Towards wearable gaze interaction

Carlos Eduardo Leão Elmadjian

Thesis presented to the
Institute of Mathematics and Statistics
of the University of São Paulo
in partial fulfillment
of the requirements
for the degree of
Doctor of Science

Program: Computer Science
Advisor: Prof. Dr. Carlos Hitoshi Morimoto

The author was supported by FAPESP (grants 2015/26802-1 and 2017/06933-0) and by CNPq (grant 140447/2016-4)

São Paulo
May 16th, 2023

Towards wearable gaze interaction

Carlos Eduardo Leão Elmadjian

This version of the thesis includes the corrections and modifications suggested by the Examining Committee during the defense of the original version of the work, which took place on May 16th, 2023.

A copy of the original version is available at the Institute of Mathematics and Statistics of the University of São Paulo.

Examining Committee:

- Prof. Dr. Carlos Hitoshi Morimoto (advisor) – IME-USP [hitoshi@ime.usp.br]
- Prof. Dr. Roberto Hirata Jr. – IME-USP [hirata@ime.usp.br]
- Prof. Dr. Enkelejda Kasneci – Technische Universität München [enkelejda.kasneci@tum.de]
- Prof. Dr. Thies Pfeifer – Hochschule Emden/Leer [thies.pfeiffer@hs-emden-leer.de]
- Prof. Dr. Marco Porta – Università di Pavia [marco.porta@unipv.it]

I authorize the total or partial reproduction and publication of this work, by conventional or electronic means, for study and research purposes, provided that the source is cited.

Sisyphus is not, finally, a useful image. You don't roll some unitary boulder of language or justice uphill; you try with others to assist in cutting and laying many stones, designing a foundation.

— ***Adrienne Rich***, *Arts of the Possible*

Acknowledgments

This work is nowhere near what it was initially planned to be. But by no means is this said in a resentful tone. On the contrary: the unforeseen challenges made the journey incredibly more meaningful and rewarding. And as in most scientific endeavors, much of the hard work, blunders, and deadlocks are hidden from the reader. To those that were not spared of the struggling moments and stood strong by my side, my special thanks. I would also like to thank the São Paulo Research Foundation (FAPESP) for the financial support of the research projects covered in this thesis (grants 2015/26802-1 and 2017/06933-0), as well as the National Council for Scientific and Technological Development (CNPq) for their partial support (grant 140447/2016-4).

Resumo

Carlos Eduardo Leão Elmadjian. **Interação pelo olhar em computação vestível**. Tese (Doutorado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2023.

A interação humano-computador vem enfrentando uma mudança significativa do paradigma tradicional de interfaces WIMP utilizando PCs e notebooks para um ambiente mais distribuído e heterogêneo, onde a interação ocorre com múltiplos dispositivos, incluindo telefones celulares, relógios inteligentes, óculos inteligentes e dispositivos IoT. Como consequência, meios tradicionais de interação podem não fornecer uma usabilidade e experiência do usuário satisfatórias nesse cenário. Nesta tese, investigamos a interação baseada em olhar como uma alternativa para atender a esse novo paradigma e alguns dos seus desafios fundamentais. Em particular, investigamos novas técnicas para reconhecimento de padrões de movimento ocular, estimativa de olhar em 3D e métodos interativos baseados em olhar para cenários de microinteração. Embora nosso objetivo final seja aprimorar a interação por meio do olhar na computação vestível, dividimos esse objetivo em várias frentes, a saber: aprimorar a estimativa de olhar para ambientes em 3D usando dispositivos vestíveis de rastreamento do olhar; aprimorar o reconhecimento em tempo real de padrões de movimento ocular para suportar sistemas sensíveis ao contexto usando modelos convolucionais temporais profundos e métodos invariantes à escala; e criar novos métodos de interação por meio do olhar caracterizados pelo mecanismo de "olhar para selecionar", a despeito do problema do "toque de Midas". Nossos resultados demonstram que um procedimento de estimativa de olhar em 3D para computação vestível é possível, embora ainda muito desafiador; que nossas técnicas de reconhecimento de padrões de olhar são capazes de alcançar o melhor dos dois mundos: acurácia do estado da arte sendo computacionalmente leves; e que nossas técnicas interativas, como GazeBar e V-Switch, têm o potencial de melhorar as microinterações sem os mecanismos de segurança comuns implantados nos métodos baseados em olhar. Embora ainda existam obstáculos significativos a serem abordados em trabalhos futuros, incluindo estudos de usuário mais aprofundados e estratégias de detecção de contexto mais complexas, esta tese joga luz em direção à interação baseada em olhar e seus desafios relacionados neste ambiente de mudança de paradigma.

Palavras-chave: Computação vestível. Microinterações pelo olhar. Reconhecimento de padrões do olhar. Estimação do olhar em 3D.

Abstract

Carlos Eduardo Leão Elmadjian. **Towards wearable gaze interaction**. Thesis (Doctorate). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2023.

Human-computer interaction has been facing a significant shift from the traditional paradigm of WIMP interfaces using desktop PCs and notebooks to a more distributed and heterogeneous one, where interaction occurs with multiple devices, including mobile phones, smart watches, smart glasses, and IoT-enabled devices. As a result, traditional means of interaction may not provide a satisfactory user experience in such a scenario. In this thesis, we look into gaze-based interaction as an alternative to comply with this new paradigm and some of its fundamental challenges. In particular, we investigate new techniques to recognize eye movement patterns, 3D gaze estimation, and gaze-based interactive methods for micro-interaction scenarios. Though our ultimate goal is to enhance gaze interaction in wearable computing, we broke down this objective into several fronts, namely improving gaze estimation for 3D environments using wearable eye-tracking devices, enhancing real-time eye movement pattern recognition to support context-aware systems using deep temporal convolutional models and scale-invariant methods, and creating novel gaze interaction methods with a just-look-to-select mechanism, in spite of the Midas touch problem. Our results demonstrate that a 3D wearable gaze estimation procedure is feasible, though still very challenging; that our techniques for eye pattern recognition are able to achieve the best of both worlds: state-of-the-art accuracy while being computationally lightweight; and that our interactive techniques, such as GazeBar and V-Switch, have the potential to improve micro-interactions without the common safety mechanisms deployed to gaze-based methods. Though there are still significant obstacles to be addressed in future work, including thorough user studies and complex context detection strategies, this thesis shines new light towards gaze-based interaction and related challenges in this paradigm-shifting environment.

Keywords: Wearable computing. Gaze-based micro-interactions. Eye pattern recognition. 3D gaze estimation.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Back in 2002, the computer scientist Daniel P. Siewiorek argued that the merger of ubiquitous and wearable computing would be the next application design frontier (Siewiorek, 2002). At the time, it was believed that emerging technologies, such as eye tracking, would provide more appropriate means to interact and "fully exploit the advantages of wearable computers" (Calhoun and McMillan, 1998). Despite several known challenges involved in gaze-based interaction (Bulling and Gellersen, 2010; Jacob, 1991), there was a general perception that the eyes should play a relevant role in wearable computing (Tanriverdi and Jacob, 2000).

Because wearables are always on and available (Mann, 1997a), understanding user context might be what separates a useful from a disruptive application. As DeVaul pointed out, wearable interfaces should never assume to have the user's undivided attention (DeVaul, 2004). That is precisely why the eyes are such an exciting tool to address the issue of context awareness: firstly, because they are generally correlated with human attention span (Duchowski and Çöltekin, 2007); secondly, because eye movement patterns often provide cues about user activity or task engagement with no extra cognitive overhead (Ishiguro *et al.*, 2010; Kunze *et al.*, 2013).

Besides context awareness, gaze-based interaction also presents many other discernible advantages. Compared to traditional manual input, the eyes provide a means for hands-free, private, and unobtrusive interaction either with wearable or ubiquitous interfaces (Akkil *et al.*, 2016). Moreover, because the eyes can move effortlessly fast and interact even with distant objects, they presumably make the perfect choice for target acquisition in 3-dimensional environments (Tanriverdi and Jacob, 2000).

However, during the initial phases of this thesis project, more than a decade after Siewiorek's article, it was already evident that some of these promises were further from being delivered. Originally, our intent was to use eye data to create "cognition-aware user interfaces" (Bulling *et al.*, 2011), but despite clear advances, there were still many important issues pertaining to eye tracking hardware, gaze estimation techniques, pattern recognition, and efficient gaze interaction in the context of wearable computing. In face of such challenges, we realized that it was necessary first to bridge the gap between early expectations and technical support for gaze-based wearable interaction.

In this sense, this thesis aims to address these issues with contributions on at least three fronts: a) enhancing 3-dimensional gaze estimation techniques to handle the problem of interacting with objects at different depths; b) improving real-time eye movement patterns recognition to support better context-aware systems; c) proposing novel techniques for gaze-based interaction suited for wearable and micro-interactive scenarios.

In the following subsections, we provide a brief overview of state-of-the-art research on wearable gaze interaction, and we also describe the aims and objectives of our research.

1.1 Background

Research on wearable gaze interaction pervades an amalgam of various independent fields, ranging from computer vision techniques to track and estimate gaze on flat displays, going through machine learning and statistical methods to understand eye movement gestures or patterns, to finally human-computer interactive approaches designed to control applications.

What wearable technology adds to this mixture, however, is its high level of constraints — at least compared to traditional computing interfaces (Starner, 1996, 2001). Just to mention a few, wearable devices must be energy and heat efficient, at the same time that they are constantly sensing the user's environment. Not only that, the form factor also matters: they have to be light to wear, besides being unobtrusive and private for social acceptance. So one question that naturally arises is: do the same algorithms and techniques created for traditional computing paradigms can work seamlessly in the wearable context?

The bulk of eye tracking research has mostly presumed that eye data could be amassed in a controlled environment, with constant illumination, limited infrared exposure, and specialized hardware (Fuhl *et al.*, 2016). In wearable computing, this assumption is not necessarily true, and only more recently have we started to witness works concerned with eye tracking "in the wild" (Fuhl *et al.*, 2016; Hansen and Pece, 2005; Zhang *et al.*, 2015), generally with a trade-off between robustness to environmental conditions and tracking accuracy.

As for the gaze estimation algorithms, it is well known that almost every method consists in finding a mapping function between either eye features or a model to a point of regard (PoR) on a display, that is, a 2-dimensional surface (Hansen and Ji, 2010; Sesma-Sanchez and Hansen, 2018). This is particularly limiting because the user might be interacting with objects at different depths in a wearable scenario. And even if the user is performing a long interaction with a single object, user mobility could introduce, for example, parallax errors (Mardanbegi and Hansen, 2012) to the estimation function without an actual 3D estimation method.

Regarding the context-awareness problem, eye data-based studies typically resort to *post hoc* offline processing techniques for task comprehension or measure of engagement (Elmadjian *et al.*, 2022). Though these works provide invaluable findings that could be applied to better wearable gaze-based interaction, there is also the practical issue of being able to generate similar accurate results in real-time, so that we could have more

realistic wearable applications and adaptive systems.

On the human-computer interaction end, many authors argue that wearables should function as an extension of the mind (Clark and Chalmers, 1998; Mann, 1997b). As Siewiorek put it, "there is no Moore's Law for humans. (...) [They] have a finite and non-increasing capacity that limits the number of concurrent activities they can perform." (Siewiorek, 2002). Thus, since wearable computers are always available, they present an opportunity to enhance cognitive and executive user functions where human limitations become conspicuous, such as with environmental attention, working memory, or recall. However, how these devices can offer aid without interrupting or imposing additional cognitive burdens? Moreover, is it possible to have a concise and transparent interactive method in a wearable scenario, where the user could be constantly performing micro-interactions with intelligent devices presenting different interfaces?

There has been a significant effort to adapt known gaze-based interactive methods to mixed reality applications (Bektas, 2020; Kytö *et al.*, 2018; Orlosky *et al.*, 2014; Piumsomboon *et al.*, 2017). However, this increasing trend does not necessarily indicate that wearable context issues are being addressed. One important question is how to perform swift target acquisition, selection, and disengagement with multiple objects in the scene, each one with its own affordances and interfaces. An all-purpose gaze-based interactive method might be perceived as a sub-optimal solution in this scenario. Furthermore, traditional techniques that parade several safety mechanisms to avoid the Midas touch problem (Istance *et al.*, 2010; Kytö *et al.*, 2018; Majaranta and Räihä, 2002) might look like overkill for micro-interactive demands.

1.2 Aims and objectives

This thesis aims to be a step forward in supporting — and possibly fostering — wearable gaze-based interaction. Because we believe that this is, ultimately, a multifactorial problem, our main objective is comprised of several subgoals in more specific domains that we consider intertwined with our main objective.

Thus, as research directions, we propose to investigate: a) whether current all-purpose methods for gaze pattern recognition are robust and adapted to wearable technology, and if not, whether we can enhance the state of the art with more lightweight techniques without compromising accuracy significantly; b) in which conditions current gaze estimation methods can work in the wearable and ubiquitous context, where they fail, and how can we devise strategies to use gaze data in 3-dimensional environments; and c) whether state-of-the-art gaze interaction techniques are usable in the wearable scenario, and how can they be improved or should they be replaced by more context-sensitive designs.

1.3 Format

Because our contributions to improve the user experience in wearable gaze interaction are multidisciplinary, we have decided to structure the thesis into 3 distinct segments: eye pattern recognition, gaze estimation, and gaze interaction. This structure allows us to

focus the discussions on specific problems and challenges, and present our contributions as a collection of the publications that resulted in this thesis.

Therefore, the remaining of this thesis is divided into the following chapters: 2.Concepts covers the general theoretical key concepts; 3.Literature review presents an overview of the main works, challenges, and opportunities for the three sub-areas mentioned previously; 4.Eye pattern recognition, 5.Gaze estimation, and 6.Gaze interaction are the chapters comprising our publications; and 7.Conclusion summarizes all our contributions from a more broad perspective.

Chapter 2

Concepts

In this chapter, we present concepts about the human eye and its basic movements, techniques to track eye movements, and how to estimate the 2D gaze point or the 3D eye gaze direction. We also introduce some concepts about how eye gaze can be used for human-computer interaction and some machine-learning concepts that might help the reader to understand our work on eye movement classification.

2.1 The human eye

The eye is a complex organ comprised of several structures working together to enable vision. The *retina*, located at the back of the eye, contains photosensitive neurons, known as *rods* and *cones*, that transmit visual signals to the brain. For the light to reach the retina, it must travel through the *cornea*, then through an opening called *pupil*, then the eye *lens*, and finally the *vitreous humor*, as shown in Figure 2.1. The pupil, controlled by the *iris*, regulates the amount of light that enters the eye, while the lens refracts the light, focusing it onto the retina. Another large and visible anatomic area is the *sclera*, which protects and maintains the shape of the eyeball.

Humans have more than 200-degree horizontal arc of visual field and around 150 degrees of vertical range, but for depth perception, which relies on binocular vision, our field-of-view drops to 114 degrees (horizontally) (Howard and Rogers, 1995). Yet, we can only perceive high visual acuity and color density in a tiny portion of our visual field — a little more than 1 degree. This is due to the *fovea*, a region of the retina with 1.2 mm in diameter and a high density of *cone* receptors, which are related to color and acuity vision (Purves, 2004).

The light ray that passes through the pupil center, i.e., the pupillary axis, is approximately aligned with the *optical axis* of the eye. However, this must not be confused with the *visual axis*, which is formed by connecting the fovea and the nodal point of the eye. It is known that the optical and visual axis deviates from each other (Guestrin and Eizenman, 2006), and this difference is commonly known as the *kappa* angle, but some authors also decompose it into *alpha* and *beta* components (see Figure 2.2). Since only the optical axis is observable through video-based tracking and since this angular offset varies between

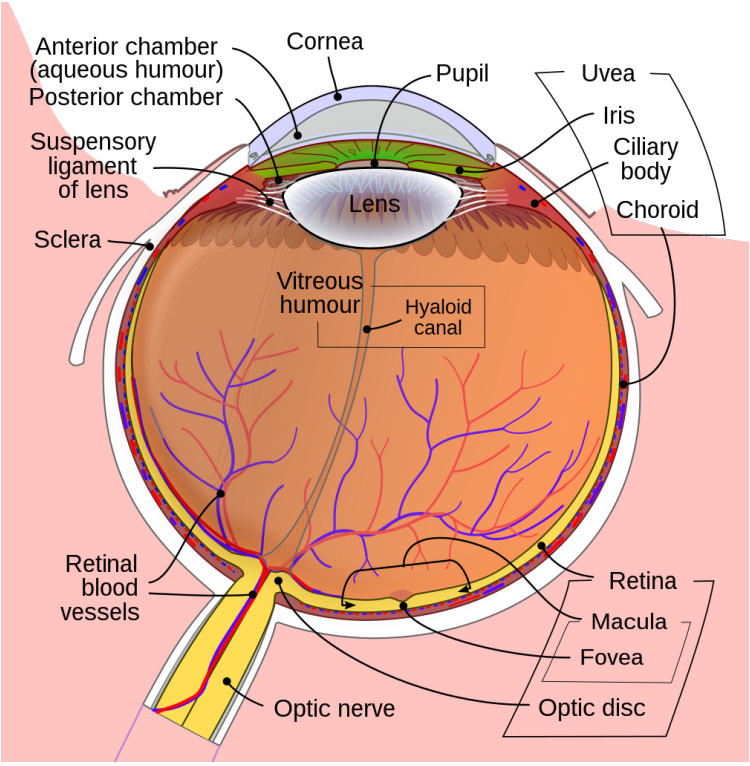

Figure 2.1: *Most relevant structures of the eye. (source: commons.wikimedia.org, authors: Rhcastilhos and Jmarchn, license: CC BY 3.0)*

subjects, the visual axis must be inferred through a series of observations comprising different optical axes and fixation targets.

Regarding eye movements, unlike other complex parts of the human body, they can be performed using only three pairs of extraocular muscles. The following is a list of the main eye movements:

- **saccades:** these are rapid and ballistic movements that suddenly change points of fixation. They are said to be ballistic because a saccade cannot be altered once in its course. Saccades can be voluntary, but most of the time are elicited unconsciously (Purves, 2004). It is believed that during a saccade no visual information processing is performed, a phenomenon known as *saccadic suppression* (Matin, 1974).
- **fixations:** they occur when the eyes remain fixed in a single point of the scene, hence the name. In actuality, the eyes never rest, and even during fixations drifts and micro-saccades are expected, which is an involuntary mechanism to keep the photoreceptor cells in the retina stimulated (Purves, 2004). The length of fixations is a research topic in itself, as it can be related to the acquisition of visual information and the observer's cognitive state (Rayner, 1998).
- **vergence:** these movements have the objective of aligning the fovea of each eye with targets at different distances. As a result, the eyes may converge or diverge to see something near or far away. Vergence is usually accompanied by the accommodation of lens and pupillary constriction or dilation to change the depth of field (Purves, 2004).

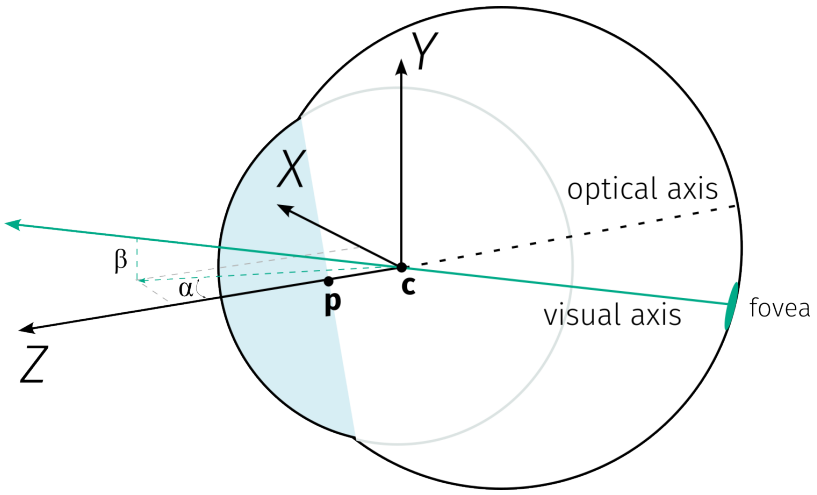

Figure 2.2: *Offset between the optical and visual axes with alpha and beta components of the kappa offset between the axes.*

- **smooth pursuits:** these are slow tracking movements designed to keep a moving target on the fovea. Though people are conscious of a pursuit movement, it is uncommon to perform it in the absence of a moving target as a stimulus (Purves, 2004).

- **vestibulo-ocular reflex:** they are a set of various different movements designed to stabilize the eyes relative to a target, i.e., a compensating mechanism to keep a point of regard in the fovea.

By far, saccades and fixations have been the most employed basic eye movements in the design of gaze-based interaction techniques and human-computer interfaces, though more recently smooth pursuits started to be explored as well. Figure 2.3 illustrates the typical "scan path" patterns seen for these three classes of eye movements.

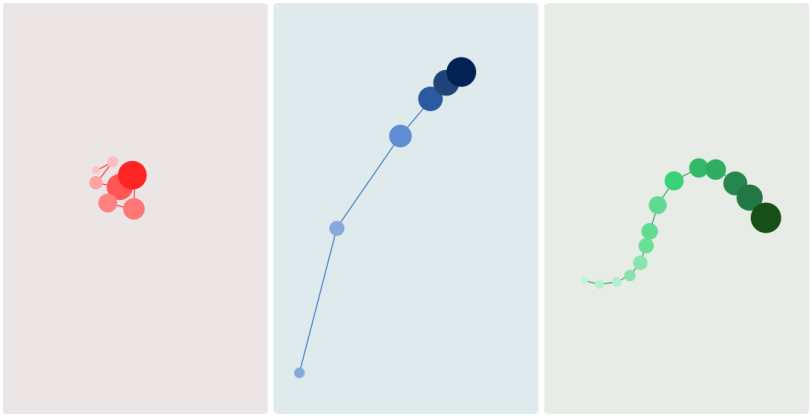

Figure 2.3: *Sample scan paths of a fixation (left), a saccade (middle), and a smooth pursuit (right). Lighter and smaller circles indicate older sampling points of each pattern.*

2.2 Eye tracking

Eye tracking is a broad terminology that has been used either to refer to the ability to track an individual's eye movements or to the technology of detecting what a person is looking at (Duchowski, 2017). More strictly, we will refer to the latter as the *gaze estimation* problem, limiting the concept of eye tracking simply as the combined set of algorithms and devices that allows a machine to capture eye displacements with time, without necessarily any assumption about gaze direction.

There are several known techniques, such as scleral search coil, infrared oculography (IOG), electrooculography (EOG), and video oculography (VOG), which is the most popular and accessible technique. We address all of them in more detail below:

- **scleral search coil:** this technique provides excellent spatial and temporal resolution. It consists of applying a coil of wire to the eye, usually on a locally anesthetized eye for only a limited amount of time. Since it is considered very invasive compared to other methods, it has been used more often with animal subjects rather than humans (Murphy *et al.*, 2001).
- **electrooculography:** in EOG, electrodes are attached to the skin around the eyes to measure the electric field produced when the eyes start to rotate. EOG is not as accurate as other methods, usually giving noisy estimates of vertical and horizontal eye movements only. However, the major advantage of EOG is to provide eye movement readings even when the eyelids are closed, which is difficult – if not impossible – to achieve with the other known methods (Barea *et al.*, 2002).
- **infrared oculography:** also known as infrared reflectance oculography, IOG relies on measuring the intensity of reflected infrared (IR) light sources directed to the eye. As the eye surface moves, it is possible to capture the differential reflexes of IR sources to estimate position changes. Since IR is not in the human visible spectrum, it does not require a bright environment to work, though it can be negatively impacted by sunlight, since it is a conflicting IR source. IOG techniques, such as corneal reflection, have been widely incorporated into most commercial video-based tracking devices as a means to increase accuracy (Chennamma and Yuan, 2013).
- **video-based oculography:** VOG makes use of one or more cameras to track eye movements, and it is the most popular method seen in commercial eye trackers. With advancements in video processing and computer power, VOG also became relatively more affordable. Video-based tracking can be done either with visible, IR light or a mixture of both. Most accurate commercial eye trackers rely primarily on IR light, as it facilitates the capturing of features such as pupil contours, that are typically used in gaze estimation algorithms.

In our research, we resort strictly to video-based tracking. The typical eye-tracking pipeline, in this case, is portrayed in Figure 2.4. As can be seen, it generally starts with capturing raw frames of the eyes, either through specialized hardware with infrared sensors or simply using commodity cameras. Then, computer vision algorithms are applied to each frame to extract features (e.g., edges, low-brightness areas, ellipses, etc.) that can be fed into a model capable of estimating the pupil center in the image based on this input.

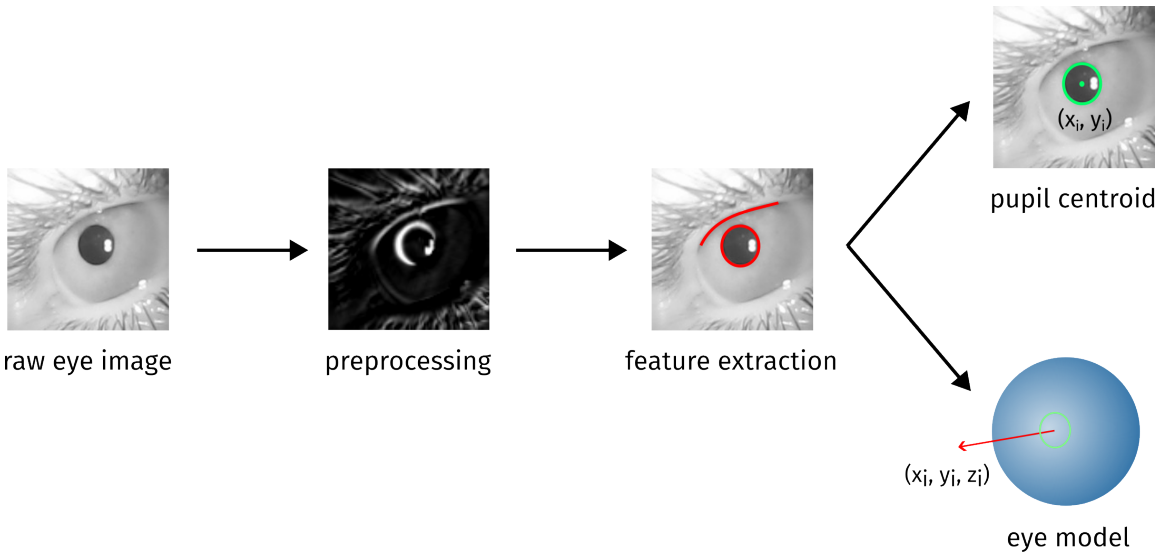

Figure 2.4: *One of the most common eye-tracking pipelines: each frame of the eye is preprocessed to allow for feature extraction, which in turn will feed either an eye model or an algorithm for pupil centroid estimation.*

Though this is common ground to most techniques, some methods might take additional steps to not only estimate the pupil position but also create a 3D virtual model of the eye. Other implicit modeling approaches, such as convolutional neural networks, might even take only the raw eye image and create latent features in hidden layers, mapping them directly to pupil location or even gaze direction estimates (Krafka *et al.*, 2016), which is commonly regarded as a "black-box" tracking approach.

2.3 Gaze estimation

Video-based gaze estimation consists in finding a model that maps eye images to either gaze direction (i.e., a 3D estimation) or a point in a surface (2D). This mapping usually assumes an eye-tracking preprocessing step, to extract useful features that can be fed into this model. To date, it is still considered a difficult problem due to the very nature of human vision physiology, as well as the limited sensing capabilities of camera systems.

One first challenge is the so-called *kappa* angle. The visual axis is hardly aligned with the pupil center and cannot be directly measured with video-based tracking alone. Therefore, estimation methods generally need several calibration points to infer the kappa angle, either explicitly or implicitly, to improve accuracy.

Another difficulty lies with how the human brain works. The eye muscles and crystalline lens work in conjunction to make a target object in space look as crisp as possible on the fovea. But even if we could let an individual fixate on this object with the head completely static, a certain degree of eye movement would be observed nevertheless. One of the theories for this phenomenon purposes that cone receptors in the retina get desensitized to constant stimuli (Purves, 2004). Regardless of the explanation, this human response makes it harder to find a precise relationship between a target and the eye position.

