* seven point Likert [297] scale. To facilitate visualization, we have coalesced the range [2-3] and [5-6] into single agreement/disagreement items.

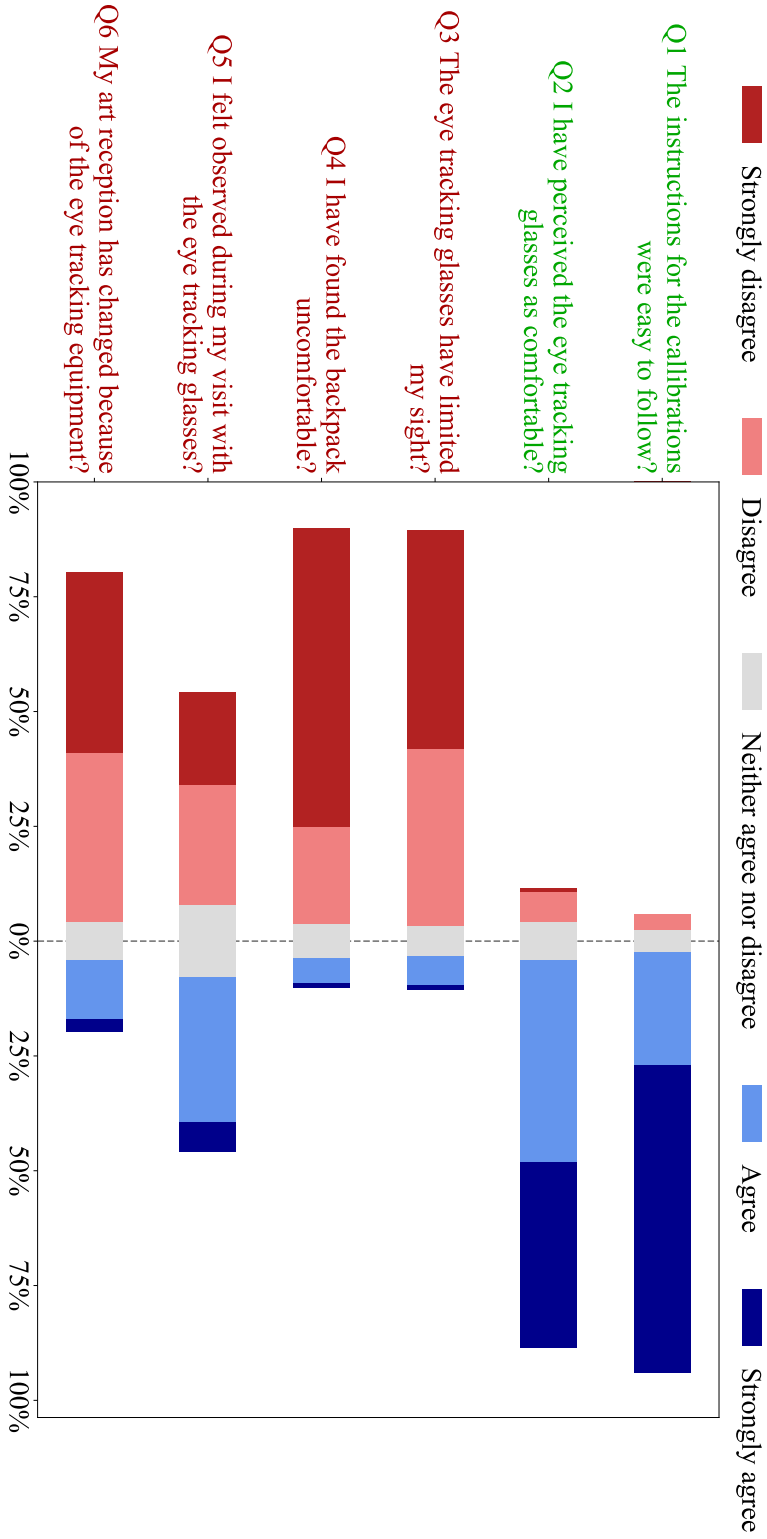


Figure 7.4: Distribution of participant's answers to the eye-tracking usability questionnaire. For the top two questions (in green), the more the subjects *agree*, the better. For the remaining four questions (in red), the more they *disagree*, the better [53].

Additionally, we also inquired the subjects at which point in time they forgot about the eye-tracking equipment, with the majority reporting they forgot it within two minutes into the experiment as shown in Fig. 7.5. Nevertheless, a significant part (33.94%) reported that they never forgot about the device, possibly because the head-mounted eye tracker used still is too salient in the subjects' field of view. This suggests that further improvements are still needed to make eye trackers less conspicuous.

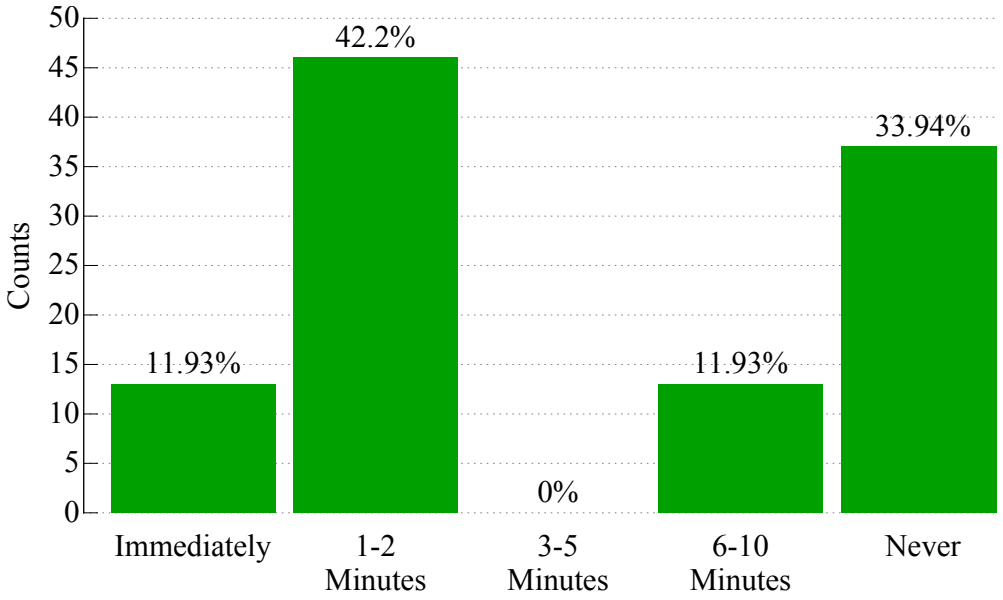


Figure 7.5: Distribution of counts regarding at which point in time participants forgot about the device during the experiment [53].

7.5 Gaze Estimation Results

As previously mentioned, for gaze estimation performance, we refer the reader to Section 4.2.3. At the time of data collection, the real-time gaze estimation of the system was that of the BPF since Grip had not yet being developed. Thus, results in Section 4.2.3 are from *post processing*. Nevertheless, Grip adds no signficant time constraints and, thus, the most recent version of EyeRecToo can be used with it in real-time.

7.6 Conclusion

In this chapter, we presented in detail the eye tracking system used in a large scale *fully-unconstrained* study in the Austrian Gallery Belvedere. Our usability results show that whereas the calibration and comfort of the system is already at an acceptable level, further improvements are necessary to make head-mounted eye trackers more inconspicuous.

These findings lead us to develop the inconspicuous and modular head-mounted eye trackers (*Proposed*) [25] shown in Fig. 7.6, reducing field of view occlusion by ≈ 66% w.r.t. to the smallest commercial offer.

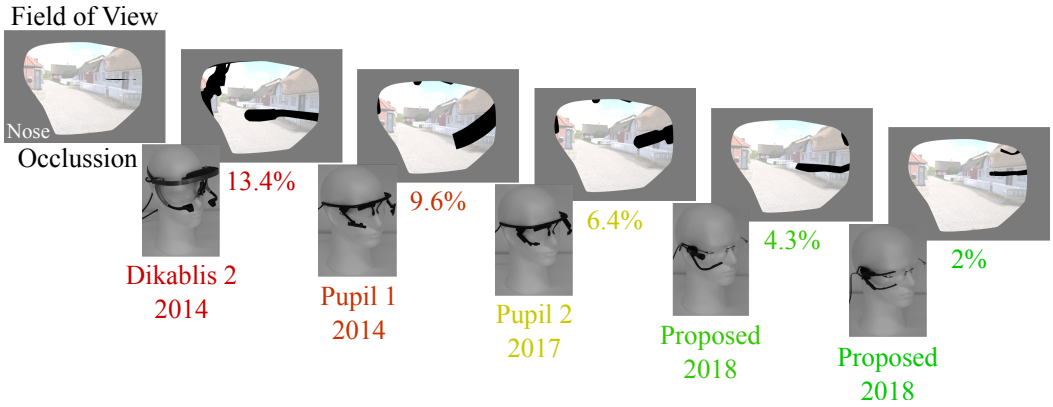


Figure 7.6: Top left shows an user's right eye field of view. The subsequent images demonstrate the evolution of head-mounted eye trackers, highlighting the occlusion visually and quantitatively for distinct eye trackers. Each eye tracker's occlusion rate (shown in %) was calculated as the ratio of pixels occluded by the eye tracker and the pixels visible in the field of view (non-grey area).

From an accuracy point of view, the eye tracker slipped for the great majority of the participants, resulting in high angular gaze offset errors. This showed us that even with a light-weight eye tracker, slippage is mostly *inevitable*. This prompted us to investigate an slippage-robust solution, which resulted in the development of Grip, introduced in Chapter 4.2.

Currently, future work focuses on how to automatically produce semantic mapping from the scene camera images – i.e., to give a semantic meaning to the image region surrounding the gaze estimate. We are currently investigating approaches to achieve this mapping. For instance, by employing keypoints (such as *ORB* [299] and *BRIEF* [300]) or by using convolutional neural networks (CNNs) for segmentation (such as Mask R-CNN [301]) to determine each painting/statue's pose and identification, allowing us to map fixations relative to the scene camera to objects of interest. Similarly, by using state-of-the-art CNNs for face detection [302] and recognition [303], it is also possible to estimate social interaction, an approach that can be easily extended to other social environments – e.g., classrooms [54]. Furthermore, the object-of-interest' pose can also provide meaningful cues such as interaction distance range. Although we did not measure participant willingness to wear the eye-tracking system, anecdotally we did not experience negative responses from the visitors towards the system. Nonetheless, this is a further usability question that should be considered in the future. Moreover, since traditional gaze-based interaction techniques (e.g., dwell time [49]) might prove insufficient for such dynamic scenarios, multimodal interfaces should be considered. Finally, it is worth considering remote gaze-sensing solutions, such as *EyePliance* [304], which might offer a less intrusive solution to map visitors' attention.

8 Final Remarks

> "*Know when you're finished, and when you are, put your pencil or your paintbrush down. All the rest is only life.*"
>
> —Stephen King

Ubiquitous wearable eye tracking holds immeasurable potential to further advance our understanding of the human mind and revolutionize the way we interact with our devices. This thesis presents a significant move in this direction, contributing several key steps in multiple areas of eye-tracking technology towards this goal.

In their 2010 large scale review [23], Hansen and Ji concluded that "The tendency to produce mobile and low-cost systems may increase the ways in which eye tracking technology can be applied to mainstream applications, but may also lead to less accurate gaze tracking. While high accuracy may not be needed for such applications, mobile systems must be able to cope with higher noise levels than eye trackers indoors use". The novel pupil detection (PuRe) and tracking (PuReST) methods contributed by this thesis deal directly with this increased noise level aspect of pervasive eye tracking. The methods herein developed provide robust real-time estimates not only for pupil center but also for its outline. Through this improved outline estimation, we were able to develop a novel slippage-robust and glint-free gaze estimation method (Grip) that has been demonstrated to outperform state-of-the-art and commercial solutions in the presence of slippage during *real and large-scale* unconstrained eye tracking for a large number of users. Combined with the proposed fast and unsupervised calibration method (CalibMe), this set of methods tackles the three major challenges presented in Section 1.1. Thus, this work enables unsupervised ubiquitous wearable eye tracking for a large portion of the population without the need for complex specialized hardware, allowing for hardware-agnostic solutions and easing the integration of eye-tracking into head-worn devices. Moreover, included in this large portion of the population are users with glasses, which have so far been relegated by the majority of commercial head-mounted eye-tracking systems.

This work however goes beyond novel methods and significant improvements over the state of the art. It contributes an open-source framework for wearable ubiquitous eye tracking (EyeRec), which allows researchers to a) easily prototype and integrate new eye-tracking methods, b) combine these methods with the remaining functionality required to achieve an eye-tracking system, and c) evaluate the combinations and interactions between a large gamma of eye-tracking methods. Moreover, these methods are readily available for non-technical users using multiple commercial and DIY eye trackers through EyeRecToo and have been proven through a large-scale fully-unconstrained eye tracking study.

Bibliography

[1] W. Fuhl, T. C. Santini, T. Kübler, and E. Kasneci, "Else: Ellipse selection for robust pupil detection in real-world environments", in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '16, Charleston, South Carolina: ACM, 2016, pp. 123–130, ISBN: 978-1-4503-4125-7. DOI: 10.1145/2857491.2857505. [Online]. Available: http://doi.acm.org/10.1145/2857491.2857505.

[2] A. George and A. Routray, "Escaf: Pupil centre localization algorithm with candidate filtering", *CoRR*, vol. abs/1807.10520, 2018. arXiv: 1807.10520. [Online]. Available: http://arxiv.org/abs/1807.10520.

[3] W. Fuhl, T. Kübler, K. Sippel, W. Rosenstiel, and E. Kasneci, "Excuse: Robust pupil detection in real-world scenarios", in *International Conference on Computer Analysis of Images and Patterns*, Springer, 2015, pp. 39–51.

[4] M. Tonsen, X. Zhang, Y. Sugano, and A. Bulling, "Labelled pupils in the wild: A dataset for studying pupil detection in unconstrained environments", in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '16, Charleston, South Carolina: ACM, 2016, pp. 139–142, ISBN: 978-1-4503-4125-7. DOI: 10.1145/2857491.2857520. [Online]. Available: http://doi.acm.org/10.1145/2857491.2857520.

[5] Qt Project, *Cross-platform software development for embedded & desktop*, Accessed in 2018-08-10. [Online]. Available: www.qt.io/.

[6] T. Santini, W. Fuhl, and E. Kasneci, "Pure: Robust pupil detection for real-time pervasive eye tracking", *Computer Vision and Image Understanding*, vol. 170, pp. 40 –50, 2018, ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2018.02.002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314218300146.

[7] ——, "Purest: Robust pupil tracking for real-time pervasive eye tracking", in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '18, Warsaw, Poland: ACM, 2018, 61:1–61:5, ISBN: 978-1-4503-5706-7. DOI: 10.1145/3204493.3204578. [Online]. Available: http://doi.acm.org/10.1145/3204493.3204578.

[8] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, ISSN: 0001-0782. DOI: 10.1145/358669.358692. [Online]. Available: http://doi.acm.org/10.1145/358669.358692.

[9] Forschungs Information Tübingen, *Smart ocular motility analysis - entwicklung eines neuartigen diagnosesystems für die automatisierte untersuchung von augenmotilitätsstörungen*, Accessed in 2018-08-10. [Online]. Available: https://fit.uni-tuebingen.de/Activity/Details?id=3822.

[10] USB Implementers Forum, Inc, *Usb device class specifications*, Accessed in 2018-08-10. [Online]. Available: http://www.usb.org/developers/docs/devclass_docs/.

[11] A. T. Duchowski, K. Krejtz, I. Krejtz, C. Biele, A. Niedzielska, P. Kiefer, M. Raubal, and I. Giannopoulos, "The index of pupillary activity: Measuring cognitive load vis-à-vis task difficulty with pupil oscillation", in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18, Montreal QC, Canada: ACM, 2018, 282:1–282:13, ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173856. [Online]. Available: http://doi.acm.org/10.1145/3173574.3173856.

[12] D. S. Asfaw, P. R. Jones, N. D. Smith, and D. P. Crabb, "Data on eye movements in people with glaucoma and peers with normal vision", *Data in Brief*, 2018.

[13] C. Braunagel, W. Rosenstiel, and E. Kasneci, "Ready for take-over? a new driver assistance system for an automated classification of driver take-over readiness", *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 4, pp. 10–22, 2017.

[14] O. Palinko, F. Rea, G. Sandini, and A. Sciutti, "Robot reading human gaze: Why eye tracking is better than head - tracking for human-robot collaboration", in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ - International Conference on*, IEEE, 2016, pp. 5048–5054.

[15] R. Bixler and S. D'Mello, "Automatic gaze-based user-independent detection of mind wandering during computerized reading", *User Modeling and User-Adapted Interaction*, vol. 26, no. 1, pp. 33–68, 2016.

[16] C. D. Frith and U. Frith, "Implicit and explicit processes in social cognition", *Neuron*, vol. 60, no. 3, pp. 503–510, 2008.

[17] K. A. Pelphrey and J. P. Morris, "Brain mechanisms for interpreting the actions of others from biological-motion cues", *Current Directions in Psychological Science*, vol. 15, no. 3, pp. 136–140, 2006.

[18] S. Baron-Cohen, S. Wheelwright, J. Hill, Y. Raste, and I. Plumb, "The "reading the mind in the eyes" test revised version: A study with normal adults, and adults with asperger syndrome or high-functioning autism", *Journal of child psychology and psychiatry*, vol. 42, no. 2, pp. 241–251, 2001.

[19] K. Lee, M. Eskritt, L. A. Symons, and D. Muir, "Children's use of triadic eye gaze information for" mind reading."", *Developmental psychology*, vol. 34, no. 3, p. 525, 1998.

[20] S. Baron-Cohen, "How to build a baby that can read minds: Cognitive mechanisms in mindreading", *The maladapted mind: Classic readings in evolutionary psychopathology*, pp. 207–239, 1997.

[21] P. Majaranta and A. Bulling, "Eye tracking and eye-based human–computer interaction", in *Advances in Physiological Computing*. London: Springer London, 2014, pp. 39–65, ISBN: 978-1-4471-6392-3. DOI: 10.1007/978-1-4471-6392-3_3. [Online]. Available: http://dx.doi.org/10.1007/978-1-4471-6392-3_3.

[22] A. Bulling and H. Gellersen, "Toward mobile eye-based human-computer interaction", *IEEE Pervasive Computing*, vol. 9, no. 4, pp. 8–12, 2010.

[23] 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*, vol. 32, no. 3, pp. 478–500, 2010. DOI: 10.1109/TPAMI.2009.30. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2009.30.

[24] T. Santini, W. Fuhl, and E. Kasneci, "Calibme: Fast and unsupervised eye tracker calibration for gaze-based pervasive human-computer interaction", in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17, Denver, Colorado, USA: ACM, 2017, pp. 2594–2605, ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025950. [Online]. Available: http://doi.acm.org/10.1145/3025453.3025950.

[25] S. Eivazi, T. C. Kübler, T. Santini, and E. Kasneci, "An inconspicuous and modular head-mounted eye tracker", in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '18, Warsaw, Poland: ACM, 2018, 106:1–106:2, ISBN: 978-1-4503-5706-7. DOI: 10.1145/3204493.3208345. [Online]. Available: http://doi.acm.org/10.1145/3204493.3208345.

[26] T. Santini, D. C. Niehorster, and E. Kasneci, "Get a grip: Slippage-robust and glint-free gaze estimation for real-time pervasive head-mounted eye tracking", in *Proceedings of the 2019 ACM Symposium on Eye Tracking Research & Applications – To appear*, ser. ETRA '19, New York, NY, USA: ACM, 2019.

[27] A. Swaminathan and M. Ramachandran, *Enabling augmented reality using eye gaze tracking*, US Patent 9,996,150, 2018. [Online]. Available: https://patents.google.com/patent/US9996150B2.

[28] A. Bulling and K. Kunze, "Eyewear computers for human-computer interaction", *Interactions*, vol. 23, no. 3, pp. 70–73, Apr. 2016, ISSN: 1072-5520. DOI: 10.1145/2912886. [Online]. Available: http://doi.acm.org/10.1145/2912886.

[29] S. Naspetti, R. Pierdicca, S. Mandolesi, M. Paolanti, E. Frontoni, and R. Zanoli, "Automatic analysis of eye-tracking data for augmented reality applications: A prospective outlook", in *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, Springer, 2016, pp. 217–230.

[30] E. Wood and A. Bulling, "Eyetab: Model-based gaze estimation on unmodified tablet computers", in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA '14, Safety Harbor, Florida: ACM, 2014, pp. 207–210, ISBN: 978-1-4503-2751-0. DOI: 10.1145/2578153.2578185. [Online]. Available: http://doi.acm.org/10.1145/2578153.2578185.

[31] D. Mardanbegi and D. W. Hansen, "Mobile gaze-based screen interaction in 3d environments", in *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications*, ser. NGCA '11, Karlskrona, Sweden: ACM, 2011, 2:1–2:4, ISBN: 978-1-4503-0680-5. DOI: 10.1145/1983302.1983304. [Online]. Available: http://doi.acm.org/10.1145/1983302.1983304.

[32] D. Kern, A. Mahr, S. Castronovo, A. Schmidt, and C. Müller, "Making use of drivers' glances onto the screen for explicit gaze-based interaction", in *Proceedings of the 2Nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, ser. AutomotiveUI '10, Pittsburgh, Pennsylvania: ACM, 2010, pp. 110–116, ISBN: 978-1-4503-0437-5. DOI: 10.1145/1969773.1969792. [Online]. Available: http://doi.acm.org/10.1145/1969773.1969792.

[33] H. Drewes, A. De Luca, and A. Schmidt, "Eye-gaze interaction for mobile phones", in *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology*, ser. Mobility '07, Singapore: ACM, 2007, pp. 364–371, ISBN: 978-1-59593-819-0. DOI: 10.1145/1378063.1378122. [Online]. Available: http://doi.acm.org/10.1145/1378063.1378122.

[34] J. P. Hansen, D. W. Hansen, and A. S. Johansen, "Bringing gaze-based interaction back to basics.", in *HCI*, Citeseer, 2001, pp. 325–329.

[35] P. Blignaut, E. J. van Rensburg, and M. Oberholzer, "Visualization and quantification of eye tracking data for the evaluation of oculomotor function", *Heliyon*, vol. 5, no. 1, e01127, 2019.

[36] M. Vidal, J. Turner, A. Bulling, and H. Gellersen, "Wearable eye tracking for mental health monitoring", *Computer Communications*, vol. 35, no. 11, pp. 1306–1311, 2012.