There are several classes of estimation methods, and the literature (Hansen and Ji, 2010) usually separates them into the following ones:

- **geometric-based models:** also known as model-based estimation, with this kind of approach, usually eye features are extracted to build a model with geometric attributes. This method has the advantage of being perceptually more stable since the estimation function is constrained by the geometric properties of the eye model, so a set of input features has to be necessarily mapped to a viable configuration, which in practice is more robust to outliers in the tracking signal. As a downside, a poorly-constructed eye model could be persistently inaccurate in comparison.
- **appearance-based models:** this class of methods typically involves finding a statistical relationship between eye features and a target. These methods are also called *probabilistic* because they are not really concerned with eye movement mechanics, modeling refraction, or corneal reflection. They tend to be very sensitive to noise in the eye-tracking signal, though there are several techniques to mitigate this issue. With the emergence of end-to-end models, this family of gaze estimation techniques is usually divided into two subcategories (Cazzato *et al.*, 2020):
  - **feature-based regression:** feature-based techniques generally try to regress eye features, such as the coordinates of projected pupil centroids in the camera space to a different coordinate system. The feature extraction process is mostly rule-based and assumes some domain knowledge to select characteristics that are more determinant to estimating gaze direction or PoR on a surface.
  - **end-to-end regression:** in this category, there is no explicit feature extraction in the estimation pipeline. These methods simply take as an input the raw image of the eyes — or sometimes the whole human face — and try to find the related gaze estimation function. The growing use of artificial neural networks helped end-to-end models become more popular. Yet, though the gap has been closing in the last few years, they tend to be a comparatively less accurate class of gaze estimation techniques (generally because they strive to be also "user-agnostic").

2.4 Gaze interaction

Gaze interaction is a prolific area of research and it is beyond the scope of this section to cover all known techniques. Rather, our intent here is to provide a more general and taxonomic view of the field, highlighting, whenever appropriate, the important concepts that have been pervading it for decades.

We say that any gaze-based form of interaction involves necessarily some clear eye pattern that can be recognized by a machine, either by algorithmic methods or statistical ones. We define the set of methods comprised of deliberated eye movement patterns made by the user to trigger a particular machine response as *active gaze* techniques, whereas applications that change their state based on spontaneous or unconscious eye movements are said to be exploiting *passive gaze* use (Duchowski, 2018), as usually seen in adaptive systems.

Gaze-based methods are commonly categorized according to how the selection task is performed. In a more broad sense, if selections are made with the use of gaze in conjunction with other means, such as hand input or voice, we say that these are *multimodal gaze* interactive methods. If a target is selected purely through certain eye movement patterns, then we say these are *gaze-only* interactive methods.

Gaze-only techniques are generally plagued by the so-called *Midas touch* problem (Jacob, 1991), which is the conflict arising from using the eyes to either gather information from an interface as well as to trigger a change on it (i.e., performing a selection by just looking at a target). Because video-based eye tracking alone cannot sense user intent, gaze-only techniques frequently have to resort to "safety" mechanisms to avoid provoking unwanted selections. There are several known strategies, such as *dwell-time* selection (Majaranta and Räihä, 2002), or *saccadic gestures* (Isokoski, 2000), and they are covered in Chapter 3.

2.5 Machine learning techniques

Machine learning (ML) techniques typically rely on data to make predictions. One striking difference to other statistical learning methods though is that ML algorithms are able to learn rules without explicit programming. In this section, we describe the major ML techniques that are associated with this thesis.

2.5.1 Support Vector Machines

Support Vector Machines, or simply SVMs, are supervised learning models that can be used either for classification or regression problems. SVMs achieved high praise and popularity during the mid-1990s and early 2000s, due to their ability to find the hyperplane or set of hyperplanes with the largest linear separation between classes in a high-dimensional space (Hearst *et al.*, 1998). Despite the original algorithm being developed for linearly separable problems, SVMs can be used with non-perfectly separable classes with the addition of soft margins (which tolerate one or more points crossing the boundary hyperplane) and can also be applied to nonlinear problems as well, using the so-called kernel trick (which transforms the input space into a higher-dimensional one).

The most elementary form of a linear SVM classifier is formally defined as follows: given a dataset of $\boldsymbol{x}$ vectors and associated y targets, we want to determine *maximum-margin* hyperplane that separates $\boldsymbol{x_i}$, for which $y_i = 1$, from $\boldsymbol{x_j}$, for which $y_j = -1$. Assuming such a hyperplane exists, then there is a vector $\boldsymbol{w}$ orthogonal to it, such that:

$$\boldsymbol{w}^T\boldsymbol{x} - b = 0$$

Since $\boldsymbol{w}$ is normal to the hyperplane and assuming the problem is linearly separable, then there must be two other parallel hyperplanes with a distance of $\frac{2}{||\boldsymbol{w}||}$ between them, such that:

$$\boldsymbol{w}^T\boldsymbol{x} - b = 1$$

$$\boldsymbol{w}^T\boldsymbol{x} - b = -1$$

Thus for each i, we can say:

$$\boldsymbol{w}^T\boldsymbol{x} - b \geq 1, \text{if } y_i = 1$$
$$\boldsymbol{w}^T\boldsymbol{x} - b \leq -1, \text{if } y_i = -1$$

Therefore, this can be solved as an optimization problem defined as follows:

$$\text{minimize } ||\boldsymbol{w}|| \text{ subject to } y_i(\boldsymbol{w}^T\boldsymbol{x_i} - b) \geq 1, \forall i \in \{1, ..., n\}$$

2.5.2 Gaussian Processes

A Gaussian Process (GP) is a stochastic process in which every linear combination of its random variables is assumed to be normally distributed. Though this might be just an approximation in most cases, in practice classifiers and regressors built on top of Gaussian Processes are known to perform really well with real-world problems. Just like SVMs, GPs are a type of kernel method, but they can typically provide highly calibrated probabilities.

GPs are primarily governed by their covariance function (i.e., the kernel), as it encodes the similarity between data input and targets. More formally, if $y = f(\boldsymbol{x}) + \epsilon$, where $f = \boldsymbol{w}^T\boldsymbol{x} - b$, then:

$$f \sim GP(\mu, k)$$

where $\mu(\boldsymbol{x})$ is the mean, and $k(\boldsymbol{x}, \boldsymbol{x'})$ is the covariance function. Thus, to model the predictive distribution $P(f_*|\boldsymbol{x_x}, D)$, where f_* denotes f on testing input and D the training dataset, we can use Bayes' rule to define our prior as:

$$P(f|\boldsymbol{x}) \sim \mathcal{N}(\mu, \Sigma)$$

where Σ is the covariance matrix, which can be decomposed into several kernel matrices. In fact, because GPs perform several matrix operations, compared to SVMs, they tend to be more resource-intensive and lack scalability as the dataset grows in size. However, contrary to SVMs, GPs have the advantage of being Bayesian, since they construct a prior distribution, and thus the model can be updated in an online fashion as more data become available, while SVMs have to be re-trained entirely.

2.5.3 Artificial Neural Networks

Artificial Neural Networks (or ANNs) are computing systems that rely on a connectionist approach and a set of activation functions to learn by example. This is achievable mainly due to the so-called *backpropagation algorithm*, which allows the network to update the weight of its units to minimize the global error with respect to a set of examples. This

allows them to learn without specific programming rules, which is often referred to as implicit modeling or just as a "black-box" model. Because ANNs is a very broad topic, we will discuss in detail only the main architectures investigated in this thesis.

LSTMs

Long Short-Term Memory networks are a type of recurrent neural network (RNN), meaning that, unlike conventional feed-forward architectures, they present feedback connections. This feature is generally regarded as the key component that makes recurrent nets well-suited to sequence learning. In practice though, vanilla RNNs cannot keep track of long-term dependencies in an input sequence due to the "vanishing gradient" problem (Hochreiter and Schmidhuber, 1997) with the backpropagation algorithm. LSTMs were developed to tackle this particular issue since each LSTM unit allows the gradient to back-propagate unaffected.

A typical LSTM unit is comprised of cells with an input gate, an output gate, and a forget gate, as depicted in Figure 2.5. The relationship between the gates regulates the information flow between the sequence of cells in the network, though the key component of an LSTM module is the cell state, which functions like a conveyor belt that lets information pass through the chain virtually unchanged if desired.

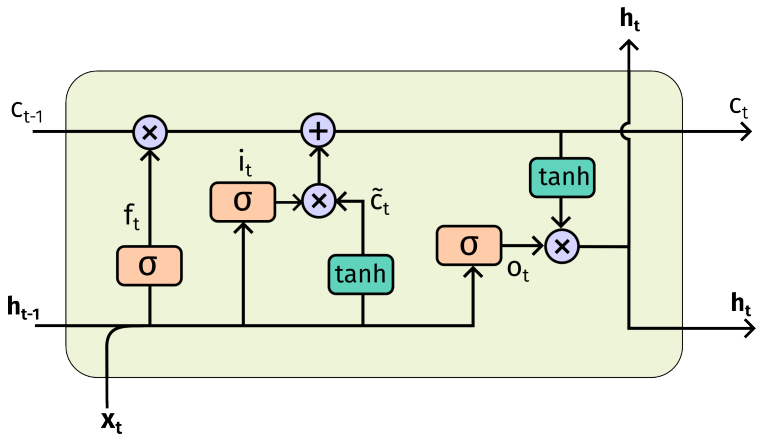

Figure 2.5: *LSTM cell and its major components: the input (x_t), the forget, input, and output gates (f_t, i_t, and o_t), the cell state and its update (c_t and $\tilde{c}_t$), and the output (h_t)*

Despite their popularity and success in several sequence-based problems, LSTMs are generally regarded as hard to train, and because of a chain-like structure, they cannot be easily parallelized as some other networks, making them comparatively slow to train.

GRUs

Often considered as a variation of LSTMs, Gated Recurrent Units (GRUs) is a type of network introduced by Cho et al. (Cho *et al.*, 2014) also as a solution to the vanishing gradient problem. One advantage of GRUs, though, is that they are comparatively simpler than LSTMs, in terms of architectural structure, typically requiring fewer parameters to train than LSTMs due to the lack of an output gate (see Figure 2.6). Because of that, GRUs generally outperform LSTMs in terms of convergence time and parameter updates.

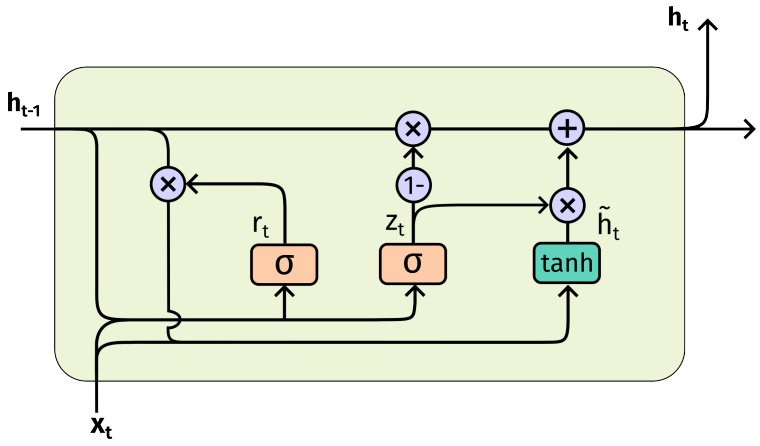

Figure 2.6: *GRU cell and its major components: the input (x_t), the relevance and update gates (r_t and z_t), and the updated cell state value ($\widetilde{h}_t$).*

Just as with LSTMs, a GRU network is typically comprised of cells (or units) with several gates and a cell state, though it is common to witness some variants of fully gated units in the literature. Also, despite being newer and presenting some practical advantages, GRU's performance as a model has been generally found to be comparable to LSTM (Ravanelli *et al.*, 2018).

CNNs

A Convolutional Neural Network (CNN or ConvNet) is a type of ANN with a shared-weight architecture of convolutional kernels (also known as "filters") that slide through the input to provide a feature-map response. Because of this, CNNs are, in theory, space-invariant with respect to the input. In practice, however, most CNNs are not really space-invariant due to the *pooling* steps that comprise most of these architectures.

CNNs are generally built with a series of convolutional layers interpolated by pooling layers, with a fully-connected layer at the end. The goal of a convolution operation is to extract features implicitly from the input, while a pooling layer is responsible for dimensionality reduction, passing forward dominant features while decreasing the computational power required in the process. Finally, the fully-connected layer is a straightforward way of learning a non-linear combination of high-level features present at the end of the chain.

Along with LSTMs, CNNs are arguably one of the most successful architectures in the deep learning domain, with state-of-the-art applications in several fields, though they have been more extensively used with image-related problems.

TCNs

A Temporal Convolutional Network (TCN) can be understood as a 1D fully-convolutional network with causal convolutions. This means that given a time instant t in a time series and a constraint y_t, then y_t can only be satisfied by $x_0, ..., x_t$, and not by $x_{t+1}, ..., x_{t+n}$. Another key feature of TCNs is their ability to map a sequence of any length to an output with the same length, similar to recurrent networks.

Despite having many aspects in common with CNNs, TCNs were developed particularly for sequence modeling. Compared to LSTMs or GRUs, though, TCNs present some important advantages. TCNs are known to have a longer memory than recurrent nets with the same capacity since they do not have gating mechanisms. They also have a flexible receptive field size, lower memory requirements for training, and can be trained with variable input length. Also, because TCNs have a convolutional structure, they can be more easily parallelized, meaning less training time than recurrent nets, which present a more sequential pipeline structure (Bai *et al.*, 2018).

Chapter 3

Literature review

As part of a multifaceted effort to address the challenges of gaze interaction in the wearable context, this chapter presents a literature review of three specific fronts: eye pattern recognition, gaze estimation, and wearable gaze interaction techniques. By exploring the challenges and opportunities associated with each of these fronts, we aim to provide a solid foundation for the research that follows. Our goal is to contribute to the advancement of gaze interaction technology and enable new and more intuitive ways for users to interact with their wearable devices.

3.1 Eye pattern recognition

3.1.1 Overview of basic eye movement recognition

Recognition of eye movement patterns is demanded in several domains, as they may provide meaningful information about human behavior. On a higher level, these patterns may give us insights about the user's cognitive state (Burch *et al.*, 2019; Sanches *et al.*, 2017) or degree of attention (Maruyama *et al.*, 2016; Wang and Hung, 2019), and on a more practical level, they can support applications, such as biometrics (Abdrabou *et al.*, 2021; Bayat and Pomplun, 2017; George and Routray, 2016), text input (Feng *et al.*, 2021; MacKenzie and Zhang, 2008; Tula and Morimoto, 2016a), accessibility (Koochaki and Najafizadeh, 2018), adaptive systems (de Greef *et al.*, 2009; Edwards, 1998), among many others.

Basic eye movement classification in the literature is generally concerned with fixations, saccades, and smooth pursuits (see Figure 2.3). In fact, the simultaneous task of identifying these three patterns is often called the *ternary eye movement classification problem* (3EMCP) (Berndt *et al.*, 2019).

Early classifiers were mostly based on threshold algorithms. One such example is the *velocity threshold* criterion (Salvucci and Goldberg, 2000) that separates saccades from other eye movements based on their characteristic high velocity that can be computed from the eye data stream. Adaptive velocity thresholds, however, tend to be more robust to noisy gaze data than static methods (Nyström and Holmqvist, 2010), being their natu-

ral replacement. Because such algorithms are simple and lightweight on computational resources, they are still widely employed, despite not being as accurate as more modern techniques.

Statistical approaches, that leverage pattern distribution knowledge to make inferences, are also reasonably common. Berg et al. (Berg *et al.*, 2009), for example, demonstrated that it is possible to use Principal Component Analysis (PCA) to establish more generic dispersion thresholds that are able to discriminate fixations, saccades, and smooth pursuits, though this approach is very parameter-sensitive. Komogortsev and Khan (Komogortsev and Khan, 2007), by their turn, created the Attention Focus Kalman Filter (AFKF), an extension of a previous method (Sauter *et al.*, 1991), in which a Kalman filter with a χ^2-test is used to detect saccades. Still in the realm of basic eye movement patterns, Santini et al. (Santini *et al.*, 2016) presented a more reliable solution called the Bayesian Decision Theory Identification (I-BDT) algorithm. This method consisted in calculating the posterior for each event, given eye velocity and movement ratio over windows according to Bayes' rule.

With the growth and popularity of novel machine learning techniques, strictly data-driven methods also started to be explored. Vidal et al. (Vidal *et al.*, 2012), for instance, considered a set of shape features to recognize smooth pursuits. Another example is the work of Zemblys et al. (Zemblys *et al.*, 2018), in which they introduced a random forests-based model to classify fixations, saccades, and post-saccadic oscillations (which can only be observed with high-frequency eye trackers).

Among all machine learning-based approaches, artificial neural models are the most accurate models known to date. Hoppe and Bulling (Hoppe and Bulling, 2016) proposed one of the earliest ones, a CNN-based end-to-end architecture to classify fixations, saccades, and smooth pursuits in a continuous gaze stream, from which frequency features are extracted to train their model. Later, Starsev et al. (Startsev *et al.*, 2019b) introduced a 1D-CNN-BiLSTM *seq2seq* network for a similar task. They showed that their 1D-CNN-BiLSTM classifier outperforms 12 previous baseline models with the GazeCom dataset ((Agtzidis *et al.*, 2016; Berg *et al.*, 2009; Dorr *et al.*, 2010; Komogortsev and Karpov, 2013; Komogortsev and Khan, 2009; Larsson *et al.*, 2015)) in offline classification.

Yet, despite the upper hand of sequence-based neural models, they do present some drawbacks, such as requiring a large amount of labeled data for training and a high computational footprint. This last aspect, in particular, can turn such networks less fit for real-time classification, since they would demand more processing time for online predictions, and in the case of bi-directional architectures that would be simply unfeasible.

As for other more complex gaze patterns, there have been important developments in several different areas, ranging from security (Carneiro *et al.*, 2019) to disease diagnosis (Molitor *et al.*, 2015). Holland and Komogortsev, for example, investigated how eye movement patterns could be utilized for biometrics (Holland and Komogortsev, 2013). Sattar et al. tried to predict potential targets during a visual search by processing a set of fixations using an SVM model (Sattar *et al.*, 2015). Kunze et al. proposed an ingenious way of detecting what kind of document a user is reading by processing fixations and saccades within a certain window span of minutes (Kunze *et al.*, 2013), while reading detection itself has been a common topic in the literature, with various algorithms available (Biedert *et al.*, 2012; Campbell and Maglio, 2001b).

Though there are many other applications involving complex gaze patterns, it is important to stress that virtually all these techniques are built on top of more basic patterns, such as saccades, fixations, and smooth pursuits. Thus, this thesis focuses on improving the recognition of elementary events to better support such techniques.

3.1.2 Online eye movement recognition in wearable computing

Because wearable devices are always available, eye movement recognition algorithms must be not only low on computer resources (and avoid battery drain), but also fast enough to make online predictions. When examining the literature on eye movement pattern recognition for ubiquitous or wearable computing, authors frequently neglect their model suitability for real-time scenarios, i.e., many proofs of concept assume some sort of post-hoc processing time.

For example, in the works from Kunze et al. and Sattar et al. mentioned earlier, it is assumed an offline processing step to identify accurately these patterns. Reading detection algorithms also often assume a high refreshing rate (i.e., at least 60 Hz) (Campbell and Maglio, 2001b), which may be prohibitive for wearable devices.

Simple and more lightweight classifiers with reasonable accuracy, such as the I-BDT (Santini *et al.*, 2016), seem fitter for the wearable context, though not without setbacks, such as user calibration and frequent parameter adjustment. Much more accurate, robust, and general models, namely the 1D-CNN-BiLSTM from Startsev et al., besides being heavier on the processing side, do require taking future information into account to perform classification (i.e., using data from time step $t + i$ to predict pattern at t).

Another approach to eye pattern recognition in the wearable context involves limiting or simplifying the desired pattern search space to enhance accuracy at the cost of less general applicability. For instance, Esteves et al. proposed the use of motion correlation between the eyes and a moving target on a smartwatch to perform selections (Esteves *et al.*, 2015; Velloso *et al.*, 2017), something that was later explored in other different wearable and ubiquitous scenarios (Carneiro *et al.*, 2019).

In this thesis, we strive to achieve the best of both worlds: the convenience and lightness of online models with the accuracy and generalization of offline data-driven methods. In the case of general basic movement understanding, we show that this is possible by leveraging sequential networks with a highly parallel structure, such as Temporal Convolutional Networks (TCNs) (Bai *et al.*, 2018), while pre-processing gaze data stream in a very efficient way. In the case of complex patterns, such as reading, we show that this is possible using differential analysis.

3.2 Gaze estimation

3.2.1 General gaze estimation methods

Although there are known gaze estimation procedures from over 100 years ago, mature and reliable computer vision-based methods are only a few decades old (Cazzato *et al.*,

2020). When we talk about the ability to obtain the user's point of regard or gaze direction in a continuous fashion, then it is generally acknowledged that there are only two main classes of gaze estimation techniques: geometric-based and appearance-based methods (Hansen and Ji, 2010), with the last class typically subdivided into either feature-based methods or end-to-end systems (Cazzato *et al.*, 2020).

In general, geometric models are known to be robust to eye tracker slippage and are also able to compensate for the parallax error (Mardanbegi and Hansen, 2012), making them a suitable choice for head-mounted eye trackers. Also, since geometric models provide gaze vectors through a rigid transformation of the eyeballs, they require in theory just one screen target to perform a user calibration, though in practice more points are necessary to improve accuracy. In fact, many of the accuracy issues found in these models can be attributed to simplifying assumptions, such as that the eyeball is a perfect sphere, ignoring the lens refraction, or assuming default human-related parameters.

One of the earliest geometric models was proposed by Shih et al. (Shih *et al.*, 2000), though Guestrin and Eizenman are generally regarded as the first to formalize and generalize this kind of model to a variable number of light sources and cameras (Guestrin and Eizenman, 2006). Many other works followed a similar approach afterwards (Hennessey *et al.*, 2006; Villanueva *et al.*, 2006), but requiring a fully calibrated setup (i.e., knowing all necessary camera parameters).

More general geometric models that do not need a fully calibrated setup appeared with the work of Yoo and Chung (Yoo and Chung, 2005), by exploiting the cross-ratio of four light sources in projective space. This work was later extended by Coutinho and Morimoto, simplifying the number of parameters needed for on-screen estimation (Coutinho and Morimoto, 2006).

If there are multiple cameras in the tracking setup, then 3D eye modeling and head movement compensation (including rotation) becomes an easier task due to stereo vision (Beymer and Flickner, 2003; Brolly and Mulligan, 2004). Still, estimating the 3D point of gaze, until recently, has only a few results available in very constrained scenarios (McMurrough *et al.*, 2012; Munn and Pelz, 2008; Takemura *et al.*, 2010).

As for appearance-based models, they basically rely on statistical methods to generate estimates. Feature-based techniques, in particular, depend on tracking specific elements of the eye image, such as the projected pupil center, whereas end-to-end systems typically provide estimates based only on raw frames of a user's face or the eyes.

In the case of feature-based estimation, since the eyeball rotation occurs in 3D space and the features are captured through their projection on the camera plane, displacement of eye attributes will not be perceived linearly. Thus, even the most simple regression functions mapping eye features to targets must be nonlinear to be relatively accurate. Among these techniques, we can cite: polynomial fitting (Morimoto and Mimica, 2005; Stampe, 1993), support vector regression (Zhu *et al.*, 2006), Gaussian processes (Sesma-Sanchez *et al.*, 2016), and artificial neural networks (Ji and Zhu, 2002).

Regardless of the regression method, feature-based estimation techniques are generally acknowledged for being very accurate in contrast to other methods, probably because the regressor learns more intrinsic information about the input data that cannot be easily

modeled in an explicit way, including device noise and sensing biases. But in practice, this also means that a large and representative set of points should be chosen from the target surface to achieve a calibration with low error.

More recently, with the growing success of end-to-end architectures in several applications, such as face recognition, voice recognition, or text translation, end-to-end gaze estimation also started to be vented as a possibility with many advantages. Among them, we could mention the lack of an intermediary and explicit feature selection process and the absence of calibration. For instance, Krafka et al. proposed a convolutional neural network that estimates the user's PoR on mobile phone displays without any calibration, though the reported accuracy still falls short of the SOtA techniques (Krafka *et al.*, 2016).

3.2.2 Gaze estimation and wearable computing

With the growth of virtual (VR) and augmented reality (AR) applications, gaze-based techniques have been considered as a means to refine and improve 3D interaction in such domains. AR applications, in particular, are generally assumed to be deployed in a wearable context. One important challenge, though, is that the AR domain occurs in the 3D space, and traditional estimation approaches might be limiting in the sense that they cannot accurately predict the user's point-of-regard (PoR) in 3D without taking vergence into account.

This issue is particularly evident when there is a partial occlusion or collinearity of 3D objects in the user's line of sight. Most known approaches in this case rely on determining the fixation target by computing the intersection of gaze direction with scene objects. This inevitably leads to a biased outcome, as the nearest objects in the line of sight will tend to be hit more often. One way of overcoming this issue is through "parallax tricks", such as requiring the user to fixate on the desired object from different perspectives.

Although gaze estimation in scene volumes has barely been explored in the literature, there have been some attempts to solve this problem imposing several constraints. In the case of remote gaze tracking, this is a more straight-forward task, since both eyeballs can be determined on camera space, as well as the κ_θ and κ_ϕ associated with the angular difference between optical and visual axes (Guestrin and Eizenman, 2006; Hennessey and Lawrence, 2009). In the case of head-mounted eye trackers, the works that explored this problem presented either a completely calibrated setup (Lidegaard *et al.*, 2014) or a very constrained calibration procedure (Abbott and Faisal, 2012). Ultimately, none of these techniques were applied to the camera scene volume.

There are only a few known techniques targeting specifically 3D environments. Yet, many of these are limited to gaze direction, with no accurate report of depth information. Datasets for this purpose are also lacking, since there has been only one work that considered scene depth knowledge, but from the perspective of a remote eye tracker (Mora and Odobez, 2014).

Mardanbegi and Hansen developed a method that enables the user to interact with planar displays in a 3D environment using a head-mounted eye tracker (Mardanbegi and Hansen, 2011). This method partially resorts to known calibration methods, but it also assumes there is a homographic mapping between the screen

on the scene camera image to the actual screen coordinates due to planarity constraints. A further development, using image features and fewer restrictions, was presented by Lander et al. (Lander *et al.*, 2015).

The earliest systems known capable of estimating 3D PoR required a fixed head-to-camera displacement. Kwon et al. introduced a novel binocular technique for this purpose, computing first gaze direction using corneal reflections and then gaze depth by interpupillary distance (Kwon *et al.*, 2006).

Works using wearable head-mounted systems usually resorted to triangulation of known features in an egocentric camera image. Mitsugami et al., for example, utilized view lines at multiple head positions to estimate 3D gaze (Mitsugami *et al.*, 2003), whereas others designed a non-real-time procedure to determine 2D PoR in some video frames, later integrating them to reconstruct the 3D PoR for posterior analysis (Munn and Pelz, 2008; Pfeiffer and Renner, 2014; Takemura *et al.*, 2010).

Abbott et al.(Abbott and Faisal, 2012) proposed a low-cost wearable eye tracker capable of gaze estimation in 3D space also using a model-based approach, but their calibration setup did not account for the parallax error and required previous knowledge about the position of both eyeballs in the scene during the procedure.

Essig et al. presented a feature-based approach that relied only on estimates generated by a neural network (Essig *et al.*, 2006). Measuring binocular gaze angles, their reported results showed significant improvement in comparison to a geometrical solution, especially regarding depth, but in a very controlled environment. More recently, (Itoh and Klinker, 2014) developed a technique to estimate gaze for HMDs using the Świrski and Dodgson algorithm (Świrski and Dodgson, 2013), which computes the optical axis by assuming the eyeballs as perfect spheres. If stereo vision is allowed, then Meyer et al. have proposed a more robust and lightweight technique to estimate normal pupil vectors (Meyer *et al.*, 2022). Other methods focused on the nature of gaze depth and its estimation (Duchowski *et al.*, 2011, 2014; Lee *et al.*, 2017). A general theory for 3D PoR estimation was provided by (Pirri *et al.*, 2011).

Still, despite computing the 3D PoR, all these approaches generally perform calibration to 2D planes, which, in the case of head-mounted eye trackers, gives room to the parallax error, created when the eye and the scene cameras are not coaxial (Mardanbegi and Hansen, 2012). A notable exception, perhaps, is the work of (Hennessey and Lawrence, 2009), as they proposed a way to compute the 3D PoR directly to a real-world 3D volume in real-time — albeit using a remote tracking system. This was accomplished by estimating the shortest distance between the two visual axis vectors, a strategy that was later used by (Abbott and Faisal, 2012).

Though the use of RGB-D cameras as a replacement for egocentric scene cameras may represent a solution to the current poor estimation of gaze depth and the parallax error, there are only a few approaches that have exploited this solution. Some works proposed to use RGB-D cameras for gaze estimation, but only to track the eyes (Li and Li, 2014; Mora and Odobez, 2014; Xiong *et al.*, 2014).

(McMurrough *et al.*, 2012) and (Paletta *et al.*, 2013) have both used a head-mounted setup with an RGB-D egocentric camera, but they limited themselves to perform only a 2D

calibration step for posterior analysis of gaze data overlaid in depth images, thus incurring on ambiguities associated with the lack of calibration to the scene volume.

In this thesis we argue that 3D PoR estimation in the wearable scenario is more advantageous in the *scene volume*, i.e., it should take advantage of the increasing ability to sense devices to reconstruct the physical scene in the digital domain, through techniques such as SLAM (Taketomi *et al.*, 2017), so that gaze interaction in 3D might be not only rid of volumetric ambiguities but also improved by patterns such as eye vergence.

3.3 Gaze interaction

The use of eye-gaze in human-computer interaction can be traced back to the 1980s, when eye trackers started to become reasonably accessible to conduct studies (Jacob and Karn, 2003). Back then, the primary use of gaze data was to understand human behavior when interacting with computers. Typical research questions revolved around where the user's attention is, or how much time is spent on specific parts of a graphical user interface.

Although in a limited fashion, it was also in the 1980s that some work pioneered the use of gaze as a means of interaction. Bolt, for instance, proposed to use the point-of-regard as a controlling element to trigger different functions on large displays with multiple windows (Bolt, 1981), whereas Ware and Mikaelian conducted experiments to investigate the use of the eye tracker as an input device (Ware and Mikaelian, 1987).

With the growth of computational power, better image processing, and gaze estimation algorithms, the prospect of using the eyes to directly or indirectly manipulate applications started to gain momentum in the HCI domain. Many researchers were attracted to the idea of a *non-command* interface (Jacob, 1995), i.e., interfaces that do not require explicit orders to provide expected actions.

But it also did not take long for researchers to realize that there were some daunting challenges to overcome when using the eyes for interaction. The most notorious one became known as the "Midas touch" problem (Jacob, 1991). Notwithstanding, other relevant eye tracking-related issues were also noticed, as the number of experiments and practitioners increased with time(Jacob and Karn, 2003):

- around 10 to 20% of the users cannot be tracked or tracked reliably;
- some movement constraints are required between the user and the tracking system for accurate measurements;
- positioning of the eyes is not always a conscious act;
- eye tracking is not as accurate and stable as other manual inputs;
- jitter and calibration shifts make data interpretation a non-trivial task;
- continuous use of deliberate eye gestures often leads to eyestrain.

In spite of these issues, gaze interaction still managed to make important strides. In particular, for people with severe disabilities, with conditions such as congenital mal-

formation, locked-in syndrome, or amyotrophic lateral sclerosis, the eyes may represent the only means to interact with the world, and gaze interaction research offered ways to successfully improve the quality of life of people with disabilities (Hutchinson *et al.*, 1989; Jacob, 1991).

Many authors also proposed solutions to mitigate or even overcome some of the aforementioned eye-tracking limitations. Studies revealed that the use of larger buttons in the interface is essential to attenuate the jitter of the eyes (Miniotas *et al.*, 2004). It has also been shown that dwell-time-based selection is an acceptable and simple approach to deal with the Midas touch problem and lack of visual feedback (Majaranta and Räihä, 2002; Penkar *et al.*, 2012), while others suggested the use of eye gestures (Mardanbegi *et al.*, 2012). More recently, smooth pursuit-based techniques have proven to be promising for small and ubiquitous interfaces, as they do not require calibration or tracking the user's head (Esteves *et al.*, 2015; Vidal *et al.*, 2013).

Investigations to improve the general performance of able-bodied users in daily tasks have also emerged. Multi-modal approaches became a common strategy in this sense, using the eyes for target acquisition and manual input for selection or other complex tasks (Kumar and Winograd, 2007; Pfeuffer *et al.*, 2014; Zhai *et al.*, 1999). To leave the user's hands completely free, there have been some proposals in the direction of coupling gaze and speech controls (Hedeshy *et al.*, 2021; Miniotas *et al.*, 2006) or using gaze with head gestures (Spakov and Majaranta, 2012).

But gaze interaction research was eventually dominated by studies with novel techniques targeting specific tasks. Eye typing became one such example, a still active topic, with a vast literature (Hansen *et al.*, 2002; Kurauchi *et al.*, 2016; Lystbæk *et al.*, 2022a; MacKenzie and Zhang, 2008; Majaranta and Räihä, 2002; Tula and Morimoto, 2016b). Gaze interaction in games has also attracted attention, especially with the affordable eye trackers available for the consumer market, and works exploring gaze-based controls in 2D and 3D games are also easily found (Agustin *et al.*, 2009; Dorr *et al.*, 2009; Isokoski and Martin, 2006; Istance *et al.*, 2010; Smith and Graham, 2006). Among many other applications, security authentication (Carneiro *et al.*, 2019; Luca *et al.*, 2007) and mobile-based controls (Drewes *et al.*, 2007; Park *et al.*, 2011) are also good examples of this area that continues to evolve, albeit with some enduring technical challenges.

3.3.1 Use of latent gaze in applications

Latent or passive use of gaze (Duchowski, 2018) has also been employed in several scenarios, though in a less extensive way. In this category, users are not aware that their eye behavior is a component of the system, but, almost as a rule, the eyes are not used for active interaction. In most cases, the eyes are a means to provide insightful information from the user and update the system state, such as with *foveated rendering* (Luebke and Hallen, 2001). Overall, the latent use of gaze might be a helpful tool in the wearable and ubiquitous context.

In DARPA's Augmented Cognition project, pupillometry was used as an indirect measure of mental workload for military systems (St. John *et al.*, 2004). Ishiguro et al. investigated a way to artificially augment user memory by resorting to a gaze-based *lifelog*,

i.e., recording daily image scenes based on fixations, saccades, and blinks (Ishiguro *et al.*, 2010). Kurauchi et al. also studied a means to enhance memory by providing information from someone being gazed at by the user (Kurauchi and Morimoto, 2013).

3.3.2 Gaze interaction in mixed reality environments

Although gaze interaction has been considered somewhat of a niche in the last decades, this perception is shifting with the emergence of commercial mixed-reality headsets. A survey from Zhou et al. clearly indicated that until the mid-2000s research in mixed reality mainly focused on developing enabling technologies. With time, as these technologies matured, it became evident that there were many interaction and visualization problems to be addressed (Zhou *et al.*, 2008).

One of the early works exploring gaze interaction in mixed reality environments was proposed by Tanriverdi and Jacob (Tanriverdi and Jacob, 2000). In it, they investigated how gaze could be utilized to point and interact with VR elements. They concluded that gaze pointing was significantly faster than hand-based controls, though it elicited less environment recollection. In a similar experiment, Cournia et al. compared gaze with hand pointing, and observed that the former was slightly slower (Cournia *et al.*, 2003).

Other works have also explored the use of gaze in VR for navigation purposes. Castellina and Corno, for instance, compared different combinations of gaze input controls with and without the use of a keyboard (Castellina and Corno, 2008), while Nacke et al. proposed to control the first-person view camera for gameplay through eye tracking only (Nacke *et al.*, 2010). Still in the VR domain, Garau et al. investigated the impact of avatar realism on the perceived quality of communication between users and they realized that the more realistic an avatar, the greater the expectation for equally realistic eye movements (Garau *et al.*, 2003). Later on, Steptoe et al. conducted an experiment demonstrating that users perform socially-dependent tasks better when the companion's gaze is being tracked and available (Steptoe *et al.*, 2009).

Nilsson et al. presented one of the first works exploring gaze interaction in augmented reality (Nilsson *et al.*, 2009). The idea was to observe how task performance could be improved using gaze controls when the user's hands were occupied. Ajanki et al. went a step further, proposing an attention-based AR interface: virtual data was only displayed when the user was looking at specific regions of the image captured by a scene camera (Ajanki *et al.*, 2011).

Park et al. designed one of the first wearable augmented reality systems equipped with a monocular eye tracker (Park *et al.*, 2008). In their setup, a simple calibration between the eye and scene camera (2D) images was performed and a dwell-time-based selection was implemented to trigger virtual information about an AR gallery. Toyama et al. used a similar approach to develop a gaze-guided OCR system (Toyama *et al.*, 2014).

More recently, some works explored the use of gaze depth as a means of interaction, but without estimating the user's point-of-regard in 3D (Kitajima *et al.*, 2015; Pai *et al.*, 2016). Another trend observed was regarding smooth pursuits-based interaction (Vidal *et al.*, 2013). Esteves et al. introduced SmoothMoves as a means to perform selections in AR using head-gaze (Esteves *et al.*, 2017). Using a purely gaze-based interface, Khamis et

al. investigated the use of pursuits in VR and observed that large trajectories result in higher accuracy and faster selections (Khamis *et al.*, 2018), which somewhat agrees with the idea that gaze-only interaction tends to improve with a larger FoV (Blattgerste *et al.*, 2018).

3.4 Challenges and opportunities

3.4.1 Gaze and mixed reality: a wearable match

In general, a gaze-enabled system is considered a more private way to interact, it can be hands-free, and it is often perceived as more effortless and socially acceptable than other popular alternatives such as voice utterances or hand gestures (Akkil *et al.*, 2016). Besides, in MR environments visual attention is a key component to determine interactive context, which can vary significantly, so some form of eye tracking is invaluable and thus expected (Höllerer and Feiner, 2004).

On the other hand, there are some legitimate concerns about how gaze should be used interactively in this context. For instance, eye contact is critical in face-to-face communication, so soliciting unnatural gaze responses in social environments is not desirable (Akkil *et al.*, 2016). Additionally, the Midas touch problem can be much more worrying in AR spaces, as it can hamper visibility and user safety, whereas eye fatigue is commonly linked to eye gesture-based techniques and inexperienced users (Chitty, 2013).

Low accuracy of gaze estimation is also a source of concern because it can severely affect the experience. For example, Pathmanathan et al. investigated the use of gaze-based multimodal techniques for the manipulation of 3D objects, such as translation, rotation, and scaling (Pathmanathan *et al.*, 2020). In general, participants preferred head-gaze as they felt they had more control than solely with the eyes. Although it is true that the number of errors and variability on task performance can be large with gaze-only interaction, Blattgerste et al. showed, however, that as the field of view increases, gaze-only interaction is preferred over head-gaze by users, since it becomes distinctively faster, less strenuous and less error-prone (Blattgerste *et al.*, 2018).

To tackle the problem of low accuracy with eye tracking in AR, Kytö et al. developed a technique called *Pinpointing*, which aims to provide precision for selections by a multimodal refinement mechanism (Kytö *et al.*, 2018). With a combination of eye-for-pointing and hand gestures, they were able to achieve an average error of 0.4 degrees in contrast to the 2.4 degrees of gaze-only selection, though this ultimately implies twice as long selections. A similar conclusion was also found by Qian and Teather, that is, eyes alone have a higher throughput but at the expense of a greater number of mistakes (Qian and Teather, 2017). More recently, a similar multimodal idea was explored in the specific context of menu selection in AR (Lystbæk *et al.*, 2022b).

With StARe, Rivu et al. proposed a user interface in AR that displays information about people during conversations (Rivu *et al.*, 2020). This kind of solution, although perceived as useful for most users, raises serious concerns in terms of AR cluttering and information overload. Besides, just looking at an object to trigger related AR content will always be

a source of the Midas touch problem. Another issue related to a collaborative setting is target ambiguity shared by multiple users in large spaces, which was addressed by Li et al. through a technique called Parallel Bars (Li *et al.*, 2019).

Discussing more broad aspects of gaze interaction in MR, Hirzle et al. suggested that the design space for MR headsets should take into account the properties of the HMDs (i.e., stereoscopic, monoscopic) and human depth perception (i.e., binocular, monocular) as its two main dimensions of concern (Hirzle *et al.*, 2019). With ARtention, on the other hand, Pfeuffer et al. indicate that the design space of AR gaze-based experiences should consider the reality-virtuality continuum, information level, and task transitions (Pfeuffer *et al.*, 2021).

Also, not all gaze-based AR experiences have to be visual, i.e., use the headset display. Bâce et al. investigated a means to relay messages to other users through physically augmentable objects in the environment. These messages can be posted and accessed using notifications on the smartwatch (Bâce *et al.*, 2016).

Most gaze-based techniques found in the MR space tend to be 3D ports from already known 2D implementations. This means that very few of them are actually incorporating hardware and contextual features such as inertial sensors of augmented glasses or the depth dimension of 3D spaces, with very few exceptions (Piumsomboon *et al.*, 2017).

3.4.2 Gaze-contingent displays and vergence-based interaction

Because gaze is highly correlated to human attention, the idea of gaze-contingent displays started to get traction with the increasing interest of the industry for AR/VR headsets or more lightweight AR glasses in more recent years. Gaze-contingent displays can be generally understood as any kind of display that leverages the user's gaze point to update its status or provide visual information to the user.

For example, to provide a more realistic experience in VR, the user's eyes can be tracked to render perceptually relevant effects, such as ocular parallax or stereo rendering, which have been demonstrated to have a significant impact on user depth perception (Kellnhofer *et al.*, 2016; Konrad *et al.*, 2020; Krajancich *et al.*, 2020). Other forms of fostering realism are through super-resolutions or improved space sampling while rendering real-time graphics depending on where the user is looking at (Guenter *et al.*, 2012; Stengel *et al.*, 2016), or simulating depth-of-field effect (Mauderer *et al.*, 2014; Vinnikov and Allison, 2014).

Of course, to provide better sampling or super-resolution regions it is also important to understand raw eye movement patterns, such as fixations and saccades, and predict where the gaze is going to land before a saccadic burst. Arabadzhiyska et al. created a polynomial model for gaze-landing prediction (Arabadzhiyska *et al.*, 2017), while Morales et al. proposed a neural network-based model that achieved the current state-of-the-art performance on this problem (Morales *et al.*, 2018).

Another important application of gaze-contingent displays is related to a scenario of the pervasive use of eye tracking: through attention inference mechanisms, such as

gaze direction, fixation duration, and pupil size, an AR system might be able to provide relevant information associated with the scene (Ajanki *et al.*, 2011; Rivu *et al.*, 2020; Toet, 2006). However, to understand what is relevant or not, it is also important to ascertain context, which is a somewhat vague concept, but generally accepted as any information that characterizes the situation of an entity (Grubert *et al.*, 2017), although it can be further narrowed down to specific parameters, such as location, identity, and activity (Abowd *et al.*, 1999).

More recently, eye vergence has been considered a key feature to understanding attention shifts, particularly with respect to task-unrelated states, such as mind wandering (Huang *et al.*, 2019). However, vergence can only be a useful resource for this kind of display if gaze depth measurement is taken into account. Vidal et al. suggested, for example, that vergence could be an important mechanism for see-through displays to define the moment when augmented information should be rendered or not (Vidal *et al.*, 2014).

As a means of interaction, vergence has been barely explored in the literature. Early works only discussed its feasibility (Kirst and Bulling, 2016; Kudo *et al.*, 2013; Ruan *et al.*, 2018; Vidal *et al.*, 2014). With the lack of effective techniques to estimate 3D PoR Pfeiffer *et al.* (2008), it has been suggested that 3D fixations in HMDs can be derived solely from pupil disparity (Hirzle *et al.*, 2018), or using Purkinje images from both eyes (Kuroda *et al.*, 2010). One of the first vergence-based applications was proposed by Kuroda et al., a system that performs AR annotations for medical use (Kuroda *et al.*, 2010). Kitajima et al. suggested an AR x-ray vision experience through the user's gaze depth (Kitajima *et al.*, 2015), while Hirzle et al. also considered a similar application (Hirzle *et al.*, 2019), but using a head-mounted eye tracker. They found out that reference points in 3D space are important to improve user comfort while performing convergence or divergence.

Pai et al. were the first to demonstrate "gaze focus depth" as an interactive modality (Pai *et al.*, 2016). They conducted a case study to evaluate the usability of continuous focus depth as a substitute for the scroll wheel. Their system consisted of a VR headset coupled with a binocular Pupil Labs eye tracker (Kassner *et al.*, 2014). Ahn et al., also in VR, proposed a system in which a visual target oscillates in depth, and selection occurs once the system detects the user's gaze modulated by the target (Ahn *et al.*, 2020).

3.4.3 Wearable context and *calm* interaction

Looking back on all the literature on gaze-based interaction, it should strike any attentive reader how rare is to find just one technique that does not impose some sort of constraint to spontaneous eye movement response to avoid the Midas touch problem (Jacob, 1991). Since most techniques require deliberate and conscious eye movements to change the system's state, they also force the user to relinquish the eyes' main functions of exploring, searching, or reading as a consequence, i.e., they restrain the natural gaze behavior with respect to the environment.

In a way, we could say that these techniques break the user flow state. In neuroscience, this is often described as an effortless, yet focused state of consciousness Dietrich (2004). Thus, interactive safeguards for gaze interaction may hamper this state by requiring the

user to consciously divert his or her attention from the task at hand to avoid the MTP.

In the wearable context, these mechanisms to avoid interactive mistakes might even inadvertently put user integrity at risk. For instance, techniques that rely on either dwell-time selection (Majaranta and Räihä, 2002; Penkar *et al.*, 2012) or that require saccadic gestures (Mardanbegi *et al.*, 2012; Spakov and Majaranta, 2012) might add latency to environmental response or divert the user attention in situations such as running down the street, going up the stairs, or driving a vehicle. Other methods, such as gaze motion correlation (Esteves *et al.*, 2015; Velloso *et al.*, 2017), that are constantly displaying a moving target on the screen, are particularly perilous in the case of heads-up displays since digital overlays can occlude user perception of the environment. Multimodal methods, such as *gaze and clicking* (Drewes and Schmidt, 2009), could be a safer alternative, but they typically impose additional cognitive overhead and are much less private than gaze-only techniques in public spaces.

Weiser conjectured that the ultimate interface should be "transparent" to the user, that is, it should be perceived as invisible, an idea conceptually close to what he later defined as *calm computing* (Ishii and Ullmer, 1997; Weiser, 1998). Although passive use of gaze in gaze-contingent displays resembles something of the sort, it generally does not convey user explicit intent, and thus has only been successfully exploited in gaze-contingent displays or adaptive systems (Ajanki *et al.*, 2011; Rivu *et al.*, 2020; Toet, 2006).

In a wearable scenario, we argue that gaze techniques that embed "calmness" in their design are more likely to thrive. Not because this has been suggested as an ideal and general goal by past authors, but because the wearable context imposes important limitations and peculiarities, namely the excess of micro-interaction situations (e.g., turning on/off smart devices), user exposure to environmental danger, or privacy in public spaces. Thus, it could be considered desirable for any gaze-based interactive method in wearable computing to possess characteristics such as transparency, unobtrusiveness, speed, and discretion.

Chapter 4

Eye pattern recognition

Eye movement pattern recognition is a key aspect of human-computer interaction, particularly in the context of wearable computing where devices are often worn continuously throughout the day. Wearable devices have the potential to enable new forms of interaction, but to achieve this, it is essential to accurately and reliably recognize the eye movements associated with different tasks and user engagement levels. By analyzing eye movement patterns, it is possible to infer a user's intentions and level of attention, as well as provide feedback to improve task performance, thus creating more seamless and intuitive user experiences.

In this chapter, we focus on eye movement classification algorithms for various scenarios based on the research opportunities unveiled by our previous literature review. In particular, we present three investigations on the following topics: offline classifiers for pre-recorded eye movement data, online classifiers for real-time eye movement data, and reading detection pattern recognition for identifying reading activities.

These investigations are reported as self-contained research projects, with their own goals and conclusions, in each section of this chapter. However, they do represent a more general effort to fulfill our goal of supporting better context-aware systems for wearable computing through improved eye pattern recognition techniques.

Among our contributions, we propose novel models for basic eye movement recognition and online classification, both of which surpass state-of-the-art performance. Our online model, in particular, achieves this while being extremely lightweight, making it suitable for deployment on resource-constrained devices. We also introduce a reading detection algorithm that can operate at very low frame rates while maintaining state-of-the-art accuracy. Additionally, we created a new dataset for eye movement classification, which is stimulus-driven and it is publicly available at `https://github.com/elmadjian/OEMC`.

4.1 Offline eye movement classification

The simultaneous classification of fixations, saccades, and smooth pursuits, or simply the 3EMCP, is one of the most fundamental problems when it comes to eye movement

pattern recognition (Berndt *et al.*, 2019), as most of the more complex eye patterns in several gaze-based applications depend on it.

Inspired by the 1D-CNN-BiLSTM model from Startsev et al. (Startsev *et al.*, 2019b), we investigated whether more lightweight and modern architectures that do not rely entirely on sequence-based processing could further improve basic eye movement recognition. In particular, we conjectured whether a simple but effective architecture such as a Temporal Convolutional Network (TCN) could achieve this goal. The complete description of the techniques, results, and analysis provided below was first published in Elmadjian *et al.* (2020).

4.1.1 Materials and methods

The dataset employed in our study was the GazeCom dataset (Dorr *et al.*, 2009). It contains 18 clips from roughly 47 viewers (the number slightly varies according to the video), with labels for fixations, saccades, smooth pursuits, and noise. The labels are given based on the judgment and agreement of two experts, after observing the patterns that each user performed when watching short videos. Clips are limited to a duration of 21 s for training.

The GazeCom data was collected using a 250 Hz remote eye tracker, and is constituted by 4.3 million samples, being 72.5% fixations, 10.5% saccades, 11% smooth pursuits, and 5.9% noise or blinks. The average confidence level given by the eye tracker is 99.4%. The average duration of each pattern is shown in Table 4.1.

	Duration (ms)	**Deviation (ms)**
Fixations	324.37	306.07
Saccades	46.38	22.17
Smooth Pursuits	410.98	323.42
Noise	291.91	294.34

Table 4.1: *Average event duration in the GazeCom dataset (Dorr et al., 2009) with clips limited to 21 s.*

In this first step of our investigation, we resorted to a non-causal TCN, which gives it the ability to look into future information at the expense of being offline. We also made use of several dilated convolutions (1, 2, 4, 8, 16, 32, 64, 128), and a relatively sizeable kernel ($k = 8$), which effectively provided a large receptive field (see Figure 4.2). Despite having almost 2 million parameters, this network trained comparatively faster due to its parallelized structure, roughly taking half of the time required to train the 1D-CNN-BLSTM (Dorr *et al.*, 2010) on the same machine.

The schematic of this architecture is depicted in Figure 4.1. All hyperparameters were defined according to a reduced grid search with a subset of the GazeCom dataset. The TCN has a total of 128 filters, an internal dropout rate of 0.3, and uses the hyperbolic tangent activation function (*tanh*) instead of the more commonly seen rectified linear units (*ReLU*). Weights were initialized following a random uniform distribution, and we do not employ batch normalization between layers. At the end of the convolutional chain, we added a fully connected layer with a softmax activation function wrapped by a timed-distributed layer,

so that we can have a certain probability distribution for each eye movement pattern at all time steps. Because of our one-hot encoded targets, we used a categorical cross-entropy loss function. We also opted for the Adam optimizer (Kingma and Ba, 2015) with an initial learning rate of 0.0001, as this configuration demonstrated a faster convergence in our grid search.

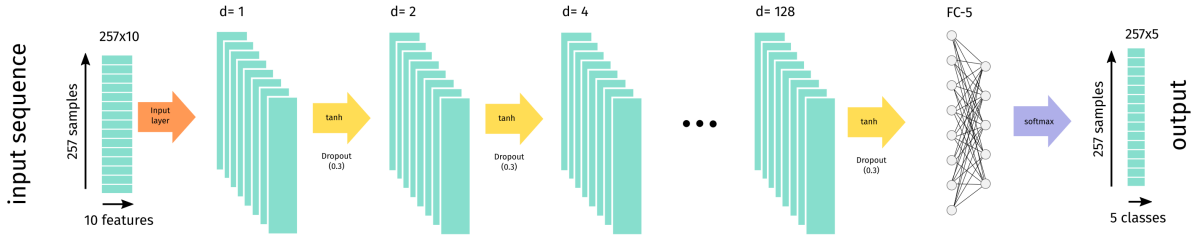

Figure 4.1: *Architecture of our TCN network. At each hidden state, we increase dilations by a factor of 2. FC-5 represents a time-distributed fully-connected layer with 5 outputs.*

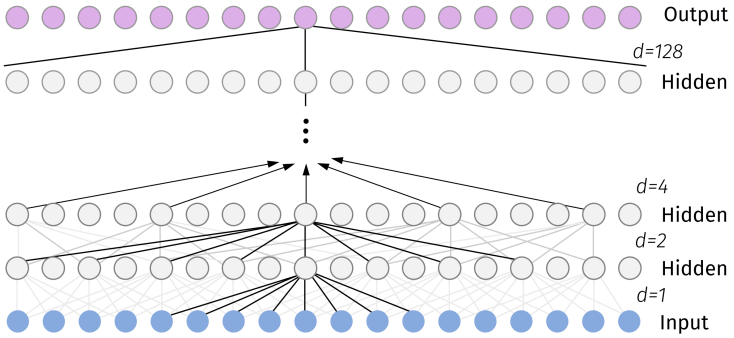

Figure 4.2: *Non-causal dilated convolutions using size 8 kernels, providing a large receptive field and the ability to look into future information (i.e., t_{i+n} samples given a timestamp t_i).*

The temporal sequences fed to the model are formed by pre-computed multi-scale features extracted from the x and y gaze coordinates. In their original implementation, Startsev et al. considered five different temporal scales: 4, 8, 16, 32, and 64 ms. In ours, we included support for a larger set of scales (128, 256, and 512 ms) to investigate their role in model classification performance.

The feature extraction process is depicted in Figure 4.3. All these features are created by pre-processing the entire set of raw gaze coordinates before training. Each sample consists of a fixed-size context window of roughly 1 s (257 steps), and each timestep of this window presents the associated multi-scale set of features, which could be either acceleration, speed, direction, or a combination of them. Additionally, we investigated the impact of adding two other common features in the time series domain: standard deviation and displacement (not to be confused with distance).

We evaluated the general performance of our model using the same configuration and metrics from Startsev et al. To compare our model with the baseline, we used a Leave-One-Video-Out (LOVO) cross-validation procedure, in which each step of training is done with

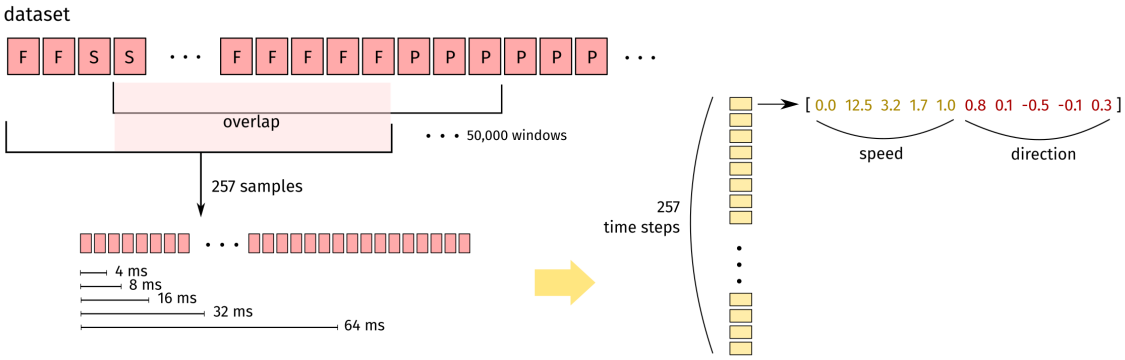

Figure 4.3: *Feature extraction procedure from the GazeCom dataset (Startsev et al., 2019b). For each 257-sample window, we extract speed and direction features related to each step of the sequence, using different scale sizes.*

17 clips, and the model is evaluated on the remaining one. We also employed the input setup with a context window of 257 samples (~1 s), an overlap of 192 samples, with speed and direction features combined, and limiting the number of training sequences to 50,000, using a random permutation of sequences with the same seed.

Assuming the standard feature space (SF) as the combination of speed and direction, we evaluated the TCN model performance in this aspect according to the following configurations: SF + standard deviation; SF + displacement; and SF + standard deviation + displacement. Regarding the feature scale size investigation, we evaluated the model with additional scales of 128, 256, and 512 ms. We used the same LOVO procedure while training using the same context window size, training sequences, batch size, and epochs, as reported before.