[37] K. Kunze, S. Ishimaru, Y. Utsumi, and K. Kise, "My reading life: Towards utilizing eyetracking on unmodified tablets and phones", in *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, ser. UbiComp '13 Adjunct, Zurich, Switzerland: ACM, 2013, pp. 283–286, ISBN: 978-1-4503-2215-7. DOI: 10.1145/2494091.2494179. [Online]. Available: http://doi.acm.org/10.1145/2494091.2494179.

[38] M. Swan, "The quantified self: Fundamental disruption in big data science and biological discovery", *Big Data*, vol. 1, no. 2, pp. 85–99, 2013.

[39] Z. Lu and K. Grauman, "Story-driven summarization for egocentric video", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2714–2721.

[40] C. Lander, A. Krüger, and M. Löchtefeld, "The story of life is quicker than the blink of an eye: Using corneal imaging for life logging", in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, ser. UbiComp '16, Heidelberg, Germany: ACM, 2016, pp. 1686–1695, ISBN: 978-1-4503-4462-3. DOI: 10.1145/2968219.2968337. [Online]. Available: http://doi.acm.org/10.1145/2968219.2968337.

[41] C. Braunagel, E Kasneci, W Stolzmann, and W. Rosenstiel, "Driver-activity recognition in the context of conditionally autonomous driving", in *IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*, 2015.

[42] F. Jungwirth, M. Murauer, M. Haslgrübler, and A. Ferscha, "Eyes are different than hands: An analysis of gaze as input modality for industrial man-machine interactions", in *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*, ser. PETRA '18, Corfu, Greece: ACM, 2018, pp. 303–310, ISBN: 978-1-4503-6390-7. DOI: 10.1145/3197768.3201565. [Online]. Available: http://doi.acm.org/10.1145/3197768.3201565.

[43] J. L. Rosch and J. J. Vogel-Walcutt, "A review of eye-tracking applications as tools for training", *Cognition, technology & work*, vol. 15, no. 3, pp. 313–327, 2013.

[44] A. T. Duchowski, V. Shivashankaraiah, T. Rawls, A. K. Gramopadhye, B. J. Melloy, and B. Kanki, "Binocular eye tracking in virtual reality for inspection training", in *Proceedings of the Eye Tracking Research & Application Symposium, ETRA 2000, Palm Beach Gardens, Florida, USA, November 6-8, 2000*, 2000, pp. 89–96. DOI: 10.1145/355017.355031. [Online]. Available: https://doi.org/10.1145/355017.355031.

[45] S. Mann, "Surveillance (oversight), sousveillance (undersight), and metaveillance (seeing sight itself)", in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on*, IEEE, 2016, pp. 1408–1417.

[46] A. T. Duchowski, N. Cournia, and H. Murphy, "Gaze-contingent displays: A review", *CyberPsychology & Behavior*, vol. 7, no. 6, pp. 621–634, 2004.

[47] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer, *Eye tracking: A comprehensive guide to methods and measures*. Oxford University Press, 2011.

[48] A. T. Duchowski, "Eye tracking methodology - theory and practice, third edition", 2017. DOI: 10.1007/978-3-319-57883-5. [Online]. Available: https://doi.org/10.1007/978-3-319-57883-5.

[49] ——, "A breadth-first survey of eye-tracking applications", *Behavior Research Methods, Instruments, & Computers*, vol. 34, no. 4, pp. 455–470, 2002, ISSN: 1532-5970. DOI: 10.3758/BF03195475. [Online]. Available: https://doi.org/10.3758/BF03195475.

[50] W. Fuhl, M. Tonsen, A. Bulling, and E. Kasneci, "Pupil detection for head-mounted eye tracking in the wild: An evaluation of the state of the art", *Machine Vision and Applications*, vol. 27, no. 8, pp. 1275–1288, 2016.

[51] C. H. Morimoto and M. R. Mimica, "Eye gaze tracking techniques for interactive applications", *Computer vision and image understanding*, vol. 98, no. 1, pp. 4–24, 2005.

[52] K. Pfeuffer, M. Vidal, J. Turner, A. Bulling, and H. Gellersen, "Pursuit calibration: Making gaze calibration less tedious and more flexible", in *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '13, St. Andrews, Scotland, United Kingdom: ACM, 2013, pp. 261–270, ISBN: 978-1-4503-2268-3. DOI: 10.1145/2501988.2501998. [Online]. Available: http://doi.acm.org/10.1145/2501988.2501998.

[53] T. Santini, H. Brinkmann, L. Reitstätter, H. Leder, R. Rosenberg, W. Rosenstiel, and E. Kasneci, "The art of pervasive eye tracking: Unconstrained eye tracking in the austrian gallery belvedere", in *Proceedings of the 7th Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction*, ser. PETMEI '18, Warsaw, Poland: ACM, 2018, 5:1–5:8, ISBN: 978-1-4503-5789-0. DOI: 10.1145/3208031.3208032. [Online]. Available: http://doi.acm.org/10.1145/3208031.3208032.

[54] T. Santini, T. Kübler, L. Draghetti, P. Gerjets, W. Wagner, U. Trautwein, and E. Kasneci, "Automatic mapping of remote crowd gaze to stimuli in the classroom", in *Proceedings of the Eye Tracking Enhanced Learning Workshop*, 2017.

[55] R. M. Aronson, T. Santini, T. C. Kübler, E. Kasneci, S. Srinivasa, and H. Admoni, "Eye-hand behavior in human-robot shared manipulation", in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '18, Chicago, IL, USA: ACM, 2018, pp. 4–13, ISBN: 978-1-4503-4953-6. DOI: 10.1145/3171221.3171287. [Online]. Available: http://doi.acm.org/10.1145/3171221.3171287.

[56] W. Fuhl, T. Santini, and E. Kasneci, "Fast camera focus estimation for gaze-based focus control", *CoRR*, vol. abs/1711.03306, 2017. arXiv: 1711.03306. [Online]. Available: http://arxiv.org/abs/1711.03306.

[57] S. Eivazi, T. Santini, A. Keshavarzi, T. Kübler, and A. Mazzei, "Improving real-time cnn-based pupil detection through domain-specific data augmentation", in *Proceedings of the 2019 ACM Symposium on Eye Tracking Research & Applications – To appear*, ser. ETRA '19, New York, NY, USA: ACM, 2019.

[58] W. Fuhl, D. Geisler, T. Santini, T. Appel, W. Rosenstiel, and E. Kasneci, "Cbf: Circular binary features for robust and real-time pupil center detection", in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '18, Warsaw, Poland: ACM, 2018, 8:1–8:6, ISBN: 978-1-4503-5706-7. DOI: 10.1145/3204493.3204559. [Online]. Available: http://doi.acm.org/10.1145/3204493.3204559.

[59] W. Fuhl, T. Santini, and E. Kasneci, "Fast and robust eyelid outline and aperture detection in real-world scenarios", in *2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017, Santa Rosa, CA, USA, March 24-31, 2017*, 2017, pp. 1089–1097. DOI: 10.1109/WACV.2017.126. [Online]. Available: https://doi.org/10.1109/WACV.2017.126.

[60] W. Fuhl, T. Santini, C. Reichert, D. Claus, A. Herkommer, H. Bahmani, K. Rifai, S. Wahl, and E. Kasneci, "Non-intrusive practitioner pupil detection for unmodified microscope oculars", *Computers in biology and medicine*, vol. 79, pp. 36–44, 2016.

[61] T. Appel, T. Santini, and E. Kasneci, "Brightness- and motion-based blink detection for head-mounted eye trackers", in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, ser. UbiComp '16, Heidelberg, Germany: ACM, 2016, pp. 1726–1735, ISBN: 978-1-4503-4462-3. DOI: 10.1145/2968219.2968341. [Online]. Available: http://doi.acm.org/10.1145/2968219.2968341.

[62] W. Fuhl, D. Geisler, T. Santini, W. Rosenstiel, and E. Kasneci, "Evaluation of state-of-the-art pupil detection algorithms on remote eye images", in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, ser. UbiComp '16, Heidelberg, Germany: ACM, 2016, pp. 1716–1725, ISBN: 978-1-4503-4462-3. DOI: 10.1145/2968219.2968340. [Online]. Available: http://doi.acm.org/10.1145/2968219.2968340.

[63] W. Fuhl, T. Santini, D. Geisler, T. Kübler, W. Rosenstiel, and E. Kasneci, "Eyes wide open? eyelid location and eye aperture estimation for pervasive eye tracking in real-world scenarios", in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, ser. UbiComp '16, Heidelberg, Germany: ACM, 2016, pp. 1656–1665, ISBN: 978-1-4503-4462-3. DOI: 10.1145/2968219.2968334. [Online]. Available: http://doi.acm.org/10.1145/2968219.2968334.

[64] W. Fuhl, T. Santini, G. Kasneci, W. Rosenstiel, and E. Kasneci, "Pupilnet v2.0: Convolutional neural networks for CPU based real time robust pupil detection", *CoRR*, vol. abs/1711.00112, 2017. arXiv: 1711.00112. [Online]. Available: http://arxiv.org/abs/1711.00112.

[65] W. Fuhl, T. Santini, G. Kasneci, and E. Kasneci, "Pupilnet: Convolutional neural networks for robust pupil detection", *CoRR*, vol. abs/1601.04902, 2016. arXiv: 1601.04902. [Online]. Available: http://arxiv.org/abs/1601.04902.

[66] T. Santini, W. Fuhl, T. Kübler, and E. Kasneci, "Bayesian identification of fixations, saccades, and smooth pursuits", in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '16, Charleston, South Carolina: ACM, 2016, pp. 163–170, ISBN: 978-1-4503-4125-7. DOI: 10.1145/2857491.2857512. [Online]. Available: http://doi.acm.org/10.1145/2857491.2857512.

[67] T. Santini, W. Fuhl, D. Geisler, and E. Kasneci, "Eyerectoo: Open-source software for real-time pervasive head-mounted eye tracking", in *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 6: VISAPP, (VISIGRAPP 2017)*, INSTICC, SciTePress, 2017, pp. 96–101, ISBN: 978-989-758-227-1. DOI: 10.5220/0006224700960101.

[68] D. Geisler, W. Fuhl, T. Santini, and E. Kasneci, "Saliency sandbox - bottom-up saliency framework", in *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017) - Volume 4: VISAPP, Porto, Portugal, February 27 - March 1, 2017.*, 2017, pp. 657–664. DOI: 10.5220/0006272306570664. [Online]. Available: https://doi.org/10.5220/0006272306570664.

[69] W. Fuhl, T. Santini, D. Geisler, T. C. Kübler, and E. Kasneci, "Eyelad: Remote eye tracking image labeling tool - supportive eye, eyelid and pupil labeling tool for remote eye tracking videos", in *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017) - Volume 5: VISAPP, Porto, Portugal, February 27 - March 1, 2017.*, 2017, pp. 405–410.

[70] T. Santini, W. Fuhl, T. C. Kübler, and E. Kasneci, "Eyerec: An open-source data acquisition software for head-mounted eye-tracking", in *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 3: VISAPP, Rome, Italy, February 27-29, 2016.*, 2016, pp. 386–391. DOI: 10.5220/0005758903840389. [Online]. Available: https://doi.org/10.5220/0005758903840389.