Since smooth pursuits have a typically longer duration than fixations and saccades, it has been hypothesized that larger context windows can benefit *seq2seq* models to improve their detection. To demonstrate this, we trained our model using two additional windows of 1.5 s and 2 s, using the same LOVO training procedure, but with the overlap having to be adjusted to keep the 50,000 sequences fixed. In other words, we used 385 samples with an overlap of 312 for the 1.5 s window, and 514 samples with an overlap of 470 for the 2 s window.

It is worth mentioning that larger contexts can be prohibitive for recurrent networks regarding memory and computational complexity, but there were no such issues training with the TCN model. Based on these findings, we ran a final LOVO evaluation combining the best configurations provided by the study on feature space, feature scale size, and temporal window width.

Finally, we also evaluated all models with and without a domain-based knowledge filtering procedure applied to the outputs. This heuristic assumes that any event that is shorter than 12 ms is a spurious output. The filter then replaces the class of the samples belonging to this event with the previous non-spurious neighbor. All our code base, data, and models trained for this work are publicly available at `https://github.com/elmadjian/3EMCP-with-TCNs`.

4.1.2 Evaluation and results

Compared to the reported results of the 1D-CNN-BLSTM model, and using the same input constraints and training regimen, our model achieved improved performance in all three major patterns, with an F-score of 94.2% for fixations, 89.9% for saccades, and 73.4% for smooth pursuits. This represents a gain over the state of the art in the GazeCom dataset of 0.3%, 0.3%, and 3%, respectively.

By extending window width, feature space, and feature scale size, we were able to achieve gains of 0.6%, 0.3%, and 5.7%, respectively. A complete assessment compiling the reported results in terms of sample F1 and event detection (episode F1 with IoU > 0.5) is shown in Table 4.2. The TCN model marked by a + sign is the one trained with the best features, while the ones marked with a * had their output filtered.

Model	Fixation				Saccade				Smooth Pursuit			
	F1	Prec.	Recall	EF1	F1	Prec.	Recall	EF1	F1	Prec.	Recall	EF1
TCN Model+	**0.945**	**0.929**	0.961	0.735	0.894	0.899	**0.889**	0.888	0.762	0.787	**0.739**	0.253
TCN Model+*	**0.945**	0.928	0.962	**0.900**	0.892	0.897	0.887	0.941	**0.764**	**0.791**	**0.739**	**0.608**
TCN Model	0.942	0.925	0.959	0.717	**0.899**	**0.903**	0.894	0.885	0.737	0.765	0.711	0.234
TCN Model*	0.942	0.924	0.960	0.896	0.896	0.901	0.890	**0.943**	0.739	0.769	0.711	0.602
Startsev et al. (Startsev *et al.*, 2019b)	0.939	0.914	**0.967**	0.868	0.893	0.897	**0.889**	0.924	0.703	0.788	0.634	0.484
Startsev et al.*	0.939	0.914	**0.967**	0.883	0.893	0.897	**0.889**	0.935	0.703	0.789	0.634	0.537
Startsev et al. (Startsev *et al.*, 2019c)	0.937	0.921	0.954	0.882	0.896	0.899	0.893	0.929	0.707	0.724	0.691	0.544
Startsev et al.*	0.937	0.921	0.954	0.892	0.896	0.899	0.892	0.939	0.708	0.725	0.692	0.576

Table 4.2: *TCN Model Evaluation. Results for sample-level detection, except column **EF1**=Event F1 (IoU>= 0.5). Rows marked with * represent the filtered output. **Prec.**= Precision. + = Our best model.*

The use of domain knowledge-based filtering for the outputs also increased event detection scores according to the metrics proposed by Hooge et al. (Hooge *et al.*, 2018). With smooth pursuits, in particular, the TCN models roughly tripled their event F-scores, while the models from Startsev et al. had only modest improvements. Besides, the filter effect over general sample-level scores was barely noticeable.

The evaluation of feature space and feature scale size has shown mixed results. It is clear that increasing the scale size led to an overall marginal improvement in classification performance when compared to the default 5-scale features input. However these gains did not occur consistently across scale sizes, and smooth pursuits showed the largest fluctuations. Figure 4.4 exhibits how the F1 and IoU values for smooth pursuits oscillated with respect to the scale size. Tables 4.3 and 4.4 show the results related to the feature space study and feature scale size, respectively.

The investigation of the size of the temporal window partially confirmed the hypothesis that smooth pursuits detection can benefit from larger contexts. With a window size of 1.5 s (385 samples), there was an improvement of 1.3% over the 1 s window (257 samples), but with a context of 2 s (514 samples), the gain dropped to 0.7%. These results can be seen in Table 4.5.

A final evaluation with the best configuration in terms of features and context windows gave us the highest F1 scores achieved for smooth pursuits in the GazeCom dataset. The results were 94.5% for fixations, 89.2% for saccades, and 76.4% for smooth pursuits.

	Fixation		Saccade		SP	
	F1	**EF1**	**F1**	**EF1**	**F1**	**EF1**
win 257	0.942	0.717	0.899	0.885	0.737	0.234
win 257*	0.942	**0.896**	0.896	0.943	0.739	**0.602**
win 257+std	0.942	0.721	**0.898**	0.890	0.738	0.239
win 257+std*	0.942	**0.896**	0.895	**0.945**	0.739	0.601
win 257+disp	0.942	0.728	0.896	0.880	0.738	0.252
win 257+disp*	0.942	0.895	0.893	0.943	0.740	0.596
win 257+std+disp	0.942	0.730	0.896	0.883	0.740	0.253
win 257+std+disp*	0.942	**0.896**	0.894	0.944	**0.741**	0.600

Table 4.3: *Feature space study.* ***EF1****= Event F1 (IoU >= 0.5), *=Filtered*

	Fixation		Saccade		SP	
	F1	**EF1**	**F1**	**EF1**	**F1**	**EF1**
5 Features	0.942	0.717	**0.899**	0.885	0.737	0.234
5 Features	0.942	0.896	0.896	**0.943**	0.739	0.602
6 Features	**0.943**	0.723	**0.899**	0.887	0.743	0.240
6 Features*	**0.943**	**0.897**	0.896	**0.943**	0.745	**0.613**
7 Features	0.942	0.725	0.898	0.884	0.745	0.244
7 Features*	0.942	0.896	0.895	**0.943**	**0.746**	0.597
8 Features	0.942	0.715	0.897	0.885	0.742	0.235
8 Features*	0.942	0.894	0.894	0.941	0.745	0.599

Table 4.4: *Feature scale study.* ***EF1****= Event F1 (IoU >= 0.5), *=Filtered*

	Fixation		Saccade		SP	
	F1	**EF1**	**F1**	**EF1**	**F1**	**EF1**
Win 257	0.942	0.717	0.899	0.885	0.737	0.234
Win 257*	0.942	0.896	0.896	**0.943**	0.739	0.602
Win 385	**0.943**	0.708	0.899	0.884	0.749	0.227
Win 385*	**0.943**	0.898	0.897	**0.943**	**0.751**	0.602
Win 514	0.941	0.694	**0.900**	0.886	0.744	0.213
Win 514*	0.942	**0.898**	0.897	**0.943**	0.747	**0.604**

Table 4.5: *Temporal window width. Results for varying temporal window sizes (257, 358, and 514 samples) of the TCN model. Results marked by * indicate an identical trained model with filtered output.*

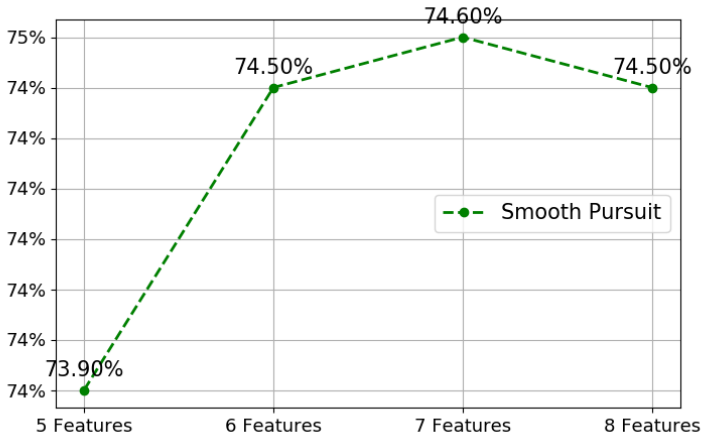

Figure 4.4: *Smooth pursuit F-score variation across different scale sizes.*

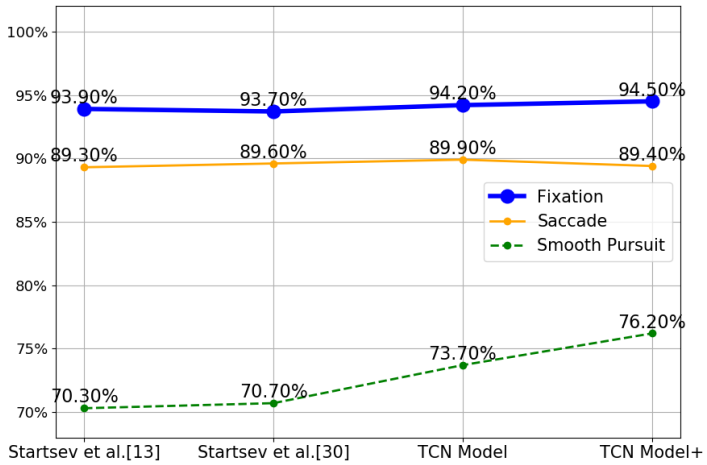

Figure 4.5: *Results of our TCN model for fixations, saccades, and smooth pursuits (SP), along with the reported results from the 1D-CNN-BLSTM model.*

4.1.3 Discussion and conclusion

Our results show that the TCN architecture outperforms the state-of-the-art model with the GazeCom dataset. The most significant improvement was in the classification of smooth pursuits, with a 3% performance increase, or 5.7% increase over previously published results when considering the increments in context window width, feature space, and scale size.

The investigation on feature space and scale size revealed that increasing both dimensions were somewhat beneficial to the TCN classification performance. The impact of adding novel features, although favorable, was comparatively less significant than the increase in feature scale size, as the latter provided comparatively higher scores on the GazeCom dataset, particularly with respect to smooth pursuits.

The improvement, though, was not consistent with size increments. It could be argued that saccade detection had no improvement at all, while smooth pursuits had the highest scores with a 7-scale size instead of the largest 8-scale one. Based on our experiments, it is still not clear whether the network capacity was not appropriate for a larger set of input scales using the same hyperparameters, or whether these inputs naturally have a detrimental effect on training when reaching a certain length.

A similar phenomenon was observed when examining the effect of larger context windows. Although training with both 1.5-second and 2-second temporal windows resulted in improved classification performance, the evaluation on the 1.5-second context presented the general highest scores. It is clear that smooth pursuits had an increased detection rate with larger contexts, but these results do not support the hypothesis that smooth pursuit classification scores increase proportionally with window size.

The filter heuristics were also demonstrated to be fundamental when it comes to event detection with TCNs. This improvement could be explained by the fact that convolutional architectures are not favored by the sequential learning constraints of recurrent nets, thus often tainting an event block with one or other wrong predictions. This, of course, can be harmful depending on the metric chosen for event assessment, but it barely affects sample-level classification accuracy.

Overall, we demonstrated that our model was capable of surpassing the previous state-of-the-art architectures on the same dataset using the same metrics. Moreover, TCNs present some intrinsic advantages such as a lower memory footprint, faster training, and the ability to learn sequences of arbitrary length.

4.2 Online eye movement classification

Little is known about the feasibility and performance of deep neural architectures for online eye movement classification (see Figure 4.6). Moreover, previous research with such models has been done with offline preprocessing techniques that limit the applicability of deep neural architectures in an online setting.

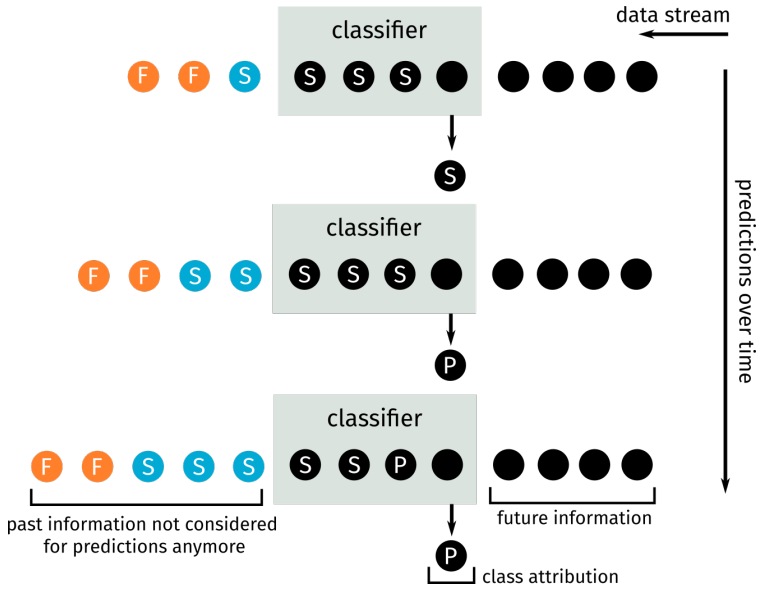

Figure 4.6: *In the online problem, a classifier does not have access beforehand to future samples and cannot take too long to make a prediction. Therefore, transitioning patterns and high-frequency data will usually be more challenging to handle. Here, we see an example of an online classifier that predicts one sample at a time from sequences of a fixed size. The symbols F, S, and P stand for fixations, saccades, and smooth pursuits, respectively. Colored samples indicate already classified samples that are not considered for predictions anymore.*

In this study, we devised a few novel techniques aiming to improve not only the general accuracy in the online 3EMCP but also to turn highly performant models feasible within constrained wearable settings. The full version of this investigation was first published in Elmadjian *et al.* (2022) by Springer Nature.

4.2.1 Materials and methods

Our main architecture is an online TCN (Bai *et al.*, 2018). The proposed model is composed of four hidden *temporal blocks*, with 30 units each. Each block comprises two one-dimensional convolutional layers and a residual connection. Activation between layers is done via *ReLU* function, and four different dilation sizes $(1, 2, 4, 8)$ are used. Unlike our previous model, all convolutions are causal, i.e., any output at time t is only convolved with elements up to time t, which ensures no leakage of future information. Zero-padding is employed to make the outputs of the temporal blocks have the same length as the inputs (see Figure 4.7).

The datasets in which we evaluate the TCN and other baseline models are the GazeCom (Dorr *et al.*, 2009) and the HMR, which we created to overcome the problem of the

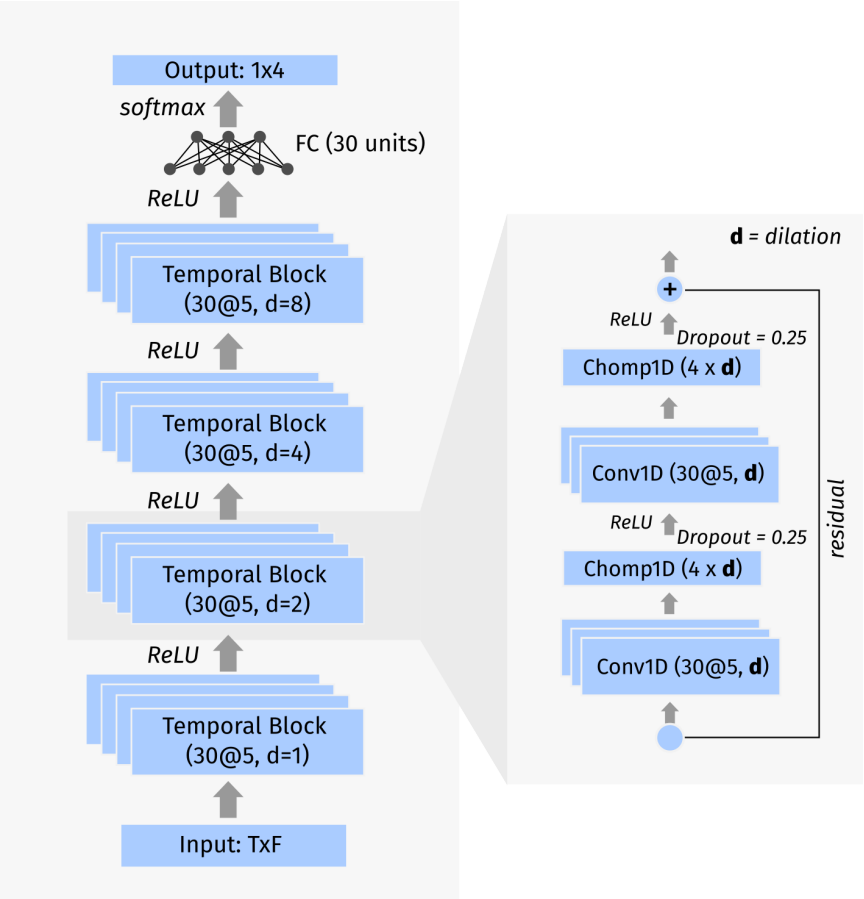

Figure 4.7: *Overall schematics of the causal TCN, comprised of temporal blocks. Unlike our previous investigation, this architecture is of the form seq2one. The letters T and F refer to time step and feature dimensions, respectively. The symbol @ is used to indicate the kernel size applied to convolutions.*

absence of stimuli-driven datasets on this topic, as it can help mitigate the bias of expert annotators. The HMR dataset is composed of data from 13 participants (6 female) using a head-mounted 200 Hz Pupil Core eye tracker (Kassner *et al.*, 2014), with a reported accuracy of 0.6^o, and resolution of 192×192 pixels. Their ages varied from 21 to 46 years old, and all of them had a normal or corrected-to-normal vision. Two had a previous experience with eye trackers. For each participant, we collected the normalized pupil center data independently from left and right eyes in the eye camera space, totaling 26 video files. Each file has roughly 5 minutes, giving us almost 2 hours of recording.

In terms of statistics, the HMR dataset comprises 1.49 million samples, of which 56.3% are fixations, 6.4% saccades, 25.4% smooth pursuits, and 11.7% blinks. The average confidence level is 88.9%, automatically provided by the Pupil Capture software, and almost all confidence level drops are correlated to blink events. Table 4.6 shows a comparison between each dataset.

To make training and evaluation suitable for online classification, we proposed a multi-scale one-way feature extraction procedure that gives more importance to more recent samples. This method avoids leakage from future events to the past by calculating speed and direction in different time frames, taking the most recent sample of a context window

Eye Movement	GazeCom		HMR	
	Samples	Events	Samples	Events
Fixations	72.55%	44.93%	56.30%	45.25%
Saccades	10.53%	45.61%	6.46%	28.57%
Smooth Pursuits	11.02%	5.39%	25.47%	9.53%
Noise / Blinks	5.90%	4.07%	11.77%	16.65%

Table 4.6: *Ratio of samples and events in each dataset.*

as our anchor point, and then performing a series of strides from there to extract the features. Figure 4.8 illustrates this procedure.

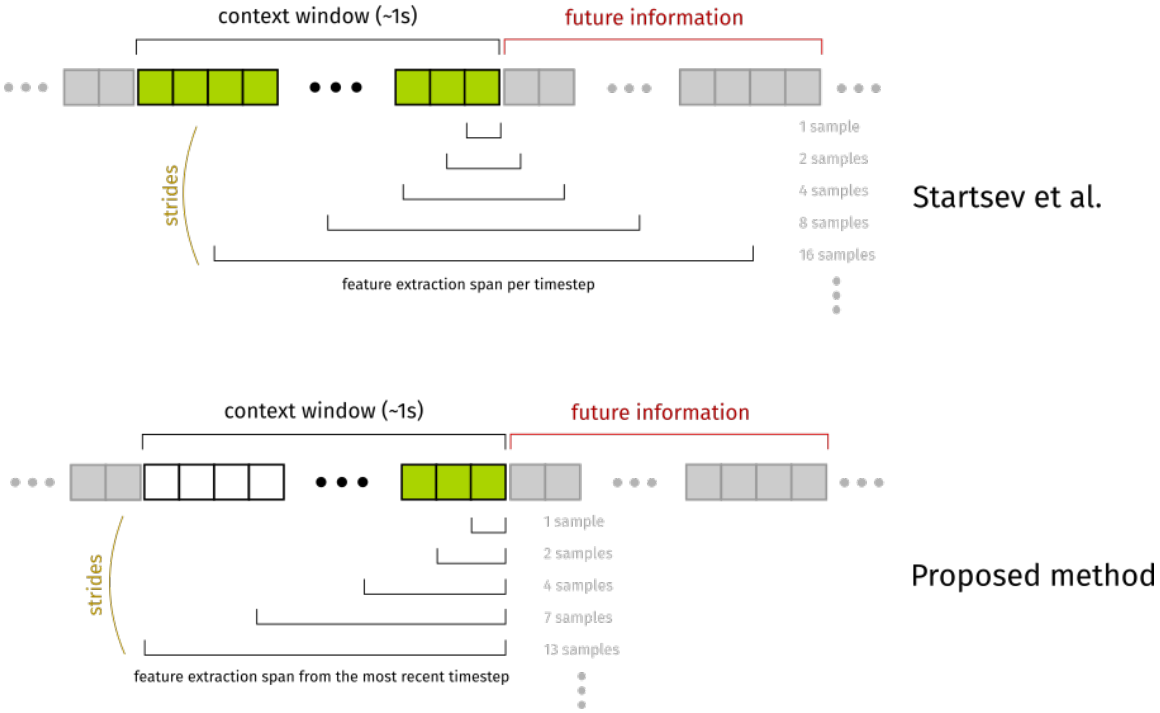

Figure 4.8: *With previous deep neural models (Startsev et al., 2019b), the multi-scale feature extraction step resorted to capturing information that is beyond the context window, with leakage of future information, and considering all possible time steps (marked in green). Our process, on the other hand, gives considerably more importance to the most recent samples, processing features only within the delimited context window and using only a fraction of time steps to minimize the response delay.*

A grid search determined the best number of strides used within a context window of 1 s for each dataset. For the HMR, we calculated 8 sets of different scales (strides) for speed and direction, whereas for the GazeCom we used 10 strides. The scale size is continuously computed from the latest sample in the context window (i.e., our anchor point) with respect to the less recent samples, with the scale increasing exponentially according to the following equation:

$$\text{stride}(x) = \lceil 2^{x-s} \times n \rceil, \{x \in \mathbb{N} | x \leq s\} \tag{4.1}$$

where s is the maximum number of strides in a given window (i.e., 8 on HMR and 10 on GazeCom), and n is the number of individual samples contained in the same window (200 for HMR and 250 for GazeCom).

We have opted for a 5-fold cross-validation training regimen for the architectures. Each one of them was trained using 70% of the data. We have reserved 10% of the data for validation and 20% for testing. No randomization or stratification was used for cross-validation, i.e., the models were trained with contiguous temporal chunks of data extracted from the $k-1$ folds of each dataset in lexicographic order, and then evaluated on the remaining fold not seen during training, moving the context window one sample at a time.

A multi-scale sliding window of about 1 s was used to feed the architectures, which corresponds to 250 samples on GazeCom and 200 samples on HMR. Overall, 2,957,080 context windows were considered on GazeCom and 1,069,027 on HMR. The direction and speed were calculated along different scales within a window. The hyperparameters were determined empirically through a grid search with a subset of the datasets. All deep neural architectures used a dropout rate of 25% and a kernel size of 5 for the convolutional layers.

All models were trained and evaluated on a desktop computer with an Intel i7-7700 CPU with 16GB of RAM and with an NVidia GeForce GTX 1070 GPU (8Gb VRAM), running Ubuntu 18.04. The neural net models were implemented in Python 3, using the Pytorch library (1.8.1). All data, models, and tools developed for this work are available at: `https://github.com/elmadjian/OEMC`.

4.2.2 Evaluation and results

To evaluate the proposed TCN model, we compare it against two other online architectures based on previous offline models in the literature, namely a CNN-BiLSTM and a CNN-LSTM network (Startsev *et al.*, 2019a). As for a state-of-the-art algorithmic baseline, we selected the IBD-T (Santini *et al.*, 2016), which is a Bayesian classifier for online eye movement classification.

We propose the following set of evaluation aspects in this comparison:

- to determine what is the optimal number of time steps of neural online models when considering information from the past to predict the next pattern. Therefore, we investigate the impact in the model classification of different extents of time steps per context window, namely, 1%, 10%, 25%, 50%, and 100% of the context window, always considering the last sample in the window as the anchor point.
- to explore whether delaying prediction could increase model performance. Considering that online classifiers are most beneficial for real-time interactive applications and that human response time to what is considered instantaneous has a significant latency (Miller, 1968), we propose to study a set of different delays in prediction within 20, 40, 60, and 80 ms. By delaying our prediction, in theory, the classifier could have access to more samples from a given pattern to make more qualified forecasting, which could lead to improved general performance.

- to investigate whether the online neural architectures can operate swiftly in commodity hardware. Deep neural models are generally known to have a high computational cost, thus if the evaluated models cannot make predictions within a certain time threshold, they might not handle the throughput of real-time interactive applications.
- to determine the overall performance of all models in terms of F1-score for both individual samples and eye movement events on both datasets (GazeCom and HMR).

Precision and recall values on the sample level were calculated based on the aggregated confusion matrix built after running each partial model against its corresponding test fold with contiguous eye movement data. To compute the event scores, we considered contiguous labels in both datasets as single events, and we defined a predicted event as the highest frequency class from a model output within the ground truth span (see Figure 4.9). This scoring criterion tends to be more forgiving than the intersection over union (IoU) (Hooge *et al.*, 2018), as it does not excessively penalize failures in contiguity for predicted events, and it is similar to the criterion established by Hoppe et al. (Hoppe and Bulling, 2016).

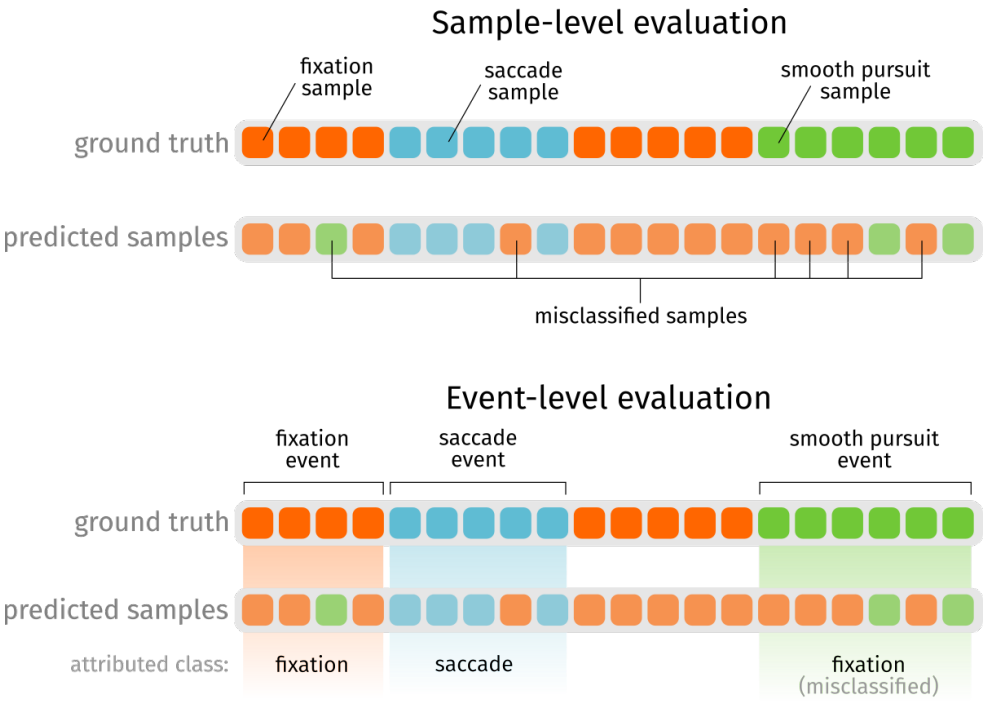

Figure 4.9: *Criteria for the sample and event-level evaluation. On the sample level, we compute the confusion matrix by comparing each individual sample predicted by the model on a continuous data stream against the ground truth. We define an event as the set of contiguous labels on ground truth and we say that the assigned event is defined by the highest frequency class among the predicted samples within a ground truth event.*

The aggregated F1-scores for all models with respect to both datasets are shown in Table 4.7. The F1-scores associated with each individual pattern can be seen in Table 4.8.