[71] Aristoteles, W. R. Roberts, W. Ross, J. A. Smith, I. Bywater, and E. S. Forster, *The works of Aristotle*. Clarendon, 1927.

[72] N. J. Wade, "Pioneers of eye movement research", *I-Perception*, vol. 1, no. 2, pp. 33–68, 2010, PMID: 23396982. DOI: 10.1068/i0389. eprint: https://doi.org/10.1068/i0389. [Online]. Available: https://doi.org/10.1068/i0389.

[73] W. Wells, "Reply to dr. darwin on vision", *The Gentleman's Magazine and Historical Chronicle*, vol. 64, pp. 905–907, 1794.

[74] F. C. Donders, "Over de snelheid van psychische processen", *Onderzoekingen gedaan in het Physiologisch Laboratorium der Utrechtse Hoogeschool, Tweede Reeks II*, 1868.

[75] E. Hering, *Uber Muskelgerausche des Auges*. 1879.

[76] E. B. Huey, "On the psychology and physiology of reading. i", *The American Journal of Psychology*, vol. 11, no. 3, pp. 283–302, 1900.

[77] A. Ahrens, *Untersuchungen über die Bewegung der Augen beim Schreiben*. C. Boldt, 1891.

[78] E Rählmann, "Über den nystagmus und seine ätiologie", *Albrecht von Græfes Archiv für Ophthalmologie*, vol. 24, no. 4, pp. 237–317, 1878.

[79] E. B. Delabarre, "A method of recording eye-movements", *The American Journal of Psychology*, vol. 9, no. 4, pp. 572–574, 1898.

[80] E. B. Huey, "Preliminary experiments in the physiology and psychology of reading", *The American Journal of Psychology*, vol. 9, no. 4, pp. 575–586, 1898.

[81] D. C. Richardson and M. J. Spivey, "Eye tracking: Research areas and applications", *Encyclopedia of biomaterials and biomedical engineering*, pp. 573–582, 2004.

[82] L. R. Young and D. Sheena, "Survey of eye movement recording methods", *Behavior research methods & instrumentation*, vol. 7, no. 5, pp. 397–429, 1975.

[83] R. Dodge, "An experimental study of visual fixation.", *The Psychological Review: Monograph Supplements*, vol. 8, no. 4, p. i, 1907.

[84] E. Schott, "Über die registrierung des nystagmus und anderer augenbewegungen vermitteles des saitengalvanometers", *Deutsches Archiv für klinische Medizin*, vol. 140, pp. 79–90, 1922.

[85] N. Torok, V. Guillemin Jr, and J. Barnothy, "Photoelectric nystagmography", *Annals of Otology, Rhinology & Laryngology*, vol. 60, no. 4, pp. 917–926, 1951.

[86] D. A. Robinson, "A method of measuring eye movemnent using a scleral search coil in a magnetic field", *IEEE Transactions on bio-medical electronics*, vol. 10, no. 4, pp. 137–145, 1963, ISSN: 0096-0616. DOI: 10.1109/TBMEL.1963.4322822.

[87] P. M. Fitts, R. E. Jones, and J. L. Milton, "Eye movements of aircraft pilots during instrument-landing approaches.", *Aeronautical Engineering Review*, vol. 9, no. 2, pp. 24–29, 1950.

[88] R. Zemblys and O. Komogortsev, "Making stand-alone ps-og technology tolerant to the equipment shifts", in *Proceedings of the 7th Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction*, ser. PETMEI '18, Warsaw, Poland: ACM, 2018, 2:1–2:9, ISBN: 978-1-4503-5789-0. DOI: 10.1145/3208031.3208035. [Online]. Available: http://doi.acm.org/10.1145/3208031.3208035.

[89] N. H. Mackworth and E. L. Thomas, "Head-mounted eye-marker camera", *JOSA*, vol. 52, no. 6, pp. 713–716, 1962.

[90] J. F. Mackworth and N. Mackworth, "Eye fixations recorded on changing visual scenes by the television eye-marker", *JOSA*, vol. 48, no. 7, pp. 439–445, 1958.

[91] W. J. Ryan, A. T. Duchowski, and S. T. Birchfield, "Limbus/pupil switching for wearable eye tracking under variable lighting conditions", in *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ser. ETRA '08, Savannah, Georgia: ACM, 2008, pp. 61–64, ISBN: 978-1-59593-982-1. DOI: 10.1145/1344471.1344487. [Online]. Available: http://doi.acm.org/10.1145/1344471.1344487.

[92] C. H. Morimoto, D. Koons, A. Amir, and M. Flickner, "Pupil detection and tracking using multiple light sources", *Image and vision computing*, vol. 18, no. 4, pp. 331–335, 2000.

[93] Y. 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*, IEEE, 1993, pp. 1268–1269.

[94] Tobii Technology, Accessed: 2019-01-23. [Online]. Available: https://www.tobiipro.com/learn-and-support/learn/eye-tracking-essentials/what-is-dark-and-bright-pupil-tracking/.

[95] J. Merchant, R. Morrissette, and J. L. Porterfield, "Remote measurement of eye direction allowing subject motion over one cubic foot of space", *IEEE Transactions on Biomedical Engineering*, no. 4, pp. 309–317, 1974.

[96] D. Mardanbegi, A. T. Kurauchi, and C. H. Morimoto, "An investigation of the distribution of gaze estimation errors in head mounted gaze trackers using polynomial functions", *Journal of Eye Movement Research*, vol. 11, no. 3, pp. 1–14, 2018.

[97] M. Mansouryar, J. Steil, Y. Sugano, and A. Bulling, "3d gaze estimation from 2d pupil positions on monocular head-mounted eye trackers", in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '16, Charleston, South Carolina: ACM, 2016, pp. 197–200, ISBN: 978-1-4503-4125-7. DOI: 10.1145/2857491.2857530. [Online]. Available: http://doi.acm.org/10.1145/2857491.2857530.

[98] K. Harezlak, P. Kasprowski, and M. Stasch, "Towards accurate eye tracker calibration–methods and procedures", *Procedia Computer Science*, vol. 35, pp. 1073–1081, 2014.

[99] P. Blignaut and D. Wium, "The effect of mapping function on the accuracy of a video-based eye tracker", in *Proceedings of the 2013 Conference on Eye Tracking South Africa*, ser. ETSA '13, Cape Town, South Africa: ACM, 2013, pp. 39–46, ISBN: 978-1-4503-2110-5. DOI: 10.1145/2509315.2509321. [Online]. Available: http://doi.acm.org/10.1145/2509315.2509321.

[100] J. J. Cerrolaza, A. Villanueva, and R. Cabeza, "Study of polynomial mapping functions in video-oculography eye trackers", *ACM Trans. Comput.-Hum. Interact.*, vol. 19, no. 2, 10:1–10:25, Jul. 2012, ISSN: 1073-0516. DOI: 10.1145/2240156.2240158. [Online]. Available: http://doi.acm.org/10.1145/2240156.2240158.

[101] L. Sesma-Sanchez, Y. Zhang, A. Bulling, and H. Gellersen, "Gaussian processes as an alternative to polynomial gaze estimation functions", in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '16, Charleston, South Carolina: ACM, 2016, pp. 229–232, ISBN: 978-1-4503-4125-7. DOI: 10.1145/2857491.2857509. [Online]. Available: http://doi.acm.org/10.1145/2857491.2857509.

[102] D. W. Hansen, J. P. Hansen, M. Nielsen, A. S. Johansen, and M. B. Stegmann, "Eye typing using markov and active appearance models", in *6th IEEE Workshop on Applications of Computer Vision (WACV 2002), 3-4 December 2002, Orlando, FL, USA*, 2002, pp. 132–136. DOI: 10.1109/ACV.2002.1182170. [Online]. Available: https://doi.org/10.1109/ACV.2002.1182170.

[103] J. Wang, G. Zhang, and J. Shi, "2d gaze estimation based on pupil-glint vector using an artificial neural network", *Applied Sciences*, vol. 6, no. 6, p. 174, 2016.

[104] M. Gneo, M. Schmid, S. Conforto, and T. D'Alessio, "A free geometry model-independent neural eye-gaze tracking system", *Journal of neuroengineering and rehabilitation*, vol. 9, no. 1, p. 82, 2012.

[105] M. Tonsen, J. Steil, Y. Sugano, and A. Bulling, "Invisibleeye: Mobile eye tracking using multiple low-resolution cameras and learning-based gaze estimation", *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 3, 106:1–106:21, Sep. 2017, ISSN: 2474-9567. DOI: 10.1145/3130971. [Online]. Available: http://doi.acm.org/10.1145/3130971.

[106] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, "Learning an appearance-based gaze estimator from one million synthesised images", in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '16, Charleston, South Carolina: ACM, 2016, pp. 131–138, ISBN: 978-1-4503-4125-7. DOI: 10.1145/2857491.2857492. [Online]. Available: http://doi.acm.org/10.1145/2857491.2857492.

[107] A. Mayberry, P. Hu, B. Marlin, C. Salthouse, and D. Ganesan, "Ishadow: Design of a wearable, real-time mobile gaze tracker", in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '14, Bretton Woods, New Hampshire, USA: ACM, 2014, pp. 82–94, ISBN: 978-1-4503-2793-0. DOI: 10.1145/2594368.2594388. [Online]. Available: http://doi.acm.org/10.1145/2594368.2594388.

[108] E. Wood, T. Baltrusaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling, "Rendering of eyes for eye-shape registration and gaze estimation", in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3756–3764.

[109] L. Świrski and N. Dodgson, "Rendering synthetic ground truth images for eye tracker evaluation", in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA '14, Safety Harbor, Florida: ACM, 2014, pp. 219–222, ISBN: 978-1-4503-2751-0. DOI: 10.1145/2578153.2578188. [Online]. Available: http://doi.acm.org/10.1145/2578153.2578188.

[110] L. H. Yu and M. Eizenman, "A new methodology for determining point-of-gaze in head-mounted eye tracking systems", *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 10, pp. 1765–1773, 2004. DOI: 10.1109/TBME.2004.831523. [Online]. Available: http://dx.doi.org/10.1109/TBME.2004.831523.

[111] L. Świrski and N. A. Dodgson, "A fully-automatic, temporal approach to single camera, glint-free 3d eye model fitting [abstract]", in *Proceedings of ECEM 2013*, Lund, Sweden, Aug. 2013.

[112] A. Tsukada and T. Kanade, "Automatic acquisition of a 3d eye model for a wearable first-person vision device", in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA '12, Santa Barbara, California: ACM, 2012, pp. 213–216, ISBN: 978-1-4503-1221-9. DOI: 10.1145/2168556.2168597. [Online]. Available: http://doi.acm.org/10.1145/2168556.2168597.

[113] S. Kohlbecher, S. Bardinst, K. Bartl, E. Schneider, T. Poitschke, and M. Ablassmeier, "Calibration-free eye tracking by reconstruction of the pupil ellipse in 3d space", in *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ser. ETRA '08, Savannah, Georgia: ACM, 2008, pp. 135–138, ISBN: 978-1-59593-982-1. DOI: 10.1145/1344471.1344506. [Online]. Available: http://doi.acm.org/10.1145/1344471.1344506.

[114] S.-W. Shih, Y.-T. Wu, and J. Liu, "A calibration-free gaze tracking technique", in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, IEEE, vol. 4, 2000, pp. 201–204.