Based on the probability outputs for each class, we also built the ROC curves for all deep neural models for both datasets on the sample level, with the respective area under the curve (AUC) for each model, shown in Figure 4.10.

Model	F1 Sample			F1 Event		
	Prec	Rec	F1	Prec	Rec	F1
I-BDT*	77.96	78.23	75.69	75.06	58.87	55.51
CNN-BiLSTM*	81.03	81.05	80.50	90.44	90.32	90.11
CNN-LSTM*	80.64	80.65	80.15	89.99	89.82	89.65
TCN*	**85.79**	**86.40**	**85.31**	**93.25**	**93.18**	**92.74**
I-BDT+	74.28	73.97	71.22	77.43	70.22	68.49
CNN-BiLSTM+	84.46	84.45	83.73	87.82	86.79	86.47
CNN-LSTM+	84.85	84.94	84.26	87.99	87.00	86.70
TCN+	**85.89**	**86.14**	**85.51**	**88.45**	**87.66**	**87.39**

Table 4.7: *Models tagged with * indicate results for the GazeCom dataset whereas models tagged with + are associated to the HMR dataset. The highest values are highlighted in bold.*

Model	F1 Sample				F1 Event			
	Fixation	Saccade	Pursuit	Blink	Fixation	Saccade	Pursuit	Blink
I-BDT*	86.80	51.02	42.16	-	68.25	47.24	37.71	-
CNN-BiLSTM*	88.38	82.09	40.65	57.17	90.82	95.77	48.48	75.98
CNN-LSTM*	88.09	82.16	38.94	57.86	90.30	95.57	45.92	64.14
TCN*	**91.97**	**83.90**	**57.30**	**59.64**	**93.60**	**97.16**	**61.15**	**77.38**
I-BDT+	82.55	56.26	49.96	-	76.51	64.25	45.70	-
CNN-BiLSTM+	88.86	69.53	85.64	62.85	89.12	91.50	81.11	74.31
CNN-LSTM+	89.19	70.41	86.49	63.48	89.36	91.28	81.85	74.96
TCN+	**90.07**	**72.07**	**88.89**	**63.73**	**90.00**	**91.87**	**84.17**	**75.02**

Table 4.8: *Models tagged * are associated with the GazeCom dataset and the ones tagged with + to the HMR dataset. The Blink columns also include noise data. The highest values are highlighted in bold.*

The TCN model presents the highest scores in both aggregated and individual eye movement patterns, though the difference was more salient against other deep neural models on the Gazecom dataset, particularly when considering event-level scores. The gap between models was also more accentuated on underrepresented individual patterns in both datasets, that is, saccades on HMR and smooth pursuits on Gazecom.

For latency measurements, we simulated trained models from all three neural architectures against a continuous stream of out-of-sample data, considering only the required time for a model to make a prediction, that is, discarding the elicited time for preprocessing the features. Figure 4.11 shows the average prediction latency and the standard deviation for each model on GPU and CPU.

For real-time applications with more relaxed latency constraints, we assessed whether increasing the sizes of look-ahead windows could improve classification accuracy. For this reason, we also evaluated the models considering look-ahead windows of 0 (no delay), 20, 40, 60, and 80 ms. Figure 4.12 shows the results for both HMR and Gazecom datasets.

Finally, we investigated how advantageous would be to train models with few or larger

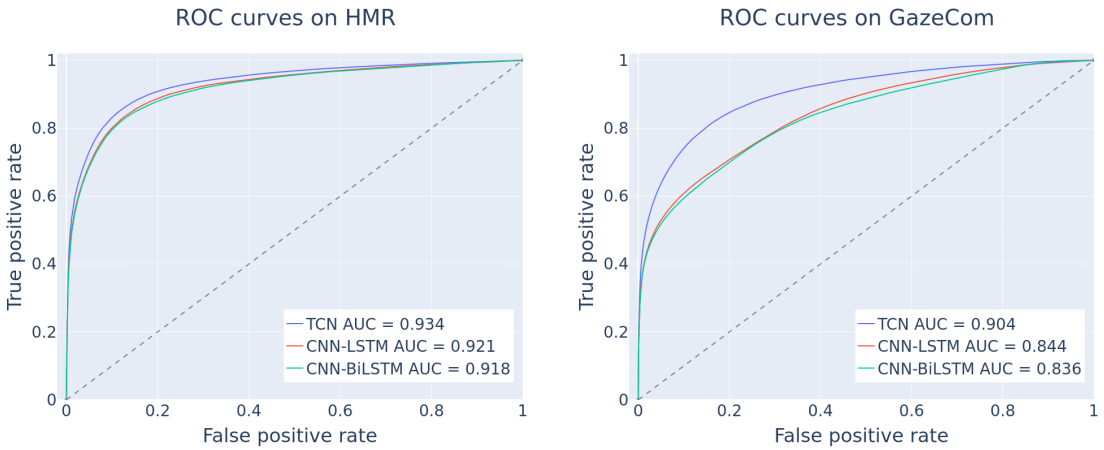

Figure 4.10: *ROC curves and AUC aggregated values using macro-averaging of all classes on sample level.*

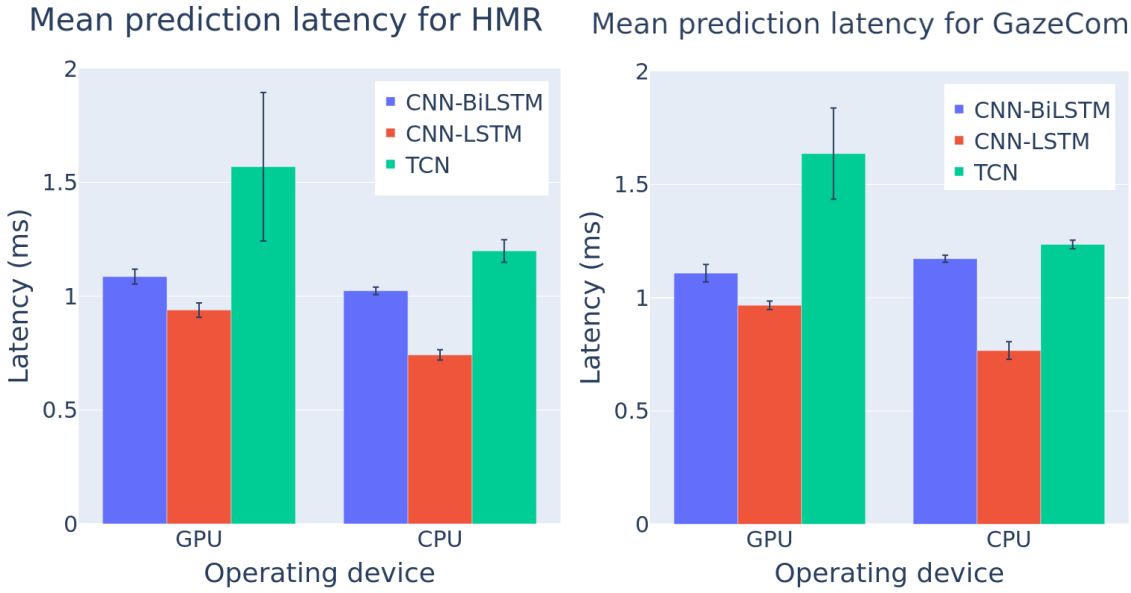

Figure 4.11: *Mean prediction latency using 100 ms of time steps for a single instance of the designed sliding window.*

numbers of time steps, as both TCNs and LSTM-based networks can learn long time-dependent relationships in the feature space. Our results indicate that all architectures peak their performance when training with approximately 100 ms of sequential past information (20 and 25 time steps, respectively, on HMR and GazeCom datasets). These results can be observed in Figure 4.13.

4.2.3 Discussion and conclusion

Our results indicate that the proposed TCN model consistently achieves the highest scores, not only in terms of general classification performance but for individual patterns as well. This outcome is in accordance with the previous offline investigation, in which we showed that a non-causal TCN architecture was able to surpass other deep neural models

Figure 4.12: *Performance of deep neural models along different look-ahead steps, ranging from a lag of 0 to 80 ms.*

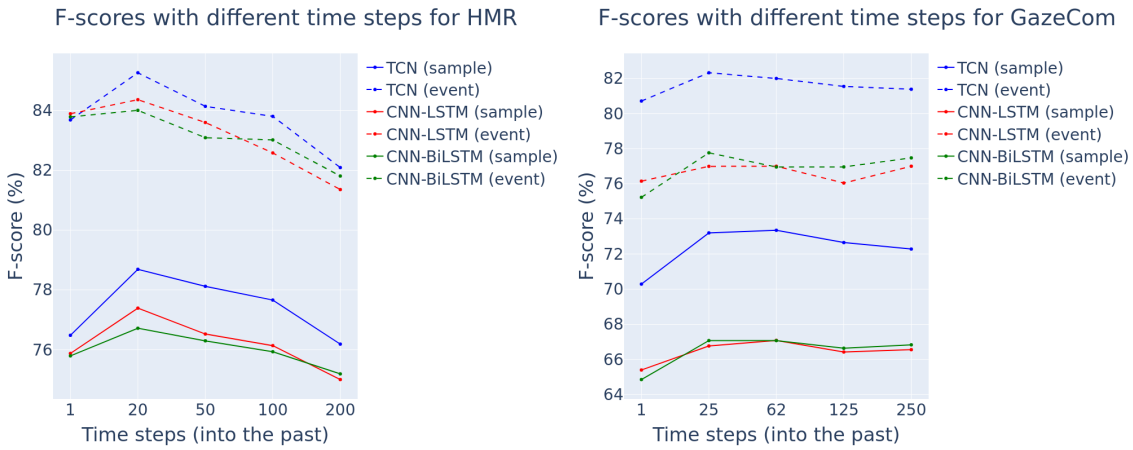

Figure 4.13: *Performance considering different time steps when looking into the past. Time steps are equivalent between HMR and GazeCom in terms of temporal span.*

in similar settings.

In general, all deep neural architectures scored higher than the I-BDT algorithm, our baseline model for 3EMCP online applications. This comparison with the baseline, though, has several caveats. To be fair to the limitations of the I-BDT, we trained and tested it only with individual users, and we excluded user videos where fixations, saccades, and smooth pursuits could not be found simultaneously, aside from blinks or noise, which were not fed to I-BDT.

Despite training the I-BDT using the window placement and size that maximized its performance, the deep networks were able to achieve not only higher accuracy but a greater generalization as well. The cross-validated test folds were intentionally filled with data from users that were not seen during training and validation in the case of the HMR, and aside from feature preprocessing, no other treatment, such as noise reduction, data alignment, or exclusion, was employed to improve model performance.

This level of generalization and accuracy comes with the cost of a more complex

training procedure compared to the I-BDT. This makes these models potentially unfit for scenarios where settings such as the eye tracker model or data stream frequency are constantly changing. On the other hand, if no significant changes are expected, this training happens only once, while the I-BDT always requires a per-user calibration.

The results using the HMR dataset show that the TCN F1-score is about 14.3% higher on sample level and 18.9% higher on event level than the scores from I-BDT, while on GazeCom these differences were 9.6% and 37.2%, respectively, resulting on an overall improvement of ~12% on sample level and 28% on event level. Considering the differences by class, the most noticeable one was with respect to smooth pursuits (sample: 27%; event: 31%).

When examining only the neural networks, the differences between the TCN and the other two CNN models were more evident with the GazeCom dataset, perhaps because it has a larger data variance compared to the HMR dataset. The margins in favor of the TCN might be an indication of its larger capacity, which is corroborated by the larger gap observed between the TCN and the second-best scores with respect to smooth pursuits, with a margin of 16.7% on sample level and 15.2% on event level on GazeCom. Overall, the average improvement of the TCN over the other neural models was about 3.0% and 1.7% on sample and event levels, respectively.

The results are generally consistent among all deep neural nets when comparing F1-sample and F1-event scores, but the same cannot be said about the I-BDT. While there was an increase from F1-sample to F1-event scores for the deep architectures (TCN: 4.7%, CNN-LSTM: 6.0%, CNN-BiLSTM: 6.2%), the I-BDT showed a performance drop when compared to its own F1-sample scores of 2.7% on HMR and of 20.2% on GazeCom.

One surprising finding was that the TCN was the slowest model in terms of prediction latency. Though all three neural architectures performed in the same order of magnitude, we expected the TCN to be more responsive due to its complete parallel structure when compared to the CNN-BiLSTM and CNN-LSTM, both of them with recurrent structures that have to be evaluated serially. There are a couple of explanations for that. First, our TCN implementation was not built completely on top of native optimized libraries from PyTorch, but the most likely reason is the fact that this TCN model is more complex, i.e., it has roughly 36,000 training parameters, while the others have about 20,000.

Overall, all deep architectures showed a prediction latency of at most 2 ms, within the expected throughput of typical commodity eye trackers, in particular the ones used to create both datasets (200 and 250 Hz), indicating that the trained models are indeed light enough to be deployed in real-time interactive applications. Another evidence that the adapted neural models are very lightweight is that there was no reduction in latency when using GPUs instead of CPUs for prediction, suggesting that the data transfer time dominated the process in the case of GPUs.

As for considering “look-ahead” buffers, that is, delaying the model prediction to improve classification accuracy, we noticed a perceptible gain. The results indicate that all architectures behave consistently, though distinctively, within each dataset. On GazeCom, larger thresholds seem more beneficial. Nonetheless, there is a clear trade-off in which the contribution of increasing the look-ahead window starts to fade away. Based on our

results, a reasonable look-ahead window, for all models, seems to be at 40 ms and at 60 ms for the HMR and GazeCom datasets, respectively.

Finally, in terms of time steps needed to build our feature tensor, our results indicate that roughly 100 ms is the ideal amount of sequential time steps from the past that need to be encoded for all models to achieve the highest scores in both HMR and GazeCom datasets. This goes contrary to the belief that more time steps lead to performance improvement, as is the case with offline classification. One explanation could be the increasing entropy between older time steps and the most recent sample in a context window, whereas in the offline problem, we can leak information from the future and traverse data in both ways to increase accuracy.

In summary, we proposed a novel preprocessing technique and adapted state-of-the-art deep neural models for the online classification of eye movement patterns. We showed, in particular, that the TCN architecture presents a larger capacity, achieving higher F1-scores than the I-BDT and the 1D-CNN-LSTM networks in the online 3EMCP. By modifying the deep neural models from *seq2seq* to *seq2one* architectures, training them with just a few time steps, and increasing the importance of more recent samples, we managed to achieve high throughput on sample prediction (approximately 500 Hz) using off-the-shelf hardware while maintaining high accuracy.

Our investigation also shows that, though it is possible to achieve reasonable accuracy levels with zero-length look-ahead buffers, the performance of all methods improves as we increase the amount of look-ahead information, which is particularly relevant during eye pattern transition. All methods have presented a 2 to 3% improvement in F1-score using a look-ahead window of 40 to 60 ms. For typical human-computer interaction applications that require response times under 100 ms, this increase might not represent a perceptual delay to the user.

4.3 Online reading detection

Since reading is an essential part of human activity and considered one of the most critical skills for which we have not been biologically programmed (Reichle *et al.*, 1998), it is reasonable to expect that many individuals will engage in reading tasks every day all the time, something for which a wearable system could display its potential benefits. For instance, if a wearable device could recognize these patterns and behaviors, it would be aware of the most appropriate moments to interrupt or shield the user against unwanted messages or provide better assistance in performing a search task that is context-relevant to the reading activity.

However, since wearables must stay on all the time and a typical head-mounted eye tracker works with at least one camera, it is imperative to devise a new way to reliably recognize eye behavior in a low-power scenario (Pouwelse *et al.*, 2001). Thus, in this investigation, we proposed a novel algorithm for reading recognition, which is arguably a more complex pattern comprised of saccades and fixations, that can operate at very low frame rates while still maintaining a comparable accuracy performance to the state-of-the-art. This investigation was first published in Elmadjian *et al.* (2016, 2017).

4.3.1 Materials and methods

In terms of eye movements, reading activity can be characterized as an interchange between saccades and small fixations with a 200-300 ms duration. The saccades are typically short in length and are predominantly observed from left to right along a text line. When the eyes change to the next line, they perform a larger saccade known as *regression*, which is a term also employed to designate rereading (Rayner *et al.*, 2001).

Other eye movements do not contribute significantly to reading (Lee, 1999). Thus, relying just on saccades and fixations, and assuming some ideal conditions like the reader's inertia and a fixed distance from the text, it is possible to devise a simple model for reading behavior. In fact, if the horizontal eye movements are plotted against time, one can observe a pattern known in the literature as the *staircase pattern* (Lee, 1999) (Figure 4.14). This behavior is also complemented by the eye movements in the y-axis, characterized by a gentle slope against time.

Previous reading detection algorithms have already taken advantage of this pattern (Buscher *et al.*, 2008; Campbell and Maglio, 2001a; Kollmorgen and Holmqvist, 2007). However, since all of these algorithms work with at least 50 to 60 images of the pupil per second and considering that a typical fixation lasts between 200 and 300 ms (Goldberg and Kotval, 1999), a very low sample rate scenario (e.g., 5 images per second) would make it impossible to differentiate saccades from fixations and it would be very difficult identifying outliers due to a miscalibration.

Thus, to correctly recognize reading in a low sampling setting and minimize unwanted effects, we propose a different approach. First, we model the reading pattern using a non-deterministic finite automaton (NFA) where the transitions between states are regulated by different thresholds (see Figure 4.15). These values, on the other hand, are calculated based on the eye movement information between two samples in the euclidean two-dimensional

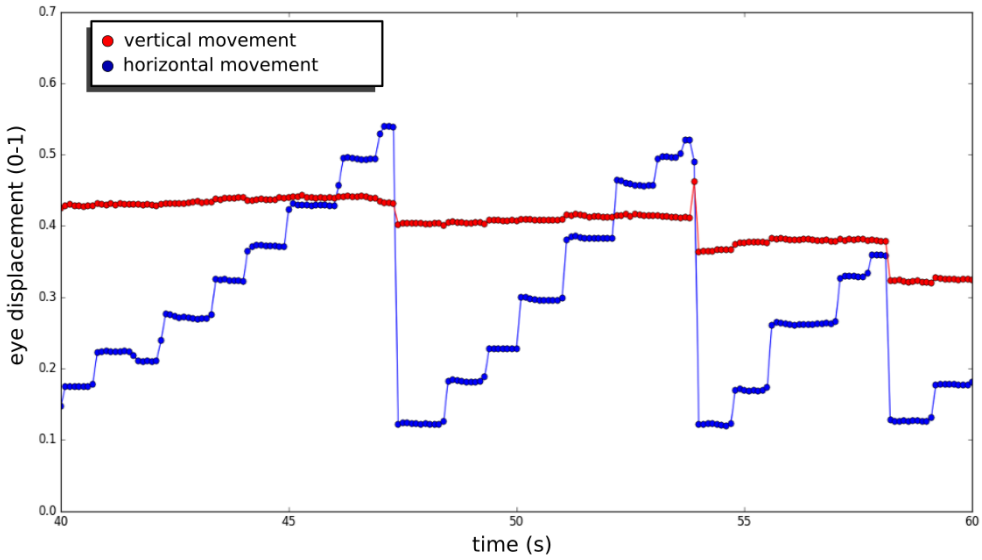

Figure 4.14: *The horizontal and vertical movement of the eye plotted against time. Horizontal samples portray a staircase pattern used in reading models. The vertical samples, on the other hand, show almost a gentle slope, with major displacements in regressions.*

normalized space. Using the slope determined by each pair of samples, the algorithm establishes the transitions to one of the three following states: long left saccades (LS), short right saccades (RS), or fixations (see lines 2-11 in Algorithm 1).

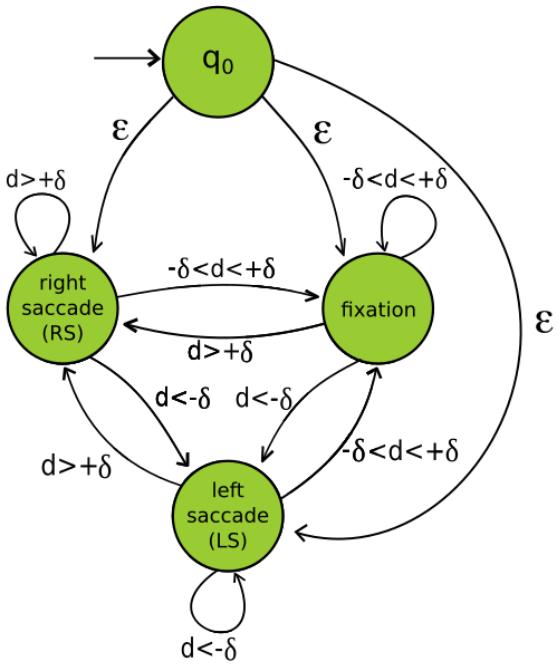

Figure 4.15: *The low-level NFA that is used by the proposed algorithm to identify fixations and the direction of saccades. The variable d indicates the slope calculated between two consecutive samples, whereas the value δ is an abstraction for the fitting criteria, which is shown in Algorithm 1.*

The final step is to build another layer of an NFA upon the previous one. Each state of this new automaton is composed of three consecutive saccades in the low-level NFA (see Figure 4.16). Other kinds of transitions are simply ignored. The remaining procedure is similar to others in the literature: every time there is a valid state transition in our NFA, points are assigned and if the accumulated score exceeds a certain threshold, reading detection is triggered.

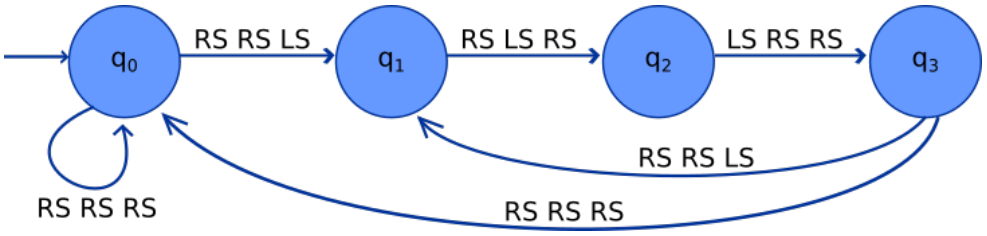

Figure 4.16: *The highl-level NFA that assigns a positive score the the reading_state variable in Algorithm 1 at each transition. All other possible transitions not mapped here are considered invalid. Every time a right saccade (RS) or a left one (LS) is detected, a queue of three slots named window is updated with this information, which triggers a call over this NFA to update reading_state.*

We call this method **IRDA** (Invariant Reading Detection Algorithm). The resulting pseudocode is shown in Algorithm 1. We use a queue named *window* to store the last three saccades, while the dx and dy variables keep the slope components in the current iteration for the x-axis and y-axis, respectively. There are also three constants: σ, which is used to differentiate small from big saccades, k, which is a parameter to determine the rate of change in the reading state, and *threshold*, the parameter to trigger reading recognition.

To assess the performance of IRDA, we designed an experiment in which 9 individuals — 5 male and 4 female — took part. Each participant was introduced to a random sequence of texts and images, for which three different tasks were required: if the content was just a text, the volunteers had to read it at their own pace; if it was a text but with a red countdown timer above it, the participants were encouraged to read it as fast as possible; finally, if an image was shown, the participant had to count a large number of objects on it.

All 15 texts used in the experiment had 120-170 words. They were all biographical excerpts from famous personalities. After each text, a multiple-choice questionnaire was applied just to assess if the content was actually being read. All 10 images showed a set of at least 30 repeated objects and were used to induce a non-reading eye movement behavior in the participant so that could account for false positives during analysis. Since we were also interested in determining whether the reading pattern was preserved with skimming behavior, some of the texts showed a countdown timer, and in this case, the participant was required to read them as fast as possible before the time ran out.

Eye data was collected at 30 Hz using the *Pupil Eye Tracker* and posteriorly subsampled to 10 Hz and 5 Hz. For each rate, there was a total of 135 samples — 45 of reading activity, 45 of skimming, and 45 of interaction with images. With this information in hand, we sought to evaluate the specificity and sensitivity of IRDA along with others with a similar score

Algorithm 1 Reading detection with a low-sampling rate

```
1:  while there is a next eye sample do
2:    dx ← point.x − previous_point.x
3:    dy ← point.y − previous_point.y
4:    if (dx > 0 and |dx| < σ) and dy ≈ 0 then
5:      push RS onto window
6:    else if (dx < 0 and |dx| > σ) and dy < 0 then
7:      push LS onto window
8:    else if dx → 0 and dy ≈ 0 then
9:      //fixation: do nothing
10:   else
11:     reading_state ← reading_state − k/3
12:   end if
13:   if is_valid(window) then
14:     reading_state ← reading_state + k
15:   end if
16:   if reading_state > threshold then
17:     //reading detected
18:   else
19:     //reading not detected
20:   end if
21: end while
```

system found in the literature, i.e., Campbell and Maglio (2001a) and Buscher *et al.* (2008). The best parameters for each one at 30 Hz were determined empirically and employed later for the lower rates (10 Hz and 5 Hz). To standardize the evaluation, we also used a normalized screen space, with values varying from 0 to 1 on the x-axis (left to right) and y-axis (bottom to top). Specifically for our algorithm, the constants were assigned as follows: $\sigma = 0.15$, $k = 15$, and $threshold = 30$.

Power consumption analysis

In order to assess the power-saving gains of reducing image resolution and frame rate for a wearable setting, we also conducted another user study with ten volunteers (undergraduate and graduate students, one female), in which they wore a binocular head-mounted Pupil Labs eye tracker (Kassner *et al.*, 2014), seated in a fixed chair.

The task consisted of looking at 17 visual targets (concentric rings presented in random order) displayed on a 22" monitor, from which 9 targets were used for calibration and the remaining for error estimation (see Figure 4.17). Four 2D markers were placed at each corner of the monitor so the monitor surface could be reliably detected on the scene image. For each target we recorded 2 seconds of video, that roughly corresponds to 60 frames for the eye and scene cameras. The two eye cameras had a resolution of 480 lines and the scene camera of 720 lines.

Data were collected under two different conditions: the **baseline** and the **central** conditions. In the baseline, participants used a chin-rest to keep their heads steady at

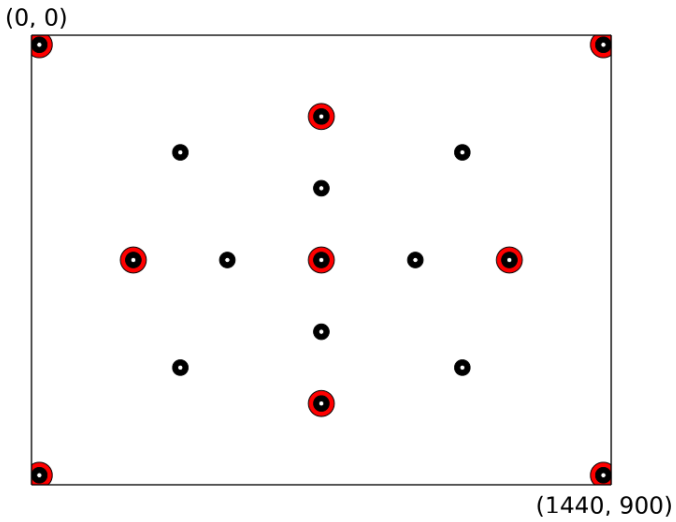

Figure 4.17: *Target disposition that was shown to participants during the data collection procedure.*

about 55 cm in front of the monitor. For the central condition, participants did not use the chin-rest and could perform natural head movements to look at the targets.

Recorded eye videos were down-sampled to 15, 10, and 5 frames per second (FPS), and the resolutions were reduced to 240, 120, and 60 lines. Then an offline adapted version of the Pupil Labs software was used to process each video combination of FPS × resolution. The scene video remained at 720 lines (its original value) for all conditions.

Power consumption was estimated indirectly by measuring the CPU time to perform the same task at different resolutions and frame rates, using the *psutil* Python library. During this estimation, only one of the processor cores (Intel Core-i7 3517U, 1.90 GHz) was activated. The processor was configured to operate statically at its maximum frequency.

4.3.2 Evaluation and results

The results for the specificity of IRDA and its performance comparison in different sampling rates are summarized in Figure 4.18, whereas the results for sensitivity can be seen in Figure 4.19. The Kolmogorov-Smirnov test showed that the data follow a normal distribution with a p-value < 0.05. Since the population variance was unknown, we calculated the average of the two measures with a 95% confidence interval using a Student's t-distribution.