[115] A. Villanueva and R. Cabeza, "Models for gaze tracking systems", *Journal on Image and Video Processing*, vol. 2007, no. 3, p. 4, 2007.

[116] A. Villanueva, J. J. Cerrolaza, and R. Cabeza, "Geometry issues of gaze estimation", in *Advances in Human Computer Interaction*, InTech, 2008.

[117] X. L. Brolly and J. B. Mulligan, "Implicit calibration of a remote gaze tracker", in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, IEEE, 2004, pp. 134–134.

[118] D. Geisler, D. Fox, and E. Kasneci, "Real-time 3d glint detection in remote eye tracking based on bayesian inference", in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 7119–7126.

[119] M. Toivanen, K. Lukander, K. Puolamäki, *et al.*, "Probabilistic approach to robust wearable gaze tracking", *Journal of Eye Movement Research*, 2017.

[120] K. Holmqvist, D. C. Niehorster, and P. Blignaut, "Data quality in eye trackers: Signal resolution", in *The Scandinavian Workshop on Applied Eye Tracking 2018*, 2018.

[121] I. Hooge, K. Holmqvist, and M. Nyström, "The pupil is faster than the corneal reflection (cr): Are video based pupil-cr eye trackers suitable for studying detailed dynamics of eye movements?", *Vision research*, vol. 128, pp. 6–18, 2016.

[122] R. H. Carpenter, *Movements of the Eyes, 2nd Rev.* Pion Limited, 1988.

[123] A. Villanueva and R. Cabeza, "A novel gaze estimation system with one calibration point", *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 38, no. 4, pp. 1123–1138, 2008. DOI: 10.1109/TSMCB.2008.926606. [Online]. Available: http://dx.doi.org/10.1109/TSMCB.2008.926606.

[124] Pupil Labs, Accessed in 2018-08-10. [Online]. Available: https://pupil-labs.com/.

[125] H. S. Raffle and C.-J. Wang, *Heads up display*, US Patent 9,001,030, 2015. [Online]. Available: https://patents.google.com/patent/US9001030.

[126] Microsoft, *Microsoft hololens*, Accessed in 2018-08-10. [Online]. Available: https://www.microsoft.com/en-us/hololens.

[127] Oculus, *Oculus rift*, Accessed in 2018-08-10. [Online]. Available: https://www.oculus.com/rift/.

[128] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder, "Foveated 3d graphics", *ACM Trans. Graph.*, vol. 31, no. 6, 164:1–164:10, Nov. 2012, ISSN: 0730-0301. DOI: 10.1145/2366145.2366183. [Online]. Available: http://doi.acm.org/10.1145/2366145.2366183.

[129] J. Schmidt, R. Laarousi, W. Stolzmann, and K. Karrer-Gauß, "Eye blink detection for different driver states in conditionally automated driving and manual driving using eog and a driver camera", *Behavior Research Methods*, pp. 1–14, 2017.

[130] J. M. Wood, R. A. Tyrrell, P. Lacherez, and A. A. Black, "Night-time pedestrian conspicuity: Effects of clothing on drivers' eye movements", *Ophthalmic and physiological optics*, vol. 37, no. 2, pp. 184–190, 2017.

[131] T. C. Kübler, E. Kasneci, W. Rosenstiel, M. Heister, K. Aehling, K. Nagel, U. Schiefer, and E. Papageorgiou, "Driving with glaucoma: Task performance and gaze movements", *Optometry & Vision Science*, vol. 92, no. 11, pp. 1037–1046, 2015.

[132] S. Trösterer, A. Meschtscherjakov, D. Wilfinger, and M. Tscheligi, "Eye tracking in the car: Challenges in a dual-task scenario on a test track", in *Adjunct Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, ser. AutomotiveUI '14, Seattle, WA, USA: ACM, 2014, pp. 1–6, ISBN: 978-1-4503-0725-3. DOI: 10.1145/2667239.2667277. [Online]. Available: http://doi.acm.org/10.1145/2667239.2667277.

[133] E. Kasneci, "Towards the automated recognition of assistance need for drivers with impaired visual field", PhD thesis, Universität Tübingen, Germany, 2013.

[134] B. S. Chu, J. M. Wood, and M. J. Collins, "The effect of presbyopic vision corrections on nighttime driving performance", *Investigative ophthalmology & visual science*, vol. 51, no. 9, pp. 4861–4866, 2010.

[135] E. Kasneci, K. Sippel, M. Heister, K. Aehling, W. Rosenstiel, U. Schiefer, and E. Papageorgiou, "Homonymous visual field loss and its impact on visual exploration: A supermarket study", *Translational vision science & technology*, vol. 3, no. 6, pp. 2–2, 2014.

[136] Y. Sugano and A. Bulling, "Self-calibrating head-mounted eye trackers using egocentric visual saliency", in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, ser. UIST '15, Daegu, Kyungpook, Republic of Korea: ACM, 2015, pp. 363–372, ISBN: 978-1-4503-3779-3. DOI: 10.1145/2807442.2807445. [Online]. Available: http://doi.acm.org/10.1145/2807442.2807445.

[137] T. Foulsham, E. Walker, and A. Kingstone, "The where, what and when of gaze allocation in the lab and the natural environment", *Vision research*, vol. 51, no. 17, pp. 1920–1931, 2011.

[138] T. Tien, P. H. Pucher, M. H. Sodergren, K. Sriskandarajah, G.-Z. Yang, and A. Darzi, "Differences in gaze behaviour of expert and junior surgeons performing open inguinal hernia repair", *Surgical endoscopy*, vol. 29, no. 2, pp. 405–413, 2015.

[139] D. W. Hansen and R. I. Hammoud, "An improved likelihood model for eye tracking", *Computer Vision and Image Understanding*, vol. 106, no. 2, pp. 220–230, 2007.

[140] D. W. Hansen and A. E. Pece, "Eye tracking in the wild", *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 155–181, 2005.

[141] Z. Zhu and Q. Ji, "Robust real-time eye detection and tracking under variable lighting conditions and various face orientations", *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 124–154, 2005.

[142] L. Świrski, A. Bulling, and N. Dodgson, "Robust real-time pupil tracking in highly off-axis images", in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA '12, Santa Barbara, California: ACM, 2012, pp. 173–176, ISBN: 978-1-4503-1221-9. DOI: 10.1145/2168556.2168585. [Online]. Available: http://doi.acm.org/10.1145/2168556.2168585.

[143] J. Canny, "A computational approach to edge detection", *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[144] G. J. Mohammed, B. R. Hong, and A. A. Jarjes, "Accurate pupil features extraction based on new projection function", *Computing and Informatics*, vol. 29, no. 4, pp. 663–680, 2012.

[145] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, IEEE, vol. 1, 2001, pp. I–I.

[146] S. Lloyd, "Least squares quantization in pcm", *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[147] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "Cudnn: Efficient primitives for deep learning", *CoRR*, vol. abs/1410.0759, 2014. arXiv: 1410.0759. [Online]. Available: http://arxiv.org/abs/1410.0759.

[148] G. Efland, S. Parikh, H. Sanghavi, and A. Farooqui, "High performance dsp for vision, imaging and neural networks", *IEEE Hot Chips*, vol. 28, 2016.

[149] F. Vera-Olmos and N Malpica, "Deconvolutional neural network for pupil detection in real-world environments", in *International Work-Conference on the Interplay Between Natural and Artificial Computation*, Springer, 2017, pp. 223–231.

[150] C.-H. Teh and R. T. Chin, "On the detection of dominant points on digital curves", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 11, no. 8, pp. 859–872, 1989.

[151] J. Kunjur, T Sabesan, and V Ilankovan, "Anthropometric analysis of eyebrows and eyelids: An inter-racial study", *British Journal of Oral and Maxillofacial Surgery*, vol. 44, no. 2, pp. 89–93, 2006.

[152] R. Spector, "The pupils", in *Clinical Methods: The HIstory, Physical, and Laboratory Examinations*, H. J. Walker HK Hall WD, Ed., Butterworths, 1990, ch. 8.

[153] G. T. Toussaint, "Solving geometric problems with the rotating calipers", in *Proc. IEEE Melecon*, vol. 83, 1983, A10.

[154] A. W. Fitzgibbon and R. B. Fisher, "A buyer's guide to conic fitting", in *Proceedings of the 6th British Conference on Machine Vision (Vol. 2)*, ser. BMVC '95, Birmingham, United Kingdom: BMVA Press, 1995, pp. 513–522, ISBN: 0-9521898-2-8. [Online]. Available: http://dl.acm.org/citation.cfm?id=243124.243148.

[155] M. Frigge, D. C. Hoaglin, and B. Iglewicz, "Some implementations of the boxplot", *The American Statistician*, vol. 43, no. 1, pp. 50–54, 1989.

[156] M. Pedrotti, S. Lei, J. Dzaack, and M. Rötting, "A data-driven algorithm for offline pupil signal preprocessing and eyeblink detection in low-speed eye-tracking protocols", *Behavior Research Methods*, vol. 43, no. 2, pp. 372–383, 2011.

[157] H. Vrzakova and R. Bednarik, "Hard lessons learned: Mobile eye-tracking in cockpits", in *Proceedings of the 4th Workshop on Eye Gaze in Intelligent Human Machine Interaction*, ser. Gaze-In '12, Santa Monica, California: ACM, 2012, 7:1–7:6, ISBN: 978-1-4503-1516-6. DOI: 10.1145/2401836.2401843. [Online]. Available: http://doi.acm.org/10.1145/2401836.2401843.

[158] S. Jansen, H Kingma, and R. Peeters, "A confidence measure for real-time eye movement detection in video-oculography", in *13th International Conference on Biomedical Engineering*, Springer, 2009, pp. 335–339.

[159] F. Bashir and F. Porikli, "Performance evaluation of object detection and tracking systems", in *Proceedings 9th IEEE International Workshop on PETS*, 2006, pp. 7–14.

[160] Ergoneers, *Dikablis glasses professional*, Accessed in 2018-08-10. [Online]. Available: http://www.ergoneers.com/en/hardware/eye-tracking/.

[161] L. Čehovin, A. Leonardis, and M. Kristan, "Visual object tracking performance measures revisited", *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1261–1274, 2016.

[162] M. Kristan, J. Matas, A. Leonardis, T. Vojíř, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin, "A novel performance evaluation methodology for single-target trackers", *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 11, pp. 2137–2155, 2016.

[163] C. Pheatt, "Intel® threading building blocks", *Journal of Computing Sciences in Colleges*, vol. 23, no. 4, pp. 298–298, 2008.

[164] D. Li, D. Winfield, and D. J. Parkhurst, "Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches", in *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, IEEE, 2005, pp. 79–79.

[165] M. Kassner, W. Patera, and A. 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*, ser. UbiComp '14 Adjunct, Seattle, Washington: ACM, 2014, pp. 1151–1160, ISBN: 978-1-4503-3047-3. DOI: 10.1145/2638728.2641695. [Online]. Available: http://doi.acm.org/10.1145/2638728.2641695.

[166] Pupil Labs, Accessed in 2018-08-10. [Online]. Available: https://pupil-labs.com/blog/2016-03/pupil-v0-7-release-notes/.

[167] D. K. Prasad and M. K. Leung, "Methods for ellipse detection from edge maps of real images", in *Machine Vision-Applications and Systems*, InTech, 2012.

[168] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following", *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.

[169] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature", *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[170] J. Sklansky, "Finding the convex hull of a simple polygon", *Pattern Recognition Letters*, vol. 1, no. 2, pp. 79–83, 1982.

[171] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[172] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration.", in *ECCV Workshops (2)*, 2014, pp. 254–265.

[173] Z. Kalal *et al.*, "Tracking-learning-detection", *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[174] I. Martinikorena, R. Cabeza, A. Villanueva, I. Urtasun, and A. Larumbe, "Fast and robust ellipse detection algorithm for head-mounted eye tracking systems", *Machine Vision and Applications*, pp. 1–16, 2018.

[175] J. Li, S. Li, T. Chen, and Y. Liu, "A geometry-appearance-based pupil detection method for near-infrared head-mounted cameras", *IEEE Access*, vol. 6, pp. 23 242–23 252, 2018.

[176] K. Wolski and R. Mantiuk, "Cross spread pupil tracking technique", *Journal of Electronic Imaging*, vol. 25, no. 6, p. 063 012, 2016.

[177] A.-H. Javadi, Z. Hakimi, M. Barati, V. Walsh, and L. Tcheang, "Set: A pupil detection method using sinusoidal approximation", *Frontiers in neuroengineering*, vol. 8, p. 4, 2015.

[178] F. Vera-Olmos, E Pardo, H Melero, and N Malpica, "Deepeye: Deep convolutional network for pupil detection in real environments", *Integrated Computer-Aided Engineering*, vol. 26, no. 1, pp. 85–95, 2019.

[179] W. Fuhl, S. Eivazi, B. Hosp, A. Eivazi, W. Rosenstiel, and E. Kasneci, "Bore: Boosted-oriented edge optimization for robust, real time remote pupil center detection", in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '18, Warsaw, Poland: ACM, 2018, 48:1–48:5, ISBN: 978-1-4503-5706-7. DOI: 10.1145/3204493.3204558. [Online]. Available: http://doi.acm.org/10.1145/3204493.3204558.

[180] Y. Zhu, W. Chen, X. Zhan, Z. Guo, H. Shi, and I. G. Harris, "Head mounted pupil tracking using convolutional neural network", *CoRR*, vol. abs/1805.00311, 2018, Withdrawn. arXiv: 1805.00311. [Online]. Available: http://arxiv.org/abs/1805.00311.

[181] N. Kan, N. Kondo, W. Chinsatit, and T. Saitoh, "Effectiveness of data augmentation for cnn-based pupil center point detection", in *2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, IEEE, 2018, pp. 41–46.

[182] W. Chinsatit and T. Saitoh, "Cnn-based pupil center detection for wearable gaze estimation system", *Applied Computational Intelligence and Soft Computing*, vol. 2017, 2017.

[183] D. G. d. G. Pérez and R. Bednarik, "Crowdpupil: A crowdsourced, pupil-center annotated image dataset", *Journal of Eye Movement Research: Abstracts of the 19th European Conference on Eye Movements 2017*, vol. 10, no. 6, 2017.

[184] F. Hausdorff, *Mengenlehre*. Walter de Gruyter Berlin, 1927.

[185] A. M. Feit, S. Williams, A. Toledo, A. Paradiso, H. Kulkarni, S. Kane, and M. R. Morris, "Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design", in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17, Denver, Colorado, USA: ACM, 2017, pp. 1118–1130, ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025599. [Online]. Available: http://doi.acm.org/10.1145/3025453.3025599.

[186] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion", *Pattern Recognition*, vol. 47, no. 6, pp. 2280 –2292, 2014, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2014.01.005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320314000235.

[187] M. Barz, F. Daiber, and A. Bulling, "Prediction of gaze estimation error for error-aware gaze-based interfaces", in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '16, Charleston, South Carolina: ACM, 2016, pp. 275–278, ISBN: 978-1-4503-4125-7. DOI: 10.1145/2857491.2857493. [Online]. Available: http://doi.acm.org/10.1145/2857491.2857493.

[188] J. Steil, M. X. Huang, and A. Bulling, "Fixation detection for head-mounted eye tracking based on visual similarity of gaze targets", in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '18, Warsaw, Poland: ACM, 2018, 23:1–23:9, ISBN: 978-1-4503-5706-7. DOI: 10.1145/3204493.3204538. [Online]. Available: http://doi.acm.org/10.1145/3204493.3204538.

[189] M. Nyström, R. Andersson, K. Holmqvist, and J. Van De Weijer, "The influence of calibration method and eye physiology on eyetracking data quality", *Behavior research methods*, vol. 45, no. 1, pp. 272–288, 2013.

[190] C. Lander, S. Gehring, A. Krüger, S. Boring, and A. Bulling, "Gazeprojector: Accurate gaze estimation and seamless gaze interaction across multiple displays", in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, ser. UIST '15, Charlotte, NC, USA: ACM, 2015, pp. 395–404, ISBN: 978-1-4503-3779-3. DOI: 10.1145/2807442.2807479. [Online]. Available: http://doi.acm.org/10.1145/2807442.2807479.

[191] K. M. Evans, R. A. Jacobs, J. A. Tarduno, and J. B. Pelz, "Collecting and analyzing eye tracking data in outdoor environments", *Journal of Eye Movement Research*, vol. 5, no. 2, p. 6, 2012.

[192] J. M. Franchak, K. S. Kretch, K. C. Soska, and K. E. Adolph, "Head-mounted eye tracking: A new method to describe infant looking", *Child development*, vol. 82, no. 6, pp. 1738–1750, 2011.

[193] B. L. Welch, "The generalization of student's' problem when several different population variances are involved", *Biometrika*, vol. 34, no. 1/2, pp. 28–35, 1947.

[194] SensoMotoric Instruments GmbH, Accessed: 2018-08-10. [Online]. Available: http://www.eyetracking-glasses.com/products/eye-tracking-glasses-2-wireless/technology/.

[195] Tobii Technology, Accessed: 2018-08-10. [Online]. Available: http://www.tobiipro.com/product-listing/tobii-pro-glasses-2/.

[196] T. C. Kübler, T. Rittig, E. Kasneci, J. Ungewiss, and C. Krauss, "Rendering refraction and reflection of eyeglasses for synthetic eye tracker images", in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '16, Charleston, South Carolina: ACM, 2016, pp. 143–146, ISBN: 978-1-4503-4125-7. DOI: 10.1145/2857491.2857494. [Online]. Available: http://doi.acm.org/10.1145/2857491.2857494.

[197] J. S. Babcock and J. B. Pelz, "Building a lightweight eyetracking headgear", in *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*, ser. ETRA '04, San Antonio, Texas: ACM, 2004, pp. 109–114, ISBN: 1-58113-825-3. DOI: 10.1145/968363.968386. [Online]. Available: http://doi.acm.org/10.1145/968363.968386.

[198] D. Li, J. Babcock, and D. J. Parkhurst, "Openeyes: A low-cost head-mounted eye-tracking solution", in *Proceedings of the 2006 Symposium on Eye Tracking Research &Amp; Applications*, ser. ETRA '06, San Diego, California: ACM, 2006, pp. 95–100, ISBN: 1-59593-305-0. DOI: 10.1145/1117309.1117350. [Online]. Available: http://doi.acm.org/10.1145/1117309.1117350.

[199] J. San Agustin, H. Skovsgaard, E. Mollenbach, M. Barret, M. Tall, D. W. Hansen, and J. P. Hansen, "Evaluation of a low-cost open-source gaze tracker", in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ser. ETRA '10, Austin, Texas: ACM, 2010, pp. 77–80, ISBN: 978-1-60558-994-7. DOI: 10.1145/1743666.1743685. [Online]. Available: http://doi.acm.org/10.1145/1743666.1743685.

[200] M Nyström, R Andersson, K Holmqvist, and J Van de Weijer, "Participants know best–the influence of calibration method and eye physiology on eye tracking data quality", *Journal of Neuroscience Methods*, pp. 1–26, 2011.

[201] K. Essig, M. Pomplun, and H. J. Ritter, "A neural network for 3d gaze recording with binocular eye trackers", *IJPEDS*, vol. 21, no. 2, pp. 79–95, 2006. DOI: 10.1080/17445760500354440. [Online]. Available: http://dx.doi.org/10.1080/17445760500354440.

[202] M. X. Huang, T. C. Kwok, G. Ngai, S. C. Chan, and H. V. Leong, "Building a personalized, auto-calibrating eye tracker from user interactions", in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16, San Jose, California, USA: ACM, 2016, pp. 5169–5179, ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858404. [Online]. Available: http://doi.acm.org/10.1145/2858036.2858404.

[203] S. Bernet, C. Cudel, D. Lefloch, and M. Basset, "Autocalibration-based partioning relationship and parallax relation for head-mounted eye trackers", *Machine Vision and Applications*, vol. 24, no. 2, pp. 393–406, 2013. DOI: 10.1007/s00138-012-0427-3. [Online]. Available: http://dx.doi.org/10.1007/s00138-012-0427-3.

[204] A. J. Hornof and T. Halverson, "Cleaning up systematic error in eye-tracking data by using required fixation locations", *Behavior Research Methods, Instruments, & Computers*, vol. 34, no. 4, pp. 592–604, 2002, ISSN: 1532-5970. DOI: 10.3758/BF03195487. [Online]. Available: http://dx.doi.org/10.3758/BF03195487.

[205] S. M. Kolakowski and J. B. Pelz, "Compensating for eye tracker camera movement", in *Proceedings of the Eye Tracking Research & Application Symposium, ETRA 2006, San Diego, California, USA, March 27-29, 2006*, 2006, pp. 79–85. DOI: 10.1145/1117309.1117348. [Online]. Available: http://doi.acm.org/10.1145/1117309.1117348.

[206] C. Lander, F. Kerber, T. Rauber, and A. Krüger, "A time-efficient re-calibration algorithm for improved long-term accuracy of head-worn eye trackers", in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '16, Charleston, South Carolina: ACM, 2016, pp. 213–216, ISBN: 978-1-4503-4125-7. DOI: 10.1145/2857491.2857513. [Online]. Available: http://doi.acm.org/10.1145/2857491.2857513.

[207] K. Binaee, G. Diaz, J. Pelz, and F. Phillips, "Binocular eye tracking calibration during a virtual ball catching task using head mounted display", in *Proceedings of the ACM Symposium on Applied Perception*, ser. SAP '16, Anaheim, California: ACM, 2016, pp. 15–18, ISBN: 978-1-4503-4383-1. DOI: 10.1145/2931002.2931020. [Online]. Available: http://doi.acm.org/10.1145/2931002.2931020.

[208] S. Garrido-Jurado, R. Munoz-Salinas, F. J. Madrid-Cuevas, and R. Medina-Carnicer, "Generation of fiducial marker dictionaries using mixed integer linear programming", *Pattern Recognition*, vol. 51, pp. 481–491, 2016.

[209] M. Nyström, I. Hooge, and K. Holmqvist, "Post-saccadic oscillations in eye movement data recorded with pupil-based eye trackers reflect motion of the pupil inside the iris", *Vision research*, vol. 92, pp. 59–66, 2013.

[210] W. Becker, "The neurobiology of saccadic eye movements. metrics.", *Reviews of oculomotor research*, vol. 3, p. 13, 1989.

[211] L. Calvet, P. Gurdjos, C. Griwodz, and S. Gasparini, "Detection and accurate localization of circular fiducials under highly challenging conditions", in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 562–570. DOI: 10.1109/CVPR.2016.67. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/CVPR.2016.67.