Our algorithm showed a hit rate superior to 90% in all sampling rates, indicating its capacity of maintaining the same level of performance without changing the input parameters as the rates decrease. A similar scenario was observed for the false negative data, with a stable performance of around 80% in all the rates, while the other baseline techniques displayed a significant decrease in the true positive rate at lower FPS. The balanced accuracy for all algorithms is shown in Table 4.9.

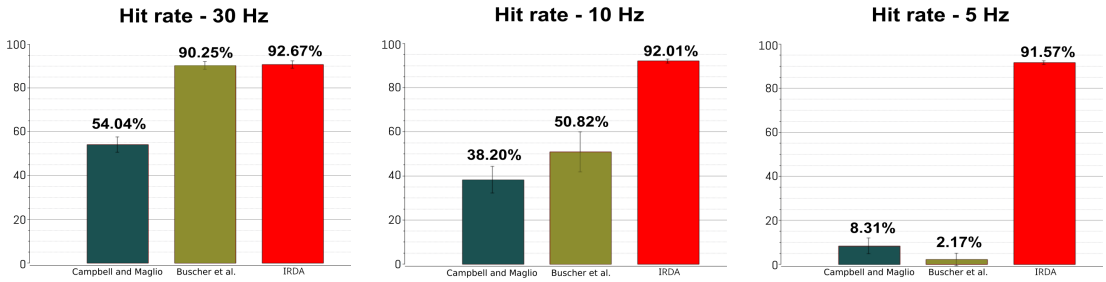

Figure 4.18: *Results for sensitivity (true positive rate) with sampling rates of 30 Hz, 10 Hz, and 5 Hz.*

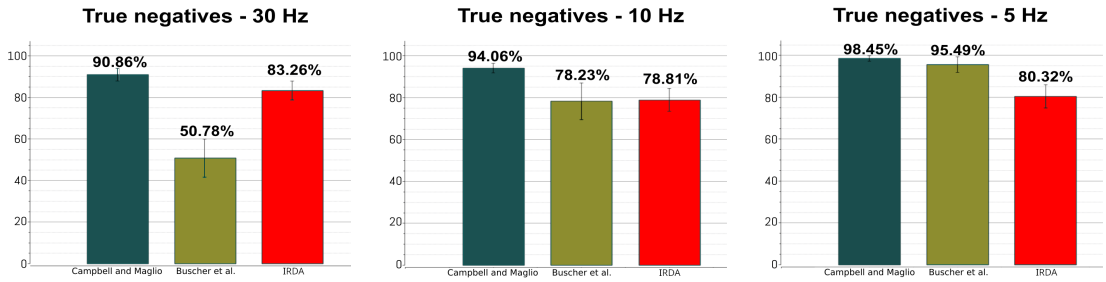

Figure 4.19: *Results for specificity (true negative rate) with sampling rates of 30 Hz, 10 Hz, and 5 Hz.*

	Frame Rate		
Method	**30 Hz**	**10 Hz**	**5 Hz**
Campbell and Maglio	72.45%	66.13%	53.38%
Buscher et al.	70.52%	64.53%	48.83%
IRDA	87.97%	85.41%	85.95%

Table 4.9: *Balanced accuracy of all evaluated algorithms for reading detection at different frame rates.*

Though no volunteer showed a reading pattern when exposed to image samples, the same cannot be said for the skimming samples. Thus, we decided to discard the skimming data of the analysis, as many of the participants dismissed the protocol when prompted for a skimming response, with the majority of them performing a reading activity at a normal speed. Also, it was not possible to accurately predict reading behavior below 5 Hz with any of the investigated techniques, so we set 5 Hz as the bottom value in the analysis.

As for the low-power investigation, results show that accuracy and precision in gaze estimation are not affected by using an eye image with 240 lines instead of 480 lines. In fact, with 240 lines, the gaze estimation error remains below 1^o of visual angle, as it can be observed in Table 4.10. Another interesting result is that accuracy and precision are not affected when the FPS is reduced from its original value of 30 Hz to 15, 10, and 5 Hz. For video resolution of 120 and 60 lines, the mean error in gaze estimation was between 8^o and 25^o for all FPS. This error is likely due to the fact that at such low resolution, pupil

contours are hard to detect. Hence, we did not include those results given the larger error and standard deviation.

	Center		Baseline	
FPS/Lines	**480**	**240**	**480**	**240**
30	0.75±0.45	0.78±0.49	0.83±0.7	0.81±0.55
15	0.76±0.46	0.77±0.47	0.84±0.75	0.81±0.55
10	0.75±0.44	0.77±0.47	0.82±0.64	0.80±0.54
5	0.75±0.44	0.77±0.50	0.86±0.83	0.79±0.55

Table 4.10: *Gaze estimation error for the center and baseline conditions.*

As for the CPU time, the Pupil algorithm was evaluated on a set of videos with the same content (eye moving to 17 different positions) but with different image resolutions (480 and 240) and FPS (30, 15, 10, and 5). For the 240 lines resolution, we found that, compared to the time spent in the native configuration of 480 lines at 30 FPS, the CPU time was reduced by 62% at 5 FPS and 480 lines, and about 91% at 5 FPS and 240 lines. Figure 4.20 exhibits the overall results considering all resolution and frame rate combinations.

4.3.3 Discussion and conclusion

In comparison to IRDA, the other two state-of-the-art methods for reading detection were clearly affected by lower rates. In particular, the algorithm from Buscher et al. (Buscher *et al.*, 2008) was the most affected in sensitivity, as it needs to cluster samples for fixations before trying to detect saccades. As for specificity, the other algorithms seemed to improve their performance with lower frequencies, but since this was accompanied by a severe decrease in sensitivity, this has to be interpreted as an overall loss of classifying performance, as it can be observed in Table 4.9.

Despite achieving superior levels of balanced accuracy in contrast with the other methods, it is not possible to state that our algorithm is better suited for reading detection in general, as the other techniques were tailored to work at higher frequencies, such as 50 and 60 Hz, and our dataset was recorded at 30 Hz, so we could not test the baseline methods in their original configuration. Nevertheless, our algorithm showed a substantial consistency of performance even in the lowest resolution, which was our primary goal.

Another important remark is that our reading detection model uses differential behavior observed in samples instead of raw gaze points. This is a fundamental difference between our method and the ones from Buscher et al. and from Campbell and Maglio because it makes our algorithm much more robust to issues like aliasing and the increasing weight of outliers in lower frequencies.

As for the data collection process, it should be said that a non-negligible amount of users during the task of counting objects on images actually displayed a temporary reading behavior with their eyes, which means that a specificity of 100% is, in practice, not

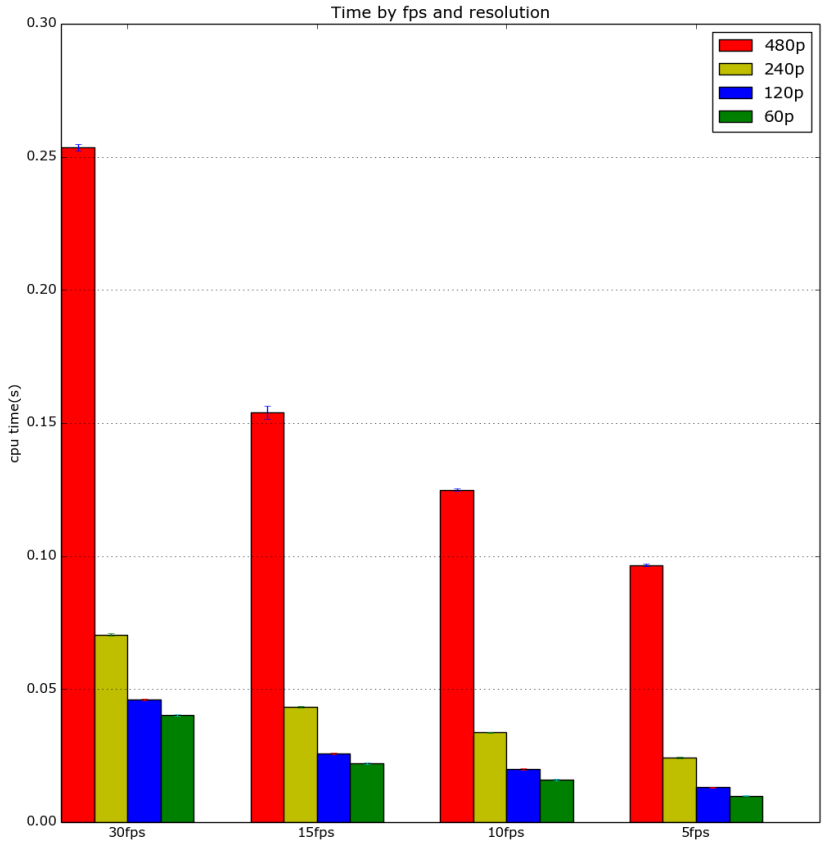

Figure 4.20: *Measured average CPU time to process a single frame of eye footage using the Pupil algorithm for different resolutions and frame rates. This evaluation was performed on a single core of an Intel i7 3517U processor.*

achievable with this dataset, although the precise figures for maximum specificity possible were not determined for this study.

Other than that, our brief exploration of gaze estimation accuracy and power consumption in constrained conditions showed that it is possible to save a considerable amount of power (more than 90%) operating at lower resolutions (i.e. 240p) and frame rates (5 Hz) without harming the performance of algorithms such as our reading detection model. This level of savings is very relevant in the context of wearable computing.

Overall, we sought to demonstrate that it is possible to devise solutions to critical limitations in wearables, such as energy consumption. We dealt, in this case, with the reading problem, developing the only algorithm — to the best of our knowledge — that is capable of reliably recognizing reading in low-sampling scenarios, such as 5 Hz.

Chapter 5

Gaze Estimation

Gaze estimation is a challenging problem, particularly in the context of wearable computing. Unlike traditional setups, wearable devices are designed to be mobile, leading to constant user motion and a lack of a fixed point of reference. Additionally, the use of wearable devices outdoors introduces further challenges such as variations in ambient light and interference from infrared radiation, which can render many eye-tracking techniques ineffective. Moreover, interactions with objects at varying depths in the user's field of view introduce parallax effects that can further compromise the accuracy of gaze estimation.

In this chapter, we will address some of these issues by investigating volumetric gaze estimation as an alternative solution in the context of wearable computing. This effort goes in line with one of our goals stated previously of enhancing 3-dimensional gaze estimation techniques to handle the problem of interacting with objects at different depths.

Among our contributions, we propose novel estimation techniques that leverage the depth sensing capabilities of 3D RGB-D cameras to estimate a 3D point of regard in the camera frustum space using only a single camera setup for each eye. While the results for the depth estimation component are still not as accurate as the directional one, we have achieved promising improvements in 3D gaze estimation. Additionally, we introduce a new dataset containing volumetric data collected from multiple users, which is publicly available for further research at `https://github.com/elmadjian2/3D_gaze_dataset`. A complete description of the work was first published in Elmadjian *et al.* (2018).

5.1 Investigation on volumetric gaze estimation

Most applications involving gaze-based interaction are supported by estimation techniques that find a mapping between gaze data and corresponding targets on a 2D surface. However, in Virtual (VR) and Augmented Reality (AR) environments, interaction occurs mostly in a volumetric space, which poses a challenge to such techniques. With the depth dimension, problems such as collinearity become apparent.

Accurate point-of-regard (PoR) estimation, in particular, is of great importance to AR applications, since virtual objects have to be perfectly aligned with the real world, and

most known setups are prone to parallax error and target ambiguity. In this work, we expose the limitations of widely used techniques for PoR estimation in 3D and propose a new calibration procedure using a non-calibrated head-mounted binocular eye tracker coupled with an RGB-D camera to track 3D gaze within the scene volume (i.e., the scene camera frustum).

The few attempts available in the literature to solve this problem impose several constraints. In the case of remote gaze tracking, this is a more straight-forward task, since both eyeballs can be determined on camera space, as well as the κ_θ and κ_ϕ associated with the angular difference between optical and visual axes (Guestrin and Eizenman, 2006; Hennessey and Lawrence, 2009). In the case of head-mounted eye trackers, the works that explored this problem presented either an already calibrated hardware (Lidegaard *et al.*, 2014) or a very constrained calibration procedure (Abbott and Faisal, 2012). Ultimately, none of these techniques were applied to the camera scene volume.

5.1.1 Materials and methods

In this investigation, we are concerned with gaze estimation for non-calibrated binocular head-mounted eye trackers in the scene volume with very few constraints. Therefore, in our calibration procedure, we do not require the user to be completely still, neither we have information about the coordinate system of eye cameras and the position of both eyeballs. Using an RGB-D camera attached to the user's head that generates a point cloud of the scene environment, we conducted a pilot study with 11 participants to investigate the feasibility of a calibration procedure for the camera frustum in such conditions. We will refer to *scene volume* or *camera frustum* instead of *3D estimation* to stress the difference between our approach and 2D surface calibration procedures applied to 3D environments.

With geometric models, one can assume that fixation in space occurs when there is an intersection between both vectors aligned with the visual axis. However, since vectors in 3D rarely intersect, a reasonable substitute for this requirement could be the midpoint of the shortest segment separating these two lines (Hennessey and Lawrence, 2009), as shown in Figure 5.1. Still, the absence of high-resolution cameras and simplifications in the 3D model of the eyeball may account for errors that compromise in a significant way gaze depth estimation, as shown in Figure 5.2. In this study, we investigate how well a simplified geometric model can be calibrated to the scene camera frustum in contrast with an appearance-based approach, given a set of targets in space covering the user's field-of-view (FoV).

Architecture description

A binocular head-mounted eye tracker from Pupil Labs was used to collect gaze data at 30 Hz with a resolution of 480p. The Intel Realsense R200 RGB-D camera was adapted to the tracker frame as the scene camera. The R200 device was configured to run at 60 Hz and capture one RGB image and one depth image from the scene at each frame. All these devices were connected to a laptop PC to process and record the streams. The head-mounted setup is shown in Figure 5.3.

The software used to compute eye features, such as the projected pupil centers and

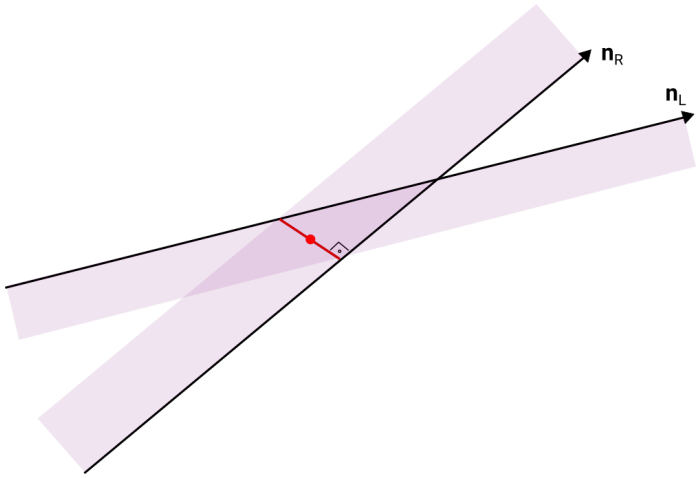

Figure 5.1: *Since two estimated vectors in 3D space coming from the right (n_R) and left (n_L) eyes will most likely not intersect, the midpoint of the shortest segment between gaze rays is a common measure of gaze estimation for geometric-based models.*

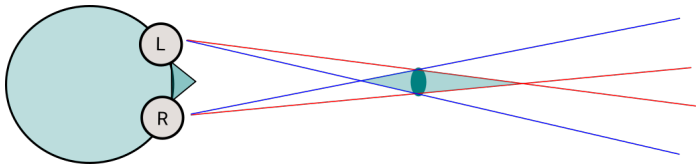

Figure 5.2: *Top-down representation of the effect of angular error (green ellipse) in gaze direction estimation: while it barely affects positioning in the plane facing the user, it has a significant impact regarding depth error, as it can be seen by the largest axis of the diamond-shaped area.*

Figure 5.3: *The Pupil binocular eye tracker coupled with an Intel Realsense R200 camera is used as a head-mounted setup.*

3D gaze vectors, was a modified version of the one provided by Pupil Labs (v0913) that also allowed us to record eye streams. We developed our own software, using *OpenCV* and *librealsense* libraries, to detect markers in the frustum, identify them, and report information about their position in 3D space. The technique used for marker detection resorted to a similar approach proposed by (Fiala, 2005). Once detected, the coordinates at the center of the marker were back-projected to 3D through a mapping procedure between the RGB and depth frames. A mean filter was applied over a window of 5 frames to smooth depth noise.

A portable smart projector was attached to a tall tripod to show the markers on a wall during the calibration procedure. A core routine was written to administer data acquisition from the eye tracker and scene camera and dispatch commands to a program running on the projector that controlled information being displayed on the wall, such as the marker to be fixated.

Calibration procedure

We designed our calibration procedure to be conservative about the number of targets that should be employed in the process, as there was no previous information about how the camera frustum should be spatially sampled. Therefore, we decided to use five planes at different depths from the user. Each plane had a grid of 5×4 binary AR markers for training purposes and a 4×3 internal grid used for testing. The relative position of the grids is shown in Figure 5.4. The size of the planes and their markers also changed with respect to depth to fill up the scene camera FoV, maintaining the same angular ratio between them.

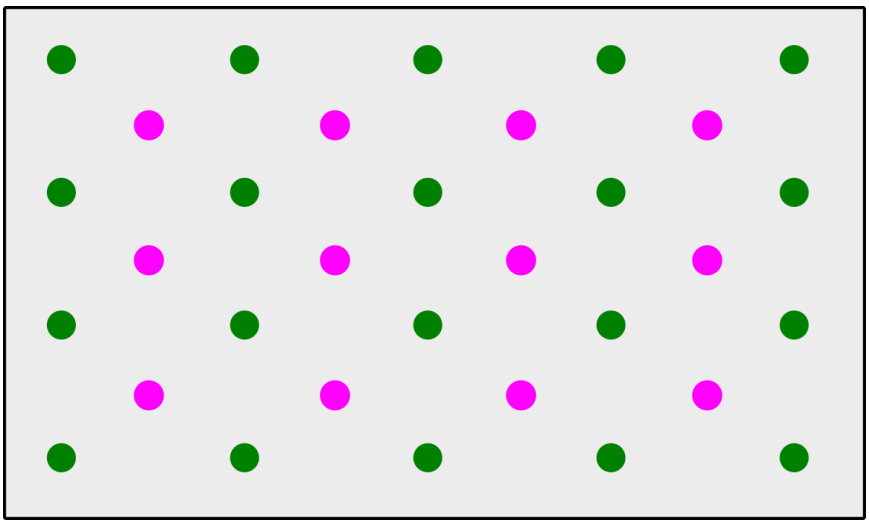

Figure 5.4: *Disposition of training (green) and testing (pink) targets per plane during the calibration procedure.*

The planes were defined by positioning the participant at 5 different depths from a projection wall: 0.75 m, 1.25 m, 1.75 m, 2.25 m, and 2.75 m. These distances were considered taking into account some limitations associated with the R200 camera, as it senses depth reliably only from 0.7 m to 3 m.

The procedure started by placing the participant 2.75 m from the wall and adjusting the projection center so it could be aligned with the camera scene FoV center. After that, we showed all the training markers on the wall and the volunteer was asked to follow a green target that remained static in the center of each marker at a time, while the system gathered 30 synchronized samples from each different camera feed. Samples were only recorded if eye features and markers were properly recognized. During this stage, participants got a chance to practice and were instructed to move their eyes to the correct target prior to triggering the recording.

Once the acquisition of training targets for a plane was complete, the participant was asked to remain still and repeat the same procedure now for the testing targets, which were shown in a similar fashion. Following that, we moved to the next depth plane by placing the user closer to the projection wall and we repeated the previous routine of centering the screen and showing the markers for data acquisition. An opportunity to redo part of the procedure was offered whenever the individual noticed a mistake. Figure 5.5 shows a diagram that summarizes the whole method, while Figure 5.6 depicts it.

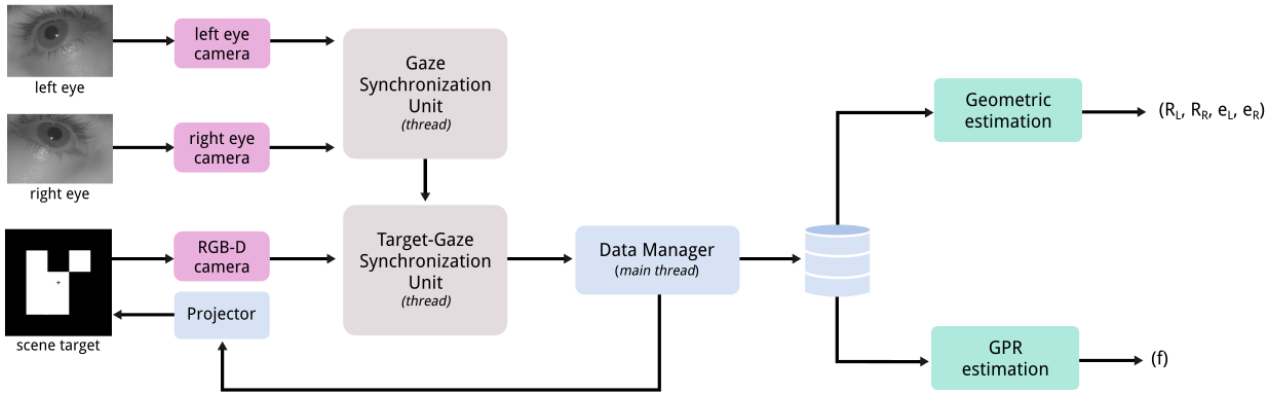

Figure 5.5: *Architecture of the calibration procedure. A Data Manager routine controls the experiment, requesting the projector to show training or testing targets and recording synchronized data from scene and eye cameras. Recorded data is used later for gaze estimation algorithms.*

Estimation procedures

Our geometric model is constituted by two normal vectors to the pupil centers provided by the Pupil tracking software. These vectors are a result of a 3D eyeball model that is built from multiple observations of projected pupil contours, which are approximated to ellipses. Assuming a camera pinhole model and a weak perspective projection, the centers of these ellipses are considered to be part of a sphere surface, which is regarded as a rough approximation of the eyeball. A detailed explanation of this method can be found in Świrski and Dodgson (2013).

To estimate the viewer's gaze depth, it is assumed that is necessary to find the convergence point of both gaze rays. However, this can only be achieved by determining the origin of the rays, i.e., the position of both eyeballs. This is not a particular challenge for remote eye trackers or even for some calibrated head-mounted devices, but it is still a challenge with a non-calibrated setup.

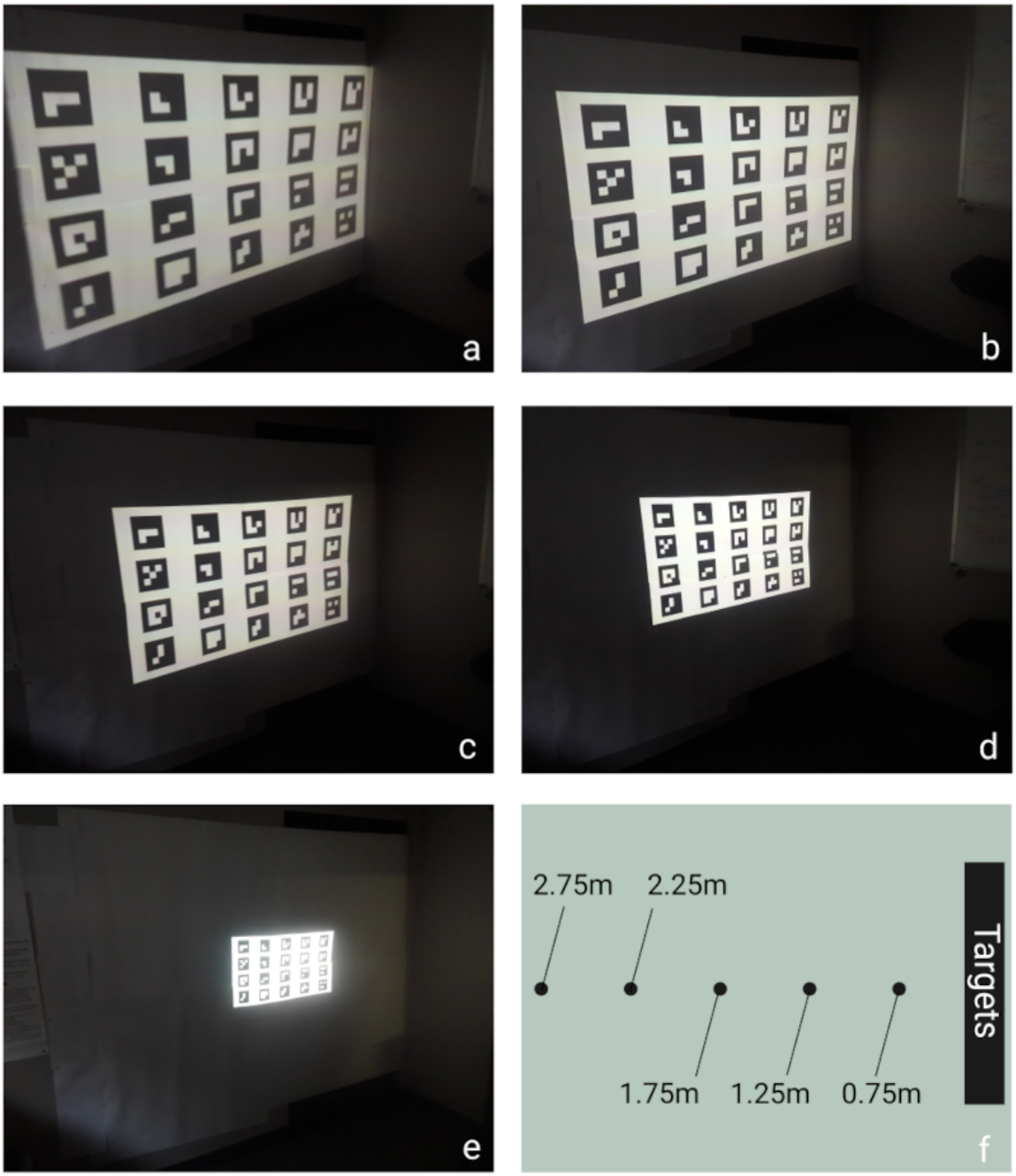

Figure 5.6: *Projection of the targets used in calibration with respect to depth, from 2.75 m to 0.75 m (Figs. a-e). Figure f illustrates the 5 different user positionings in this setup.*

With no constraints and prior knowledge about the user anatomy, finding the eyeball position in the scene camera space can be treated as an optimization problem. To reduce the complexity of the search space, we propose an approach that breaks the optimization procedure into two steps: first we compute the eyeball position and then we determine the rotation that places its gaze vectors into the scene camera space. This computation assumes that angular disparity patterns are roughly preserved among gaze vectors between camera spaces. Therefore, given a set of angles between gaze vectors in the eye camera space and a set of targets sampled in the scene camera frustum, the eyeball position is determined by minimizing the squared disparities between associated angles in both coordinate systems, as shown in Equation 5.1.

$$F(e) = \sum_{i=1}^{N-1} \left| (n_i \cdot n_{i+1}) - \left(\frac{t_i - e}{||t_i - e||} \cdot \frac{t_{i+1} - e}{||t_{i+1} - e||} \right) \right| \tag{5.1}$$

The idea here is to minimize a function F, where e represents the eyeball position, t_i stands for gaze targets, and n_i for the corresponding gaze vectors. A reasonable initial value for e is $(0, 0, 0)$. Although this procedure is also non-convex, the search space is comparatively reduced. Also, as the number of disparities involved in the search increases, the local minimum also decreases, making convergence faster and closer to ground truth.