[212] C. H. Meyer, A. G. Lasker, and D. A. Robinson, "The upper limit of human smooth pursuit velocity", *Vision research*, vol. 25, no. 4, pp. 561–563, 1985.

[213] E. G. Mlot, H. Bahmani, S. Wahl, and E. Kasneci, "3d gaze estimation using eye vergence", in *Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies*, 2016, pp. 125–131, ISBN: 978-989-758-170-0. DOI: 10.5220/0005821201250131.

[214] M. Toivanen, V. Salonen, and M. Hannula, "Self-made mobile gaze tracking for group studies", in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '18, Warsaw, Poland: ACM, 2018, 97:1–97:2, ISBN: 978-1-4503-5706-7. DOI: 10.1145/3204493.3208347. [Online]. Available: http://doi.acm.org/10.1145/3204493.3208347.

[215] K. Lukander, S. Jagadeesan, H. Chi, and K. 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*, ser. MobileHCI '13, Munich, Germany: ACM, 2013, pp. 408–411, ISBN: 978-1-4503-2273-7. DOI: 10.1145/2493190.2493214. [Online]. Available: http://doi.acm.org/10.1145/2493190.2493214.

[216] K. Dierkes, M. Kassner, and A. Bulling, "A novel approach to single camera, glint-free 3d eye model fitting including corneal refraction", in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ser. ETRA '18, Warsaw, Poland: ACM, 2018, 9:1–9:9, ISBN: 978-1-4503-5706-7. DOI: 10.1145/3204493.3204525. [Online]. Available: http://doi.acm.org/10.1145/3204493.3204525.

[217] F. B. Narcizo, *Using Priors to Improve Head-Mounted Eye Trackers in Sports*. IT-Universitetet i København, 2017.

[218] F. Karmali and M. Shelhamer, "Automatic detection of camera translation in eye video recordings using multiple methods", in *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, IEEE, vol. 1, 2004, pp. 1525–1528.

[219] B. Pires, M. Hwangbo, M. Devyver, and T. Kanade, "Visible-spectrum gaze tracking for sports", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 1005–1010.

[220] F. Karmali and M. Shelhamer, "Compensating for camera translation in video eye-movement recordings by tracking a representative landmark selected automatically by a genetic algorithm", *Journal of neuroscience methods*, vol. 176, no. 2, pp. 157–165, 2009.

[221] Z. Yun, Z. Xin-Bo, Z. Rong-Chun, Z. Yuan, and Z. Xiao-Chun, "Eyesecret: An inexpensive but high performance auto-calibration eye tracker", in *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ser. ETRA '08, Savannah, Georgia: ACM, 2008, pp. 103–106, ISBN: 978-1-59593-982-1. DOI: 10.1145/1344471.1344498. [Online]. Available: http://doi.acm.org/10.1145/1344471.1344498.

[222] A. Clarke, J Ditterich, K Drüen, U Schönfeld, and C Steineke, "Using high frame rate cmos sensors for three-dimensional eye tracking", *Behavior Research Methods, Instruments, & Computers*, vol. 34, no. 4, pp. 549–560, 2002.

[223] Pupil Labs, Accessed in 2018-12-31, 2018. [Online]. Available: https://github.com/pupil-labs/pupil/releases?after=v0.8.3.

[224] R. Kredel, C. Vater, A. Klostermann, and E.-J. Hossner, "Eye-tracking technology and the dynamics of natural gaze behavior in sports: A systematic review of 40 years of research", *Frontiers in psychology*, vol. 8, p. 1845, 2017.

[225] L. K. Slone, D. H. Abney, J. I. Borjon, C.-h. Chen, J. M. Franchak, D. Pearcy, C. Suarez-Rivera, T. L. Xu, Y. Zhang, L. B. Smith, *et al.*, "Gaze in action: Head-mounted eye tracking of children's dynamic visual attention during naturalistic behavior", *JoVE (Journal of Visualized Experiments)*, no. 141, e58496, 2018.

[226] R. Safaee-Rad, I. Tchoukanov, K. C. Smith, and B. Benhabib, "Three-dimensional location estimation of circular features for machine vision", *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 624–640, 1992.

[227] F. Schüssel, J. Bäurle, S. Kotzka, M. Weber, F. Pittino, and A. Huckauf, "Design and evaluation of a gaze tracking system for free-space interaction", in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, ser. UbiComp '16, Heidelberg, Germany: ACM, 2016, pp. 1676–1685, ISBN: 978-1-4503-4462-3. DOI: 10.1145/2968219.2968336. [Online]. Available: http://doi.acm.org/10.1145/2968219.2968336.

[228] A. Bulling, J. A. Ward, H. Gellersen, and G. Troster, "Eye movement analysis for activity recognition using electrooculography", *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 4, pp. 741–753, 2011.

[229] Y. Sugano, X. Zhang, and A. Bulling, "Aggregaze: Collective estimation of audience attention on public displays", in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ser. UIST '16, Tokyo, Japan: ACM, 2016, pp. 821–831, ISBN: 978-1-4503-4189-9. DOI: 10.1145/2984511.2984536. [Online]. Available: http://doi.acm.org/10.1145/2984511.2984536.

[230] O. V. Komogortsev and J. I. Khan, "Eye movement prediction by kalman filter with integrated linear horizontal oculomotor plant mechanical model", in *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ser. ETRA '08, Savannah, Georgia: ACM, 2008, pp. 229–236, ISBN: 978-1-59593-982-1. DOI: 10.1145/1344471.1344525. [Online]. Available: http://doi.acm.org/10.1145/1344471.1344525.

[231] D. Mardanbegi and D. W. Hansen, "Parallax error in the monocular head-mounted eye trackers", in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, ser. UbiComp '12, Pittsburgh, Pennsylvania: ACM, 2012, pp. 689–694, ISBN: 978-1-4503-1224-0. DOI: 10.1145/2370216.2370366. [Online]. Available: http://doi.acm.org/10.1145/2370216.2370366.

[232] M. Hutter and N. Brewer, "Matching 2-d ellipses to 3-d circles with application to vehicle pose identification", in *Image and Vision Computing New Zealand, 2009. IVCNZ'09. 24th International Conference*, IEEE, 2009, pp. 153–158.

[233] R. Jampel and D. Shi, "The primary position of the eyes, the resetting saccade, and the transverse visual head plane. head movements around the cervical joints.", *Investigative ophthalmology & visual science*, vol. 33, no. 8, pp. 2501–2510, 1992.

[234] J. Velez and J. D. Borah, *Visor and camera providing a parallax-free field-of-view image for a head-mounted eye movement measurement system*, US Patent 4,852,988, 1989. [Online]. Available: https://patents.google.com/patent/US4852988A.

[235] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Inferring human gaze from appearance via adaptive linear regression", in *2011 International Conference on Computer Vision*, IEEE, 2011, pp. 153–160.

[236] R. J. Leigh and D. S. Zee, *The neurology of eye movements*. Oxford University Press, 2015.

[237] E. Kasneci, G. Kasneci, T. Kübler, and W. Rosenstiel, "Online recognition of fixations, saccades, and smooth pursuits for automated analysis of traffic hazard perception", English, in *Artificial Neural Networks*, ser. Springer Series in Bio-/Neuroinformatics, P. Koprinkova-Hristova, V. Mladenov, and N. K. Kasabov, Eds., vol. 4, Springer International Publishing, 2015, pp. 411–434, ISBN: 978-3-319-09902-6. DOI: 10.1007/978-3-319-09903-3_20. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-09903-3_20.

[238] M. Vidal, A. Bulling, and H. Gellersen, "Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets", in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '13, Zurich, Switzerland: ACM, 2013, pp. 439–448, ISBN: 978-1-4503-1770-2. DOI: 10.1145/2493432.2493477. [Online]. Available: http://doi.acm.org/10.1145/2493432.2493477.

[239] L. Larsson, M. Nyström, R. Andersson, and M. Stridh, "Detection of fixations and smooth pursuit movements in high-speed eye-tracking data", *Biomedical Signal Processing and Control*, vol. 18, pp. 145–152, 2015.

[240] G. A. O'Driscoll and B. L. Callahan, "Smooth pursuit in schizophrenia: A meta-analytic review of research since 1993", *Brain and cognition*, vol. 68, no. 3, pp. 359–370, 2008.

[241] W. A. Fletcher and J. A. Sharpe, "Smooth pursuit dysfunction in alzheimer's disease", *Neurology*, vol. 38, no. 2, pp. 272–272, 1988.

[242] D Sauter, B. Martin, N Di Renzo, and C Vomscheid, "Analysis of eye tracking movements using innovations generated by a kalman filter", *Medical and biological Engineering and Computing*, vol. 29, no. 1, pp. 63–69, 1991.

[243] O. V. Komogortsev and J. I. Khan, "Kalman filtering in the design of eye-gaze-guided computer interfaces", in *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments*, Springer, 2007, pp. 679–689.

[244] O. V. Komogortsev and A. Karpov, "Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades", *Behavior research methods*, vol. 45, no. 1, pp. 203–215, 2013.

[245] D. J. Berg, S. E. Boehnke, R. A. Marino, D. P. Munoz, and L. Itti, "Free viewing of dynamic stimuli by humans and monkeys", *Journal of Vision*, vol. 9, no. 5, p. 19, 2009.

[246] J. S. A. Lopez, "Off-the-shelf gaze interaction", PhD thesis, IT University of Copenhagen, Denmark, 2009.

[247] L. Larsson, "Event detection in eye-tracking data", Master's thesis, Lund University, Sweden, 2010.

[248] E. Tafaj, G. Kasneci, W. Rosenstiel, and M. Bogdan, "Bayesian online clustering of eye movement data", in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA '12, Santa Barbara, California: ACM, 2012, pp. 285–288, ISBN: 978-1-4503-1221-9. DOI: 10.1145/2168556.2168617. [Online]. Available: http://doi.acm.org/10.1145/2168556.2168617.

[249] M. Vidal, A. Bulling, and H. Gellersen, "Detection of smooth pursuits using eye movement shape features", in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA '12, Santa Barbara, California: ACM, 2012, pp. 177–180, ISBN: 978-1-4503-1221-9. DOI: 10.1145/2168556.2168586. [Online]. Available: http://doi.acm.org/10.1145/2168556.2168586.

[250] R. Zemblys, "Eye-movement event detection meets machine learning", in *Proceedings of the 20th International Conference on Biomedical Engineering 2016*, vol. 20, 2016. [Online]. Available: http://biomed.ktu.lt/index.php/BME/article/view/3387.

[251] S. Hoppe and A. Bulling, "End-to-end eye movement detection using convolutional neural networks", *CoRR*, vol. abs/1609.02452, 2016. arXiv: 1609.02452. [Online]. Available: http://arxiv.org/abs/1609.02452.

[252] R. Zemblys, D. C. Niehorster, O. Komogortsev, and K. Holmqvist, "Using machine learning to detect events in eye-tracking data", *Behavior Research Methods*, vol. 50, no. 1, pp. 160–181, 2018, ISSN: 1554-3528. DOI: 10.3758/s13428-017-0860-3. [Online]. Available: https://doi.org/10.3758/s13428-017-0860-3.

[253] M. E. Bellet, J. Bellet, H. Nienborg, Z. M. Hafed, and P. Berens, "Human-level saccade detection performance using deep neural networks", *Journal of Neurophysiology*, vol. 121, no. 2, pp. 646–661, 2019, PMID: 30565968. DOI: 10.1152/jn.00601.2018. eprint: https://doi.org/10.1152/jn.00601.2018. [Online]. Available: https://doi.org/10.1152/jn.00601.2018.

[254] R. Zemblys, D. C. Niehorster, and K. Holmqvist, "Gazenet: End-to-end eye-movement event detection with deep neural networks", *Behavior research methods*, pp. 1–25, 2018.

[255] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols", in *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ser. ETRA '00, Palm Beach Gardens, Florida, USA: ACM, 2000, pp. 71–78, ISBN: 1-58113-280-8. DOI: 10.1145/355017.355028. [Online]. Available: http://doi.acm.org/10.1145/355017.355028.

[256] D. E. Knuth, "Two notes on notation", *American Mathematical Monthly*, pp. 403–422, 1992.

[257] M. Kleiner, D. Brainard, D. Pelli, A. Ingling, R. Murray, and C. Broussard, "What's new in psychtoolbox-3", *Perception*, vol. 36, no. 14, p. 1, 2007.

[258] D. M. Stampe, "Heuristic filtering and reliable calibration methods for video-based pupil-tracking systems", *Behavior Research Methods, Instruments, & Computers*, vol. 25, no. 2, pp. 137–142, 1993.

[259] O. V. Komogortsev, D. V. Gobert, S. Jayarathna, D. H. Koh, and S. M. Gowda, "Standardization of automated analyses of oculomotor fixation and saccadic behaviors", *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 11, pp. 2635–2645, 2010.

[260] O. C. Gyllensten, "Evaluating current algorithms for smooth pursuit detection on tobii eye trackers", Master's thesis, Royal Institute of Technology, Sweden, 2014.

[261] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes", *Pattern Recognition*, vol. 44, no. 8, pp. 1761–1776, 2011.

[262] M. Vidal, A. Bulling, and H. Gellersen, "Analysing eog signal features for the discrimination of eye movements with wearable devices", in *Proceedings of the 1st International Workshop on Pervasive Eye Tracking & Mobile Eye-based Interaction*, ser. PETMEI '11, Beijing, China: ACM, 2011, pp. 15–20, ISBN: 978-1-4503-0930-1. DOI: 10.1145/2029956.2029962. [Online]. Available: http://doi.acm.org/10.1145/2029956.2029962.

[263] L. Larsson, M. Nystrom, and M. Stridh, "Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit", *Biomedical Engineering, IEEE Transactions on*, vol. 60, no. 9, pp. 2484–2493, 2013.

[264] A. Ben-David, "A lot of randomness is hiding in accuracy", *Engineering Applications of Artificial Intelligence*, vol. 20, no. 7, pp. 875–885, 2007.

[265] Microsoft, *Eye control for windows 10*, Accessed in 2018-08-10. [Online]. Available: https://www.microsoft.com/en-us/garage/wall-of-fame/eye-control-windows-10/.

[266] Tobii Technology, *Tobii gaming*, Accessed in 2018-08-10. [Online]. Available: https://tobiigaming.com/.

[267] ——, *Tobii and qualcomm collaborate to bring eye tracking to mobile vr/ar headsets*, Accessed in 2018-08-10. [Online]. Available: https://www.tobii.com/group/news-media/press-releases/2018/3/tobii-and-qualcomm-collaborate-to-bring-eye-tracking-to-mobile-vrar-headsets/.

[268] M. Nyström, R. Andersson, D. C. Niehorster, and I. Hooge, "Searching for monocular microsaccades–a red hering of modern eye trackers?", *Vision research*, vol. 140, pp. 44–54, 2017.

[269] J. M. Juran, "The non-pareto principle; mea culpa", *Quality Progress*, vol. 8, no. 5, pp. 8–9, 1975.

[270] ——, *Pareto, Lorenz, Cournot, Vernoulli, Juran and others*. 1950, p. 25.

[271] Swartz Center for Computational Neuroscience, *Mobigaze*, Accessed in 2018-08-10. [Online]. Available: https://sccn.ucsd.edu/labinfo/equipment/eyetracker/.

[272] C. Topal, O. N. Gerek, and A. Doğan, "A head-mounted sensor-based eye tracking device: Eye touch system", in *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ser. ETRA '08, Savannah, Georgia: ACM, 2008, pp. 87–90, ISBN: 978-1-59593-982-1. DOI: 10.1145/1344471.1344494. [Online]. Available: http://doi.acm.org/10.1145/1344471.1344494.

[273] J. S. Agustin, M. Tall, D. W. Hansen, A. Voßkühler, H. Skovsgaard, and R. Newton, *Itu gaze tracker*, Accessed in 2018-08-10. [Online]. Available: https://sourceforge.net/projects/gazetrackinglib/.

[274] R. Mantiuk, M. Kowalik, A. Nowosielski, and B. Bazyluk, "Do-it-yourself eye tracker: Low-cost pupil-based eye tracker for computer graphics applications", in *International Conference on Multimedia Modeling*, Springer, 2012, pp. 115–125.

[275] O. Ferhat, F. Vilarino, and F. J. Sanchez, "A cheap portable eye-tracker solution for common setups", *Journal of eye movement research*, vol. 7, no. 3, 2014.

[276] E. S. Dalmaijer, S. Mathôt, and S. Van der Stigchel, "Pygaze: An open-source, cross-platform toolbox for minimal-effort programming of eyetracking experiments", *Behavior research methods*, vol. 46, no. 4, pp. 913–921, 2014.

[277] Diako Mardanbergi, *Haytham gaze tracker*, Accessed in 2018-08-10. [Online]. Available: https://sourceforge.net/projects/haytham/.

[278] E. Tribe, Accessed: 2018-08-10. [Online]. Available: https://theeyetribe.com.

[279] M. P. Kassner and W. R. Patera, "Pupil: constructing the space of visual attention", PhD thesis, Massachusetts Institute of Technology, 2012.

[280] COGAIN, *A catalogue of currently available eye trackers, categorized into systems for assistive technology, research purposes etc.* Accessed in 2018-08-10. [Online]. Available: https://wiki.cogain.org/index.php/Eye_Trackers.

[281] Randall Munroe (XKCD Comics), *Standards*, Accessed in 2018-08-10. [Online]. Available: https://xkcd.com/927/.

[282] Javier Barandiaran Martirena, *Videoman*, Accessed in 2018-08-10. [Online]. Available: http://videomanlib.sourceforge.net/index.php.

[283] L. Larsson, A. Schwaller, M. Nyström, and M. Stridh, "Head movement compensation and multi-modal event detection in eye-tracking data for unconstrained head movements", *Journal of neuroscience methods*, vol. 274, pp. 13–26, 2016.

[284] E. Kasneci, "Towards pervasive eye tracking", *IT-Information Technology*, vol. 59, no. 5, pp. 253–257, 2017.

[285] D. Wessel, E. Mayr, and K. Knipfer, "Re-viewing the museum visitor's view", in *Workshop Research Methods in Informal and Mobile Learning, Institute of Education, London, UK*, 2007, pp. 17–23.

[286] F. Walker, B. Bucker, N. C. Anderson, D. Schreij, and J. Theeuwes, "Looking at paintings in the vincent van gogh museum: Eye movement patterns of children and adults", *PloS one*, vol. 12, no. 6, e0178912, 2017.

[287] S Filippini Fantoni, K Jaebker, D Bauer, and K Stofer, "Capturing visitors' gazes. three eye tracking studies in museums", in *The annual conference of museums and the web*, 2013, pp. 17–20.

[288] B. W. Tatler, R. G. Macdonald, T. Hamling, and C. Richardson, "Looking at domestic textiles: An eye-tracking experiment analysing influences on viewing behaviour at owlpen manor", *Textile history*, vol. 47, no. 1, pp. 94–118, 2016.

[289] M. Mokatren, T. Kuflik, and I. Shimshoni, "Using eye-tracking for enhancing the museum visit experience", in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ser. AVI '16, Bari, Italy: ACM, 2016, pp. 330–331, ISBN: 978-1-4503-4131-8. DOI: 10.1145/2909132.2926060. [Online]. Available: http://doi.acm.org/10.1145/2909132.2926060.

[290] E. Mayr *et al.*, "In-sights into mobile learning an exploration of mobile eye tracking methodology for learning in museums", in *Proceedings of the Research Methods in Mobile and Informal Learning Wrokshop*, Citeseer, 2009.

[291] T. C. Kübler, C. Rothe, U. Schiefer, W. Rosenstiel, and E. Kasneci, "Subsmatch 2.0: Scanpath comparison and classification based on subsequence frequencies", *Behavior research methods*, vol. 49, no. 3, pp. 1048–1064, 2017.

[292] I. Morgan and K. Rose, "How genetic is school myopia?", *Progress in retinal and eye research*, vol. 24, no. 1, pp. 1–38, 2005.

[293] I. G. Morgan, A. N. French, R. S. Ashby, X. Guo, X. Ding, M. He, and K. A. Rose, "The epidemics of myopia: Aetiology and prevention", *Progress in retinal and eye research*, vol. 62, pp. 134–149, 2018.

[294] Amazon, *Hydration backpack*, Accessed in 2018-08-10. [Online]. Available: https://www.amazon.de/gp/product/B019IAB38A/ref=oh_aui_detailpage_o09_s02?ie=UTF8&psc=1.

[295] R. Rao and S. Vrudhula, "Performance optimal processor throttling under thermal constraints", in *Proceedings of the 2007 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, ser. CASES '07, Salzburg, Austria: ACM, 2007, pp. 257–266, ISBN: 978-1-59593-826-8. DOI: 10.1145/1289881.1289925. [Online]. Available: http://doi.acm.org/10.1145/1289881.1289925.

[296] Intel, Accessed in 2018-08-10. [Online]. Available: https://downloadcenter.intel.com/product/66427/Intel-Extreme-Tuning-Utility-Intel-XTU-.

[297] R. Likert, "A technique for the measurement of attitudes.", *Archives of psychology*, 1932.

[298] A. Poole and L. J. Ball, "Eye tracking in hci and usability research", *Encyclopedia of human computer interaction*, vol. 1, pp. 211–219, 2006.

[299] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf", in *Computer Vision (ICCV), 2011 IEEE international conference on*, IEEE, 2011, pp. 2564–2571.

[300] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features", in *European conference on computer vision*, Springer, 2010, pp. 778–792.

[301] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn", in *Computer Vision (ICCV), 2017 IEEE International Conference on*, IEEE, 2017, pp. 2980–2988.

[302] H. Jiang and E. Learned-Miller, "Face detection with the faster r-cnn", in *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, IEEE, 2017, pp. 650–657.

[303] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition", in *Proceedings of the British Machine Vision Conference 2015, BMVC 2015, Swansea, UK, September 7-10, 2015*, 2015, pp. 41.1–41.12. DOI: 10.5244/C.29.41. [Online]. Available: https://doi.org/10.5244/C.29.41.

[304] J. S. Shell, R. Vertegaal, and A. W. Skaburskis, "Eyepliances: Attention-seeking devices that respond to visual attention", in *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '03, Ft. Lauderdale, Florida, USA: ACM, 2003, pp. 770–771, ISBN: 1-58113-637-4. DOI: 10.1145/765891.765981. [Online]. Available: http://doi.acm.org/10.1145/765891.765981.