Once a reasonable estimate is determined for both eyeballs, we proceed to compute the rotation that transforms the gaze vector in eye camera space to the appropriate orientation and position in the scene camera coordinate system. This is summarized by the parametric equation (5.2), where λ is a free parameter. Again, this is an optimization problem where we want to compute the rotation matrix (R) and the translation (T) that minimize the cosine distance between transformed gaze vectors and corresponding normalized vectors with origin at the 3D eye center and pointing towards the target. This can be expressed by (5.3). During minimization iterations, the parameter β can be used to penalize larger dissimilarities, speeding up convergence, if achievable.

$$e_{cam} + T + \lambda R\mathbf{n} \tag{5.2}$$

$$f(R) = \sum_{i=1}^{N} \left(1 - Rn_i \cdot \frac{t_i - e}{||t_i - e||} \right)^{\beta} \tag{5.3}$$

Finally, the PoR in 3D is computed as the midpoint of the shortest segment between both rotated gaze rays n_l and n_r, with respective origins in e_l and e_r. Assuming that this segment (r) is perpendicular to both rays and given the parametric equations of each ray, we solve for λ_l and λ_r to determine the midpoint m of this shortest segment, as shown in Equations 5.4, 5.5, 5.6, and 5.7. A diagram illustrating the geometric estimation pipeline is shown in Figure 5.7.

$$r = e_l - e_r \tag{5.4}$$

$$\lambda_l = \frac{(n_l \cdot n_r)(n_r \cdot r) - (n_l \cdot r)(n_r \cdot n_r)}{(n_l \cdot n_l)(n_r \cdot n_r) - (n_l \cdot n_r)(n_l \cdot n_r)} \tag{5.5}$$

$$\lambda_r = \frac{(n_l \cdot n_l)(n_r \cdot r) - (n_l \cdot r)(n_l \cdot n_r)}{(n_l \cdot n_l)(n_r \cdot n_r) - (n_l \cdot n_r)(n_l \cdot n_r)} \tag{5.6}$$

$$m = \frac{(e_l + \lambda_l n_l + \lambda_r n_r + e_r)}{2} \tag{5.7}$$

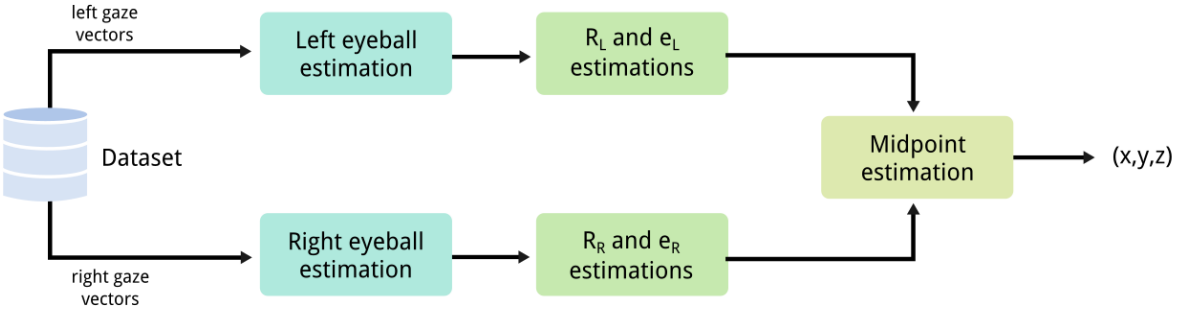

Figure 5.7: *Pipeline of the geometric method.*

As for the appearance-based model, we opted for a Gaussian processes regression due to its flexibility and reportedly good results from similar problems (Sesma-Sanchez *et al.*, 2016; Sugano *et al.*, 2013).

For the purpose of finding a regression between gaze data and scene targets, we selected a Squared Exponential Kernel, which is shown in Equation 5.8, with initial parameters $\sigma = 1.5$ and scale factor $l = 1.0$, as this configuration has demonstrated better generalization properties during our preliminary trials.

$$K(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right) \tag{5.8}$$

As with the geometric approach, the most difficult feature to learn is arguably eye vergence. Some authors have proposed to model this movement as a logarithmic function of the interpupillary distance (IPD) (Kwon *et al.*, 2006). Although there is some truth to that, in practice IPD is only markedly noticeable on camera when the targeted object is very close to the user. After 1.0 m, this measurement starts to be seriously affected by lack of resolution and noise in the sensor, as changes in IPD become small. Furthermore, IPD might not be constant in regard to depth, especially for fixation points situated obliquely to the viewer's center of view. IPD cannot be used with a non-calibrated hardware setup in which both eye cameras do not share the same coordinate system.

Considering these limitations, we approached the problem of depth regression separately, i.e., building a Gaussian processes regressor for gaze depth and another one for gaze direction in the projected scene camera plane. The results of both regressors were combined later to perform gaze estimation. A diagram illustrating the process is shown in Figure 5.8.

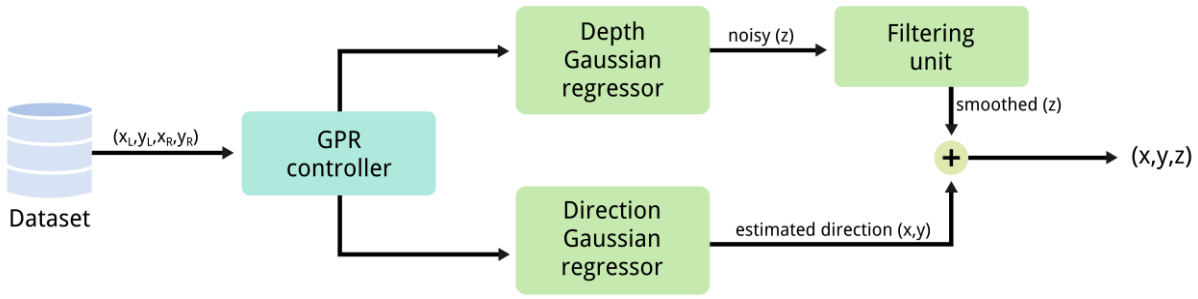

Figure 5.8: *Pipeline of the Gaussian processes regressor method.*

5.1.2 Data collection

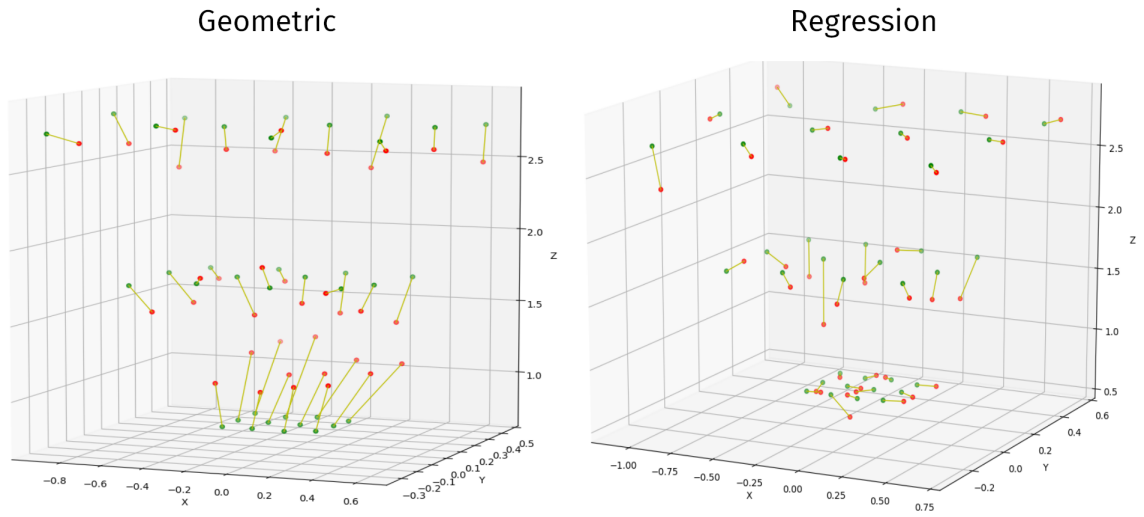

Figure 5.9: *Gaze estimation from a participant using the geometric approach (on the left) and the regression-based one (on the right). Green points indicate the ground truth, while red ones are the corresponding estimates.*

We collected gaze data from 11 subjects (5 women) with ages ranging between 22 to 35 years old. All of them had a normal or corrected-to-normal vision during the procedure, which followed the protocol that is fully described in 5.1.1.

The dataset comprises information about both left and right normalized pupil centers in each image (*LE2D*, *RE2D*), as well as both normalized gaze vectors acquired through 3D eyeball modeling (*LE3D*, *RE3D*), and the ground truth targets in 3D coordinates provided by the RGB-D camera (*RS3D*). During this process, we also acquired grayscale frames from both eyes and scene cameras for debugging purposes, although this information was not integrated into the dataset due to size limitations. Therefore, a valid sample was defined as:

$$S = \{RS3D, LE2D, RE2D, LE3D, RE3D\}$$

A total of 30 samples were used per target, which accounts for roughly 1 second of

observation with our current architecture. This value was chosen considering the subject's likely extenuation due to the large number of targets in the experiment. Additionally, to minimize individual errors and increase comfort during data acquisition, each participant received a device to activate the moment of recording each target being gazed at. Table 5.1 summarizes information about the number of samples for training and testing targets. Our dataset is publicly available at *https://github.com/elmadjian2/3D_gaze_dataset*.

Samples	Training	Testing
Per Target	30	30
Per Plane	600	360
Total	3000	1800

Table 5.1: *Summary of the number of training and testing targets collected per user.*

5.1.3 Evaluation and results

The precomputed gaze vectors from each eye were used separately as data input for the geometric model, while a four-dimensional vector containing the data from left and right projected pupils were assembled as input for the Gaussian processes regressor, a step necessary as depth can only be inferred through simultaneous information from both eyes.

In total, five planes were used for training of both methods, while two intermediate planes for testing were discarded, keeping only the closest (0.75 m), the farthest (2.75 m), and the middle one (1.75 m). The reason for that was to assess whether intermediate planes would increase or harm depth estimates, considering the results that have already been reported in another study by Lee *et al.* (2017).

We evaluated accuracy in terms of depth error, angular error, and Euclidean distance to ground truth. Figure 5.10 summarizes the basic statistical results with respect to each metric by plane using all 5 planes for prediction. These results are compiled in Table 5.2. Figure 5.11 portrays the impact of the number of training planes for gaze estimation. Since there was no significant difference in using 3 to 5 planes for training, we report only the analysis of variance (ANOVA) results for the latter.

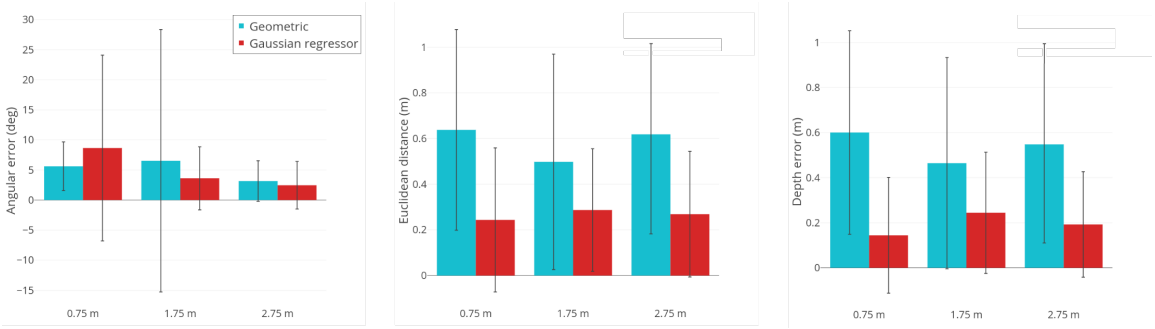

Figure 5.10: *Average angular error, Euclidean distance, and depth error for each testing plane separately.*

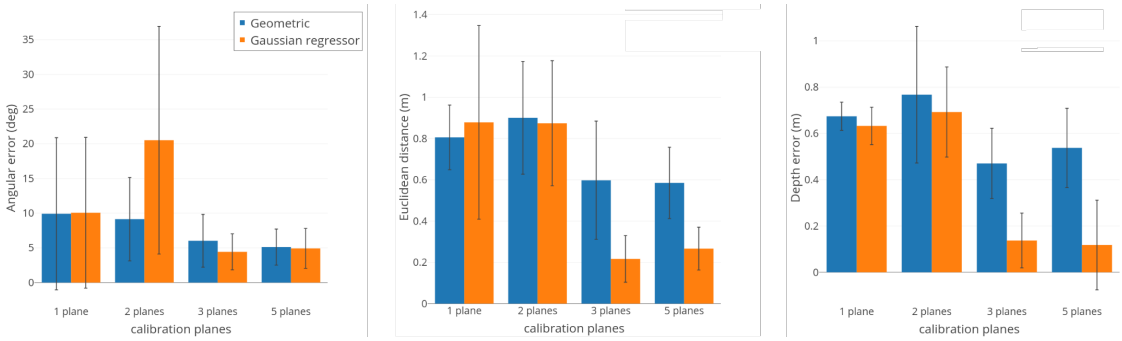

Figure 5.11: *Average angular error, Euclidean distance, and depth error with respect to the number of planes used for calibration.*

Metric	Geometric	GPR
Depth error (*m*)	0.538 ± 0.171	0.194 ± 0.118
Angular error (*deg*)	5.105 ± 2.594	4.911 ± 2.878
Euclidean distance (*m*)	0.585 ± 0.173	0.266 ± 0.103

Table 5.2: *Summary of the results for the three metrics considering all the testing planes.*

Regarding the depth estimate metric, a two-way repeated measures ANOVA showed that there was a main effect on the method ($F(1, 10) = 50.74$, $p < 0.001$), indicating that the regression approach was more accurate than the geometric one. However, there was no noticeable effect on the plane or interaction between the method and the plane.

Regarding the angular error, despite the regression-based technique being apparently more accurate, no significant effect was observed either on the method, given high standard deviation values. An effect on the plane ($F(2, 20) = 4.04$, $p < 0.05$) was observed, suggesting that angular accuracy is not homogeneous among distinct planes.

For the Euclidean distance metric, a significant effect was observed again on the method ($F(1, 10) = 65.31$, $p < 0.001$), suggesting once more the Gaussian processes regressor as a more accurate estimator. No significant effect was perceived on the plane, or between the method and the plane though.

5.1.4 Discussion and conclusion

To the best of our knowledge, this was the first work that attempted to perform this kind of gaze estimation for the purpose of establishing PoR in scene volume using a head-mounted eye tracker. During this process, we investigated and further developed two different estimation techniques for this problem: one geometric and one appearance-based.

Besides providing an entirely new dataset for volumetric gaze estimation, we also explored other minor problems, such as estimating the eyeball position through angular disparities between the different camera coordinate systems.

It is clear from our results that gaze estimation in a scene frustum is still an open challenge, especially in terms of gaze depth. A first observation from our results was

that the number of calibration planes has a noticeable impact on estimation accuracy. Therefore, even throughout a geometric approach, a single-plane calibration does not suffice to provide useful gaze estimation in the scene volume.

We also noticed higher angular errors at the closer plane (0.75 m), particularly with respect to the appearance-based method. This might contradict the expectation that the farthest plane (2.75 m) should yield the worst results. Yet, one possible explanation for this outcome could be the effect of angular disparity of vergence on closer planes.

Overall, it was possible to observe that by adding the depth dimension to the calibration problem, XY-plane estimates also tend to degenerate in both approaches, which was expected at some level, as depth sensing provided by the R200 camera had a considerable amount of noise.

A point that needs to be cleared is how contributive the adaptive response of the ocular-motor system is to input data quality. In other words, further investigation is necessary to find out whether the speed or stability of human focusing response is affecting estimation so that more samples should be acquired per target to assess appropriate gaze depth.

That said, both methods considered for this study presented different strengths and weaknesses. The geometric model showed a tendency of preserving the spatial relationship between testing targets, although inaccurate estimations of the rotation matrix and the eyeball position clearly compromised the whole system, as gaze estimates tended to appear shifted by a relatively constant degree from the user perspective. The appearance-based method does not suffer from this problem, but estimates seemed to be more random comparatively. Figure 5.8 illustrates both of these phenomena.

Although the Gaussian processes regression might have shown improved results in two metrics, it should be noted that it requires a large number of targets for training to provide a suitable regression, whereas the geometric model, at least in theory, requires fewer targets. This over-reliance on the number of samples by the appearance-based approach can easily be observed in Figure 5.11.

Finally, despite somewhat large errors, it is possible to devise some uses for the proposed techniques in AR scenarios, such as allowing the user to have different contextual interfaces based on gaze depth, or triggering access to detailed information about scene objects by looking at an HMD or through it. With improvement in estimation accuracy, vergence-based controls for 3D interaction could also be feasible.

Chapter 6

Gaze interaction

In this chapter, we present a diverse set of interactive techniques that we have investigated for the wearable scenario, as part of this thesis. Although some still need to be validated empirically with more thorough user studies, they represent a tentative and exploratory effort in the direction of a viable gaze-based interaction in the wearable domain.

While gaze-based input may not be as efficient as manual input (Majaranta and Räihä, 2002), it allows for privacy and hands-free interaction, besides being a natural indicator of attention (St. John *et al.*, 2004). Our position, nevertheless, is that, though possible, gaze-based interaction is not appropriate for every task, since the eyes are our primary source of information about the environment, and using the eyes as a primary means of interaction would heavily constrain the user's perception.

For some scenarios, however, the use of gaze would be not only feasible but might indeed improve the overall user experience. In the envisioned future of everything connected and "smart", micro-interactions abound (Ashbrook, 2010). Micro-interactions are about self-contained moments that revolve around one piece of functionality. Although small and quickly forgettable, they occur everywhere. From adjusting the room temperature or dimming the lights to turning appliances on and off or connecting devices, they are the glue that ties together different features in a myriad of devices and platforms (Saffer, 2013). We unconsciously perform micro-interactions all the time, but we rarely realize their presence, except when they fail.

As an effort to fulfill the last of our three goals stated in the introductory part of this thesis, this chapter reports some core ideas, design principles, proof of concepts, and early results of different techniques that leverage gaze in these brief interaction scenarios that are so pervasive in the wearable computing world.

Similarly to the chapter on eye pattern recognition, each section here describes a self-contained investigation, with locally independent goals and conclusions. In particular, we cover three different research projects: GIMIC, GazeBar, and V-Switch. Part of the contributions presented was first published in Elmadjian and Morimoto (2021).

6.1 Gaze-based micro-interactions with AR-enabled devices

With the combined emergence of Internet-of-Things (IoT) and Augmented Reality (AR) technologies, our everyday lives will be increasingly surrounded by smart devices, potentially collaborating with each other to create personal AR-enhanced ecosystems. Once these smart environments become available, how do we interact with them in an intuitive and efficient way? Currently, the most common approach is using a mobile device. This solution is commercially convenient in many ways: smartphones are widely available and they offer off-the-shelf hardware to support AR apps. But from the user's standpoint, this may not be the best choice. Many tasks associated with smart devices are very simple and brief, which means that the cognitive and temporal cost to start engaging with a smart appliance might be higher than the actual task.

In this work, we discuss some of the foreseeable interactive scenarios ahead, introducing what we call Gaze In Micro-interactions (GIMIC). The idea behind it is that we can leverage natural eye contact and brief gaze gestures for efficient short-term interactions.

6.1.1 GIMIC use cases and design principles

The kitchen is an interesting home scenario due to the high number of smart appliances that could become available. For example, once the user gets to the kitchen, all smart appliances such as the microwave, sink, tap, coffee maker, and toaster, just to name a few, might fight for the user's attention, creating confusion on the heads-up display (HUD) (see Figure 6.1. Observe that in this scenario, even when devices are placed side by side, their physical dimensions will probably create enough separation for a head-mounted gaze tracker to robustly determine which object is being gazed at.

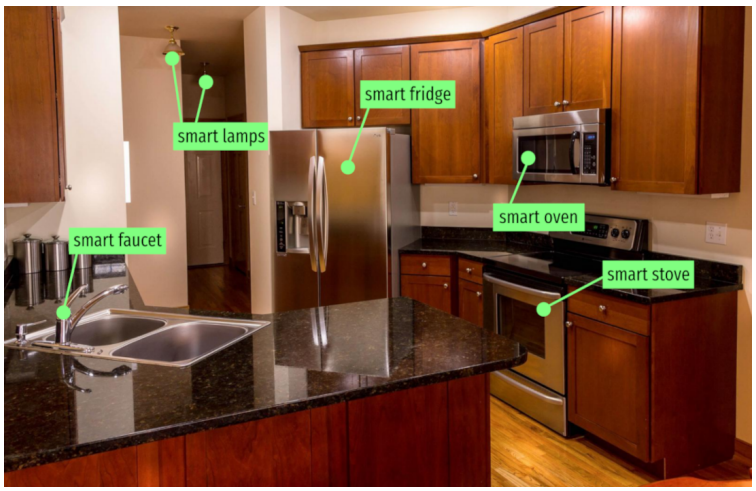

Figure 6.1: *The kitchen can be a stage for several smart appliances and use cases for GIMIC. There are many opportunities for brief tasks, such as dimming a specific lamp, checking for missing ingredients in the fridge, warming the milk in the oven, preheating the convectional oven, or changing the temperature of the tap water.*

Following natural conversational behavior when we talk to multiple people, we use

eye contact to start an interaction with a smart device. Before that, to remind the user of which objects in the scene are interactive, the HUD can visually enhance (or tag) the smart objects within the field of view of the wearable AR camera.

For example, by equipping food containers with RFID tags, the smart refrigerator 'knows' its contents. When the user makes eye contact with the refrigerator, its status and the list of its contents can be displayed on the HUD. Using an AR display one could create much more elaborate experiences, such as x-ray vision to see the contents of the refrigerator. We will keep the solutions simple so we can focus on the benefits of using GIMIC.

After the user finds all the ingredients, he can fill up a cup of milk and put the cup in the microwave. Setting up the microwave manually is simple enough, but using the wearable platform might be even simpler if the platform is aware of the context, i.e., that the user wants to warm a cup of milk, and knows the user's habits and preferences. Still, the user might want to adjust the temperature from warm to hot. By keeping eye contact with the microwave, the user can see its interface on the HUD, and select the desired temperature by gaze or speech.

Similar to social situations when people establish eye contact and wink or blink to send a quick message, once eye contact is established with a smart device, a quick eye gesture could be used for simple and short interactions. GIMIC's purpose is to speed up these brief communications without causing eyestrain, which is a common phenomenon when resorting to active gaze selection for long periods of time. Besides, since micro-interactions are performed over a limited set of options, interfaces become naturally less cluttered and can make better use of screen space, which makes the user less susceptible to eye-tracking issues related to accuracy and precision.

GIMIC can also be used as a first step in long interactions. For example, if the user wants to check the power consumption history of the microwave, the wearable interface could offer an option to open the appropriate app on the user's smartphone.

Design principles

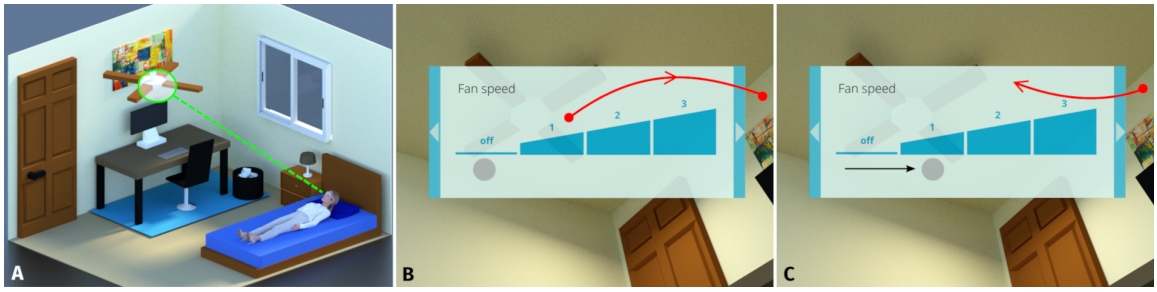

Figure 6.2: *An application of the GIMIC principle. (A) The user performs gaze contact with a smart fan through her AR-enhanced glasses. (B) Once engaged with the fan, the user starts an eye gesture to the right (red arrow) to change the fan speed. (C) After looking back at the UI to complete the gesture (red arrow), the user gets feedback indicating that the fan speed has changed.*

The following four items embody the guidelines to create interactive applications using GIMIC. Figure 6.2 illustrates these principles with a use case.

- *Simplicity*. GIMIC assumes uncluttered interfaces and easy commands. Gaze point estimation is not as accurate and precise as other traditional pointing mechanisms, so selectable targets on an interface should not be placed very close — which also limits the number of targets that can be shown. Besides, bloated feedback naturally degrades performance due to visual search. As far as commands go, GIMIC should use simple eye gestures because they are easy to learn and perform, as well as being more secure against the Midas touch problem (Jacob, 1991).

- *Brevity*. Because the eyes are primarily used as a source of visual information to us, we should not rely on gaze as a predominant means of interaction. Additionally, continuous active use of gaze can lead to discomfort and eyestrain. Therefore, GIMIC should only be applied to tasks that are naturally brief or perhaps too short and self-contained parts of a complex task. Requiring active use of gaze for long periods of time can be perceived as awkward and unnatural to users (Sibert and Jacob, 2000).

- *Effortlessness*. Engaging and disengaging with smart devices should be perceived as effortless. That is why we support the idea of using natural conversational behavior to toggle engagement. Because our attention is correlated to eye fixation, accessing a smart device menu this way would go along with the transparent computing paradigm (Zhang and Zhou, 2006). Also, gaze interaction often lacks precision, and therefore mistakes will happen, perhaps more frequently than with other modalities. Thus, the cost of correcting a mistake should be low. A simple directional gaze gesture takes only a fraction of a second, aside from being more robust to precision drifts.

- *Swiftness*. GIMIC should be fast. We argue that GIMIC can be particularly advantageous if it can help the user accomplish tasks that become more worthwhile if completed quickly. Therefore, the process of engaging with a smart device, making a selection, and then disengaging should not take more than a few seconds. That is why we claim that dwell-time techniques, though easy and comfortable, are unsatisfactory for brief tasks.

6.1.2 GIMIC prototype

Based on the aforementioned design guidelines, we assembled a wearable prototype, as shown in Figure 6.3. It is composed of a binocular Pupil Labs eye tracker and the Vuzix M100 Smart Glasses. The M100 has a monocular HUD and other inertial sensors, such as an accelerometer and a gyroscope, and can be mounted in front of the right or left eye to be used as a smartphone extension. In our prototype, we only set as active the eye camera located on the same side of the HUD.

The M100 is attached to the head-mounted eye tracker using a custom mount we designed, and manufactured with a 3D printer. This mount allows users to adjust the height of the HUD in front of their eyes. The head frame can be fastened to the user's head using an elastic band to avoid slippage during its use.

Figure 6.3: *Prototype showing the Vuzix M100 Smart Glasses attached to a Pupil Labs head-mounted eye tracker using our custom mount.*

Ideally, the smart glasses should be independent of other devices and run all the computations required by wearable applications, including heavy image processing tasks, such as AR scene understanding, pupil tracking, and gaze estimation. Due to hardware processing limitations, we have used a notebook as the host of our personal network where the M100 is connected through a TCP/IP wireless link, and the eye tracker through a USB port.

The software architecture is shown in Figure 6.4. There are 5 modules that run in the notebook: the **eye tracking software**, the **gaze data server**, the **WebSockets server**, the **gaze HUD**, and the **application**. There are two modules running on the M100: **video renderer** and **head gestures detection**.

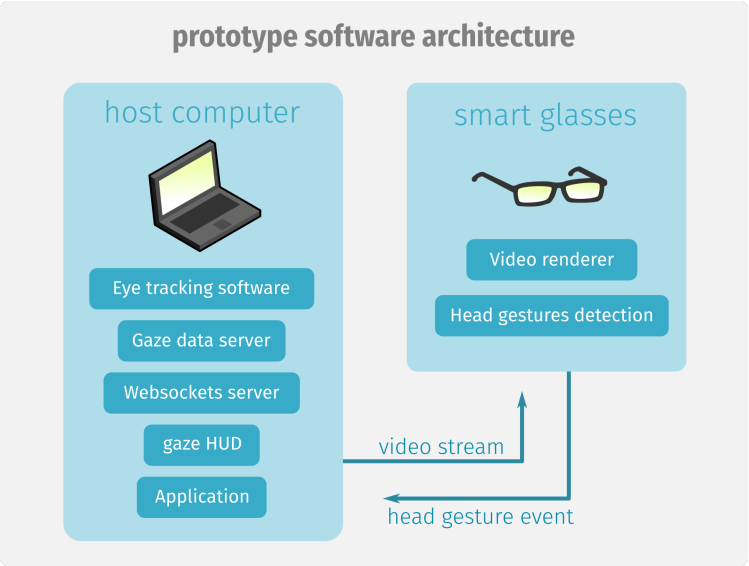

Figure 6.4: *Software architecture of the personal network. The left block contains the software components running on the notebook host and the right block shows the components running on the M100.*

The eye tracking software detects the pupil in the images of the Pupil eye camera and estimates the observed point in the images of the Pupil scene camera, using a transformation computed from a 9-point calibration procedure. A second calibration is required to estimate gaze positions on the HUD. Figure 6.5 shows the HUD calibration interface. Additionally, this module detects 4 simple eye gestures. Each gesture starts by looking at the center of the HUD and then looking up, down, left, or right, and returning the gaze to the center. Each gesture can be associated with a gaze-activated command (i.e., a micro-interaction).

After calibration, the system sends the gaze data streams over 0MQ sockets, an asynchronous messaging library. This data is read by the gaze data server and stored in a circular queue. Any wearable application registered in the personal network can read the streams from the sockets.

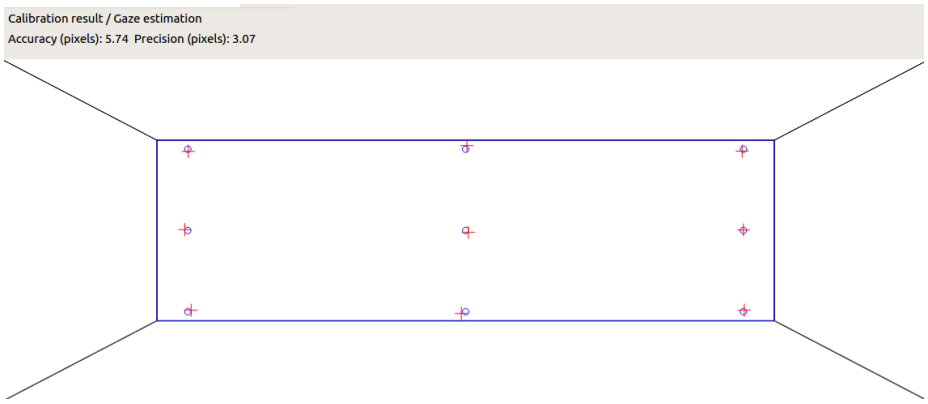

Figure 6.5: *The HUD gaze calibration interface. After a 9-point calibration on the HUD, the panel shows the calibration points (blue circles) and estimated gaze (red crosses), along with accuracy and precision in pixels. Trapezoidal areas around the HUD are the areas used to execute 2-step gaze gestures.*

The WebSockets server is a proxy between the application and the HUD. Applications can send a video stream to the HUD via a 0MQ socket. The WebSockets server receives this stream and sends it to the HUD via WebSockets. An HTML5 application running in the M100 reads the video stream from the WebSockets server and renders it on the HUD. The communication via the WebSockets server is bidirectional since the wearable applications running in the M100 can also detect head gestures using the embedded gyroscope and send them to the WebSockets server. The server then sends this information over 0MQ sockets to be read by any registered application in the personal network. Although we implemented a fully functional head gestures detection module to allow for multimodal interaction, we conducted our pilot study focusing only on gaze.

The HUD gaze shows an interface for training gesture execution. Because these gestures have no eccentric target to guide the saccade (differently from Ohno (1998) and Diaz-Tula and Morimoto (2015)), the user must look outside the HUD area.

The application module is the software that implements the interface between the user and a smart device. It receives input from the gaze data server, the gaze HUD, and the WebSockets server (for head gestures detection). The interface is shown on the HUD during the interaction.

The modules running on the smart glasses are implemented in HTML5 and have two main functionalities. The video renderer module receives a video stream from the host (via WebSockets server) and renders it on the HUD. The head gestures detection module detects four different head gestures (tilt to the left or right, and turn the head up and down) using the gyroscope embedded into the Vuzix. Detected gestures are sent to the WebSockets server, which in turn sends them via 0MQ sockets to the application.

6.1.3 Evaluation and results

We have conducted a pilot study to compare the performance of two interactive scenarios with smart devices: one with our wearable prototype using GIMIC versus a mobile device. We have chosen a mobile device as our comparative baseline since this is the most widely adopted platform to manage smart appliances.

Six volunteers participated in the pilot study (1 female), all able-bodied, with normal or corrected-to-normal vision. All participants reported being very comfortable with interactive techniques for mobile devices. Three of them said they had previous experience with eye-tracking devices.

The gaze-enhanced wearable platform prototype was used to control three different digitally rendered devices: a lamp, a fan, and a thermostat. Instead of using actual IoT appliances, we decided to use AR markers in their place to reduce complexity and increase environmental control (see Figure 6.8). This decision also provided us with means for video-based localization and engagement.

For each device, the interface of its wearable application was developed to allow simple commands such as turning it on/off, and adjusting the speed of the fan, or the temperature set on the thermostat. These applications were specifically designed to be controlled using micro-interactions — either with gaze or mobile touch input. Figure 6.6 shows the interfaces of the lamp, fan, and thermostat apps on the smart glasses. Note that the commands were placed on the sides (left, right, up, and down) so that they could be activated using a corresponding directional gaze gesture. For the mobile phone we implemented an application in Qt/QML having 3 tabs, one for each device, as shown in Figure 6.7. We also considered using the AR tags for mobile, but we noticed that the phone's tracking capabilities and processing power would not be on par with the host computer.

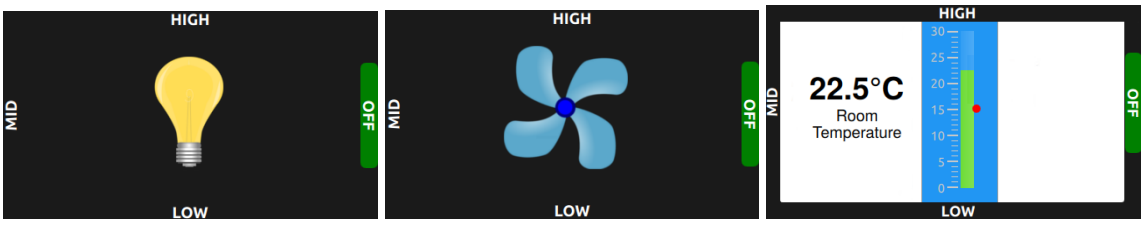

Figure 6.6: *GIMIC applications to control a smart lamp (left), a smart fan (center), and a smart thermostat (right).*

The study follows a within-subject design with interaction mode (GIMIC vs mobile

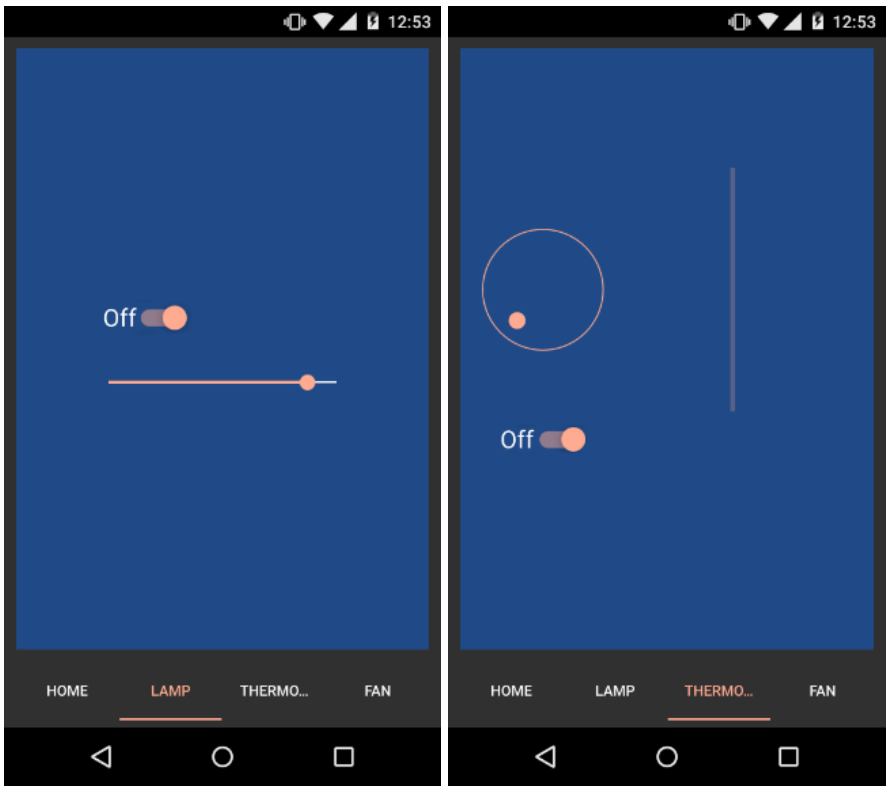

Figure 6.7: *The mobile application with a tab for each smart device. The lamp (left) has an on/off switch button and a dimmer slider. The thermostat (right) also has on/off switch and a wheel for temperature.*

phone) as the independent variable and task completion time as the dependent variable. Participants started with either GIMIC or mobile (randomly and balanced) and completed a set of 12 trials using one of the conditions. After a short break of roughly 3 minutes, they started the set of 12 trials using the other interaction mode. Before the set of trials, volunteers participated in a brief training session to learn the tasks using each mode.

To use the wearable platform, participants first completed the standard Pupil Labs calibration procedure with 9 points. The calibration points were displayed on a 24" monitor placed about 60 cm from the participant's head. After calibrating the eye tracker with its scene camera, participants were subjected to the HUD 9-point calibration procedure. Following this step, they were asked to practice gaze gestures in each of the four directions.

Each set of 12 trials started with a welcome message indicating which interaction mode was going to be tested. Participants had to press the space bar to step through the experiment. For each trial, a brief instruction was presented, such as "Turn the lamp on" or "Set the fan to high". The volunteers had to press the space bar to start the trial after reading the instruction, and press the space bar again when the task was completed. The time for completion was considered as the interval between the two presses. Instructions for the next trial immediately followed the end of a trial, until all 12 were completed.

Figure 6.8 shows the initial screen for the tasks using the wearable platform. The

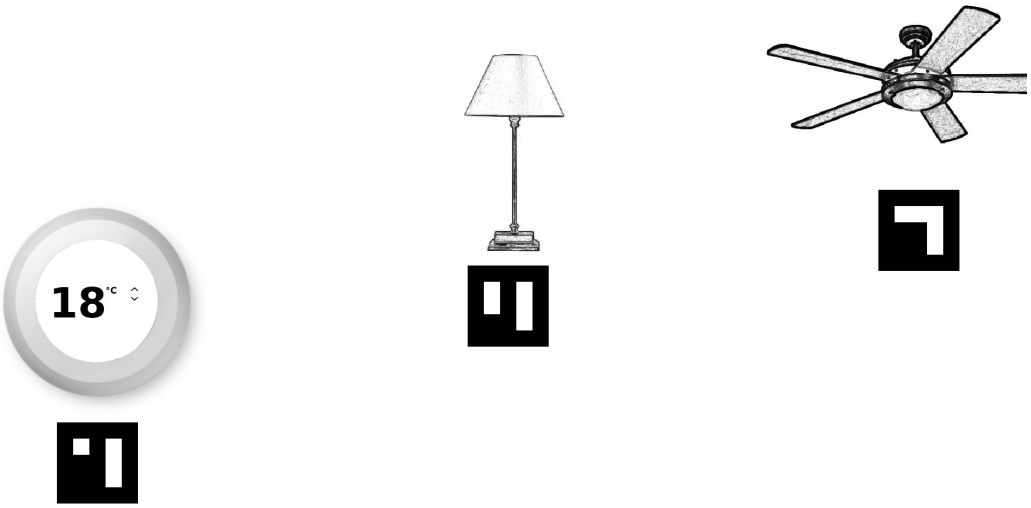

Figure 6.8: *Initial screen shown during the experiment consisting of the smart objects and their corresponding AR markers.*

application that controls each device was shown on the HUD after the user established eye contact with the corresponding AR marker. After engaging with the desired device, the user had to identify the correct command and perform the corresponding eye gesture. The participant was allowed to end the trial after the wearable interface was updated with the new device state.

To evaluate the performance in the mobile scenario, the phone was initially always placed on the table in front of the user, facing down. To complete a trial, the user had to pick up the phone, unlock it using a 3-stroke gesture, open the application that controlled the devices, select the appropriate device, and configure it according to the instructions. Figure 6.7 shows the mobile phone application to adjust the lamp and thermostat parameters. At the end of each trial, the phone was reset to start at the home screen for the next trial and placed facing down on the table again. The application was located on the top left corner of the second page, so it was easy to activate using one gesture and one tap after the phone was unlocked. The same location was used during practice sessions.

Results for all participants are shown as box plots in Figure 6.9. The lower and upper horizontal lines of each box correspond to the 1st and 3rd quartiles. The median is shown in red, and the limits of the dashed lines indicate the minimum and maximum values not considered outliers.

Because of the small sample size and the lack of evidence for a parametric distribution in both cases, we opted for a non-parametric test to assess the statistical significance of these results. Since we used a within-subjects design with correlated samples and two conditions, we ran a one-tailed Wilcoxon signed-rank test. Despite its lesser statistical power, we still managed to observe that GIMIC's completion time on the task was significantly smaller than the time required in the mobile scenario ($W=1$, $p = 0.016$).

Observe that the GIMIC median total task time to complete 12 trials was below 10 seconds for all participants. Only participants 5 and 6 had a few outliers, of more than 20

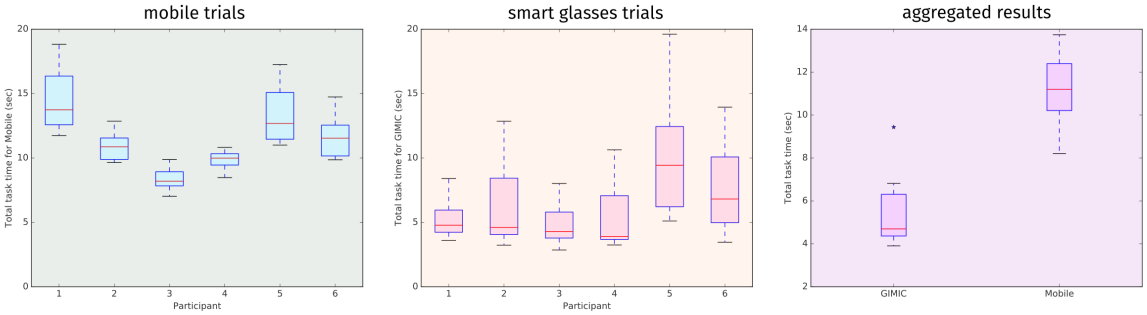

Figure 6.9: *In the left, results for all participants in the mobile scenario. In the middle, are the results for all participants with GIMIC. In the right, the aggregated results for both scenarios.*

seconds (not shown in the chart). Figure 6.9 shows the results using the mobile phone. Observe that the median time for most participants was above 10 seconds, with a few outliers above 20 seconds (not shown in the chart).

Figure 6.9 also shows the results considering all participants for each interaction mode. As can be observed, the grand median for GIMIC was 4.7 seconds, while the grand median for the mobile phone was 11.2 seconds. We can also notice in Figure 6.9 that the time distribution for the mobile phone was almost symmetric, while for GIMIC it was right-skewed, basically due to participant 5 having a larger median than the others.

6.1.4 Discussion and conclusion

In this investigation, we discussed some scenarios that could benefit from GIMIC. We stress that GIMIC is only appropriate when engaging with smart devices to perform brief tasks with simple commands. There are several other situations in which interacting with smart devices requires more complex techniques, such as long text entry or manipulation of 3D objects rendered in AR. Despite GIMIC having demonstrated significant gains in performance over the baseline method, there are several other relevant dimensions that were not addressed in this pilot study, such as user experience.

In spite of some technical difficulties, all users were able to complete the tasks using GIMIC, on average, in less than half the time they took to complete an analog scenario using the mobile phone, with a statistically-measured significance. This result can be even more remarkable if we consider that both the prototype's hardware and software architecture were very experimental and prone to unexpected behavior when compared to the mobile platform.

For instance, we believe that the fact of two participants having shown outliers greater than 20 s was a result of momentary pupil tracking issues — but it could also be related to the volunteer's struggle in performing the correct eye gestures. This difficulty could indicate, for example, that the training session should have been longer prior to the trials. In the mobile condition though, the variance was smaller, indicating a more consistent behavior from volunteers, but it could also have been due to familiarity with the platform.

It could be argued that requiring the user to pick up the phone, unlock it, and finally

start the app was not a fair experimental choice in comparison to GIMIC. We refute this argument by stating that our goal was to compare interactive scenarios, not simply interactive modes. Besides, the browsing latency with mobile trials was not observed in practice, since the app was placed in the same screen location, which allowed participants to remember it. But even if we managed to turn the mobile platform into a wearable one — i.e., a device that is always on and available to the user — we argue that GIMIC still presents potential advantages that go beyond task performance, such as user privacy, consistency with user visual attention, or hands-free interaction with distant devices.

In the pilot study, we limited the GIMIC selection mechanism to gaze gestures, despite also implementing head gestures. Our goal was to demonstrate its feasibility in scenarios with brief interactions, but there is no reason why GIMIC could not be combined with other hands-free interaction modes, such as speech and head gestures. In fact, we see and advocate GIMIC as an enhancing principle for the whole interactive scenario.

6.2 Exploiting the Midas Touch for seamless interaction in 2D

In this investigation, we explored the idea of taking advantage of the Midas touch problem (MTP) (Jacob, 1991) to design seamless interactive techniques, instead of creating safety measures to avoid it. We argue that mechanisms such as *dwell-time* or *touch* selections are like breaks that can be removed under certain circumstances so that the user can achieve higher states of flow.

The concept of flow (Nakamura and Csikszentmihalyi, 2014) regards a highly focused mental state where users are fully immersed in the primary task they are performing. It is highly desirable to create interaction designs that allow users to "flow". In typical computer applications, such as text editors, users can fully focus on writing (the primary task) after they become comfortable using the interface. Typical graphical user interfaces (GUIs) use hierarchical menus for controlling and configuring an application and frequently used resources are placed in menu bars for quick access. Easy transition between resources or interaction modes provided by the interface facilitates the user to achieve a state of flow.

The transition between modes can disturb productivity though, as users have to temporarily relinquish their primary task and navigate through the interface to select the desired mode. Gaze might be helpful in this sense, to accelerate pointing and menu navigation. Dwelling at a target until it gets selected is the most common method used to circumvent the MTP. It is not ideal though: short dwell times might still cause involuntary selections and long dwell times slow the interaction. Using gaze for pointing and some other mechanism for selection, such as touching or clicking a button (Drewes and Schmidt, 2009), has been suggested as a natural multimodal method robust to the MTP.

In the first part of this “seamless gaze interaction” exploration, we propose a proof of concept where the interface options are triggered by “just looking” at them. In the following sections, we describe how this can be achieved without suffering from the ill effects of the Midas touch problem.

6.2.1 GazeBar: designing a seamless mode-switching interaction

GazeBar is both a novel interface and gaze interaction technique that exploits the original concept of "just-look" for target selection, i.e., no dwell, gesture, touch, or click is required when combined with manual input for the primary task. GazeBar interaction design was inspired by MAGIC Pointing (Zhai *et al.*, 1999) and takes advantage of the spontaneous and characteristic gaze paths made by the user when switching modes in GUIs. Figure 6.10 illustrates the four steps required to switch modes using GazeBar.

First, we assume there is an active mode (blue button in the GazeBar shown in the leftmost picture) and the user is focusing on his or her primary task, which is located within the interface central area. When a change of mode is desired, the user directs his or her eyes toward a bar on the bottom edge of the screen, looking for the appropriate

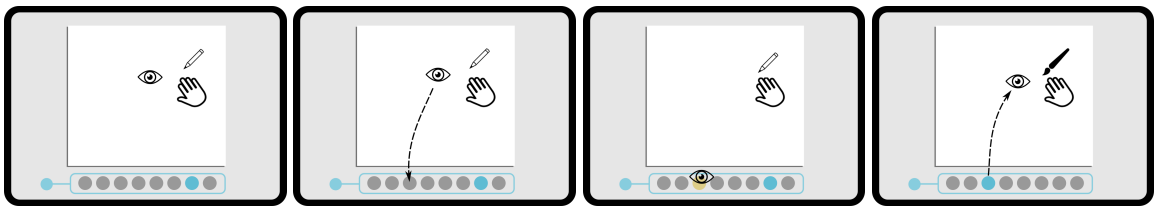

Figure 6.10: *Steps required to switch modes using GazeBar. When the user decides to switch modes, she looks at the GazeBar and hovers over different options on the menu (in yellow). The last gazed option is selected when her gaze leaves the GazeBar.*

mode option. While the gaze is within the bar area, the bar visually indicates which mode is being targeted (yellow button). Upon locating the desired mode, the user looks back at the central area, resuming the primary task. The currently active mode is defined by the last gazed option, and a short visual feedback is shown to the user. Different than gaze-and-touch techniques, no manual confirmation is required.

To give the impression to the user that safe selections are done by simply looking at a button, we use a trigger mechanism similar to reverse-crossing (Feng *et al.*, 2014): once the gaze is captured in the GazeBar area, a selection is only confirmed after leaving it, allowing the user to freely navigate in the bar area. To determine which mode option is to be set, we resort to the minimum distance between the estimated gaze point and a button, so no dwell time is required.

Modes in GazeBar are sorted hierarchically so that the state of secondary modes only becomes available if the parent's mode is selected first, as shown in Figure 6.11. With respect to the interactive logic, this is somehow similar to the idea of hierarchical pie menus, or pEYEs (Urbina and Huckauf, 2010), though our design presents stark differences in visualization and functionality.

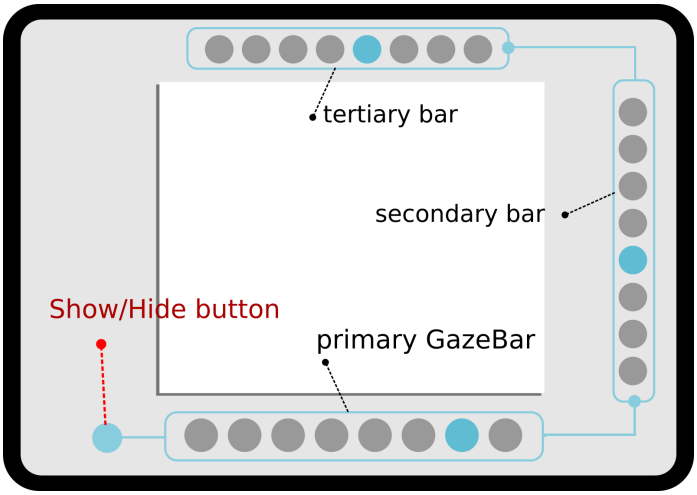

Figure 6.11: *GazeBar options are sorted hierarchically. Root mode options are always at the bottom. One option can trigger a secondary bar, which, by its turn, can trigger another secondary bar. Gazebar interface design supports at most four levels of submenus.*

The content and appearance of GazeBar are also context-sensitive, meaning that if the user is gazing at an application for which there are mode options available, a corresponding

GazeBar will pop up on the screen. But no GazeBar is shown if the gaze context contains no known applications.

Lastly, we do not enforce any coordination between gaze and manual input. Unlike other multimodal gaze-based approaches, we decouple the primary task from the mode-switching task, making manual input solely responsible for the former, and gaze for the latter, promoting a higher state of flow.

Interface design

The use of gaze on GUIs imposes several constraints. In particular, gaze-based interfaces have to at least account for eye-tracking accuracy and eye jittery. Mode buttons in GazeBar were created to span roughly 2^o on screen, which is above the higher bound gaze estimation error found in most commercial eye trackers. Due to eye jittery, we also place buttons relatively apart from each other.

Gaze focus over a mode button is determined by spatial hysteresis (Hansen *et al.*, 2018). We say that the user is gazing at a button if the Euclidean distance between the estimated gaze point G and the button center M is less than an empirically determined threshold d. But we say the user stopped gazing if this distance is greater than $2 \times d$, as indicated by Figure 6.12. This is used to avoid involuntary switches due to eye-tracking instability.

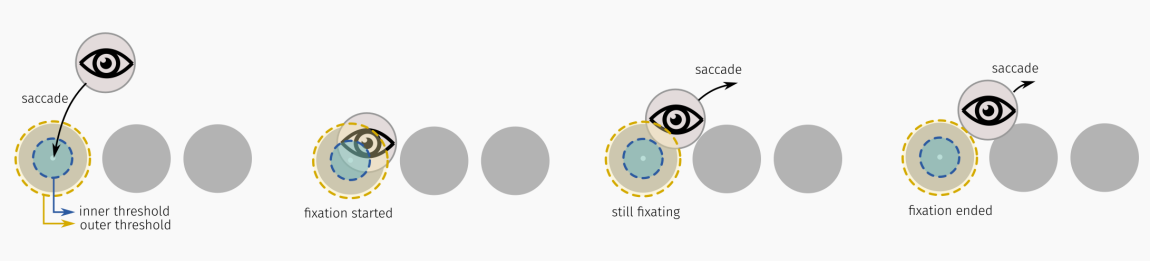

Figure 6.12: *GazeBar uses an implementation of spatial hysteresis (Hansen et al., 2018) based on two criteria to tell when gaze focus over a target starts and ends. Once a gaze estimate is trapped inside a target, GazeBar only considers it out with a larger threshold. This avoids involuntary selections due to eye tracker fidelity or eye jittery.*

Also, while gazing at a mode button, GazeBar highlights its color and the target is expanded. Expanding a target is a way to secure gaze focus detection, since the target spans over a larger screen area and its neighbors are pushed away, lessening selection ambiguity.

Involuntary selections, although still possible, are mitigated by interface design. The bar's one-dimensional format and its placement on screen edges make it more unlikely for accidental gaze incursions, as the user's primary task is located at the central part of the screen. Also, GazeBar and its secondary bars can always be deactivated at the user's will (Figure 6.11). As for involuntary changes, they can always be avoided by fixating on the currently active mode before leaving the bar area.

To minimize the use of screen space, secondary menus are always presented as another bar, and these bars are also placed near the edges of the screen. The hierarchical structure of menus and submenus is demarcated by lines connecting the parent mode option with

the child bar. By default, the current path chosen by the user in the options tree is always visible, which means that multiple bars might be shown on the screen at the same time. This is done to speed up the activation of secondary options without impairing the visibility of the primary task.

6.2.2 GazeBar: prototype

We have built a prototype designed to be used on top of the open-source application for digital painting Krita (Foundation, 2020). In this section, we address some implementation details, the devices used in our setup, and how GazeBar can be adapted as an overlay to any other application intensive on mode switching. The code for this proof of concept (PoC) is available at `https://github.com/elmadjian/GazeBar`.

The PoC was designed to run on laptops and desktop PCs. We used the Tobii 4C eye tracker, which operates at 90 Hz, providing a constant stream of eye-gaze points for a single 24" monitor at 75 Hz (see Figure 6.13). Data is collected through a mixed C#/Python application in the backend, which is also responsible for managing the communication between GazeBar and Krita.

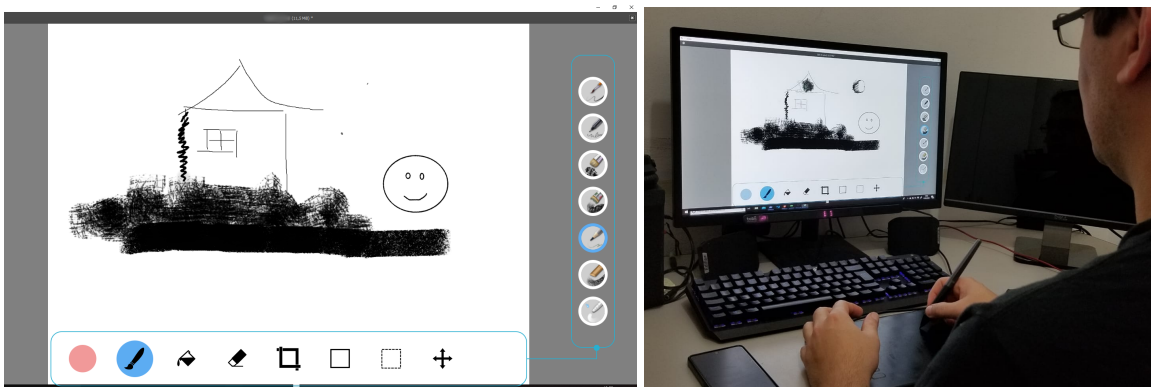

Figure 6.13: *On the left, is a screenshot of our prototype, while on the right we demonstrate its use case with a digital pen and a graphics tablet.*

The GazeBar interface was completely written in QML. The interface controls the actual user interaction with mode buttons, secondary bars, and eventual switches. Changes in the bar state that affects Krita are signaled to the Python backend module using the PySide library for Qt 5. Mode switches are mapped to Krita through hotkeys, which means that some fine-grained input changes, such as color picking, are simply not possible with GazeBar's PoC. However, Krita allows us to create custom hotkeys for any discrete mode available in the GUI, thus enabling GazeBar to capture most of the mode-switching workflow.

Expert users working with digital painting or image editing often make simultaneous use of keyboard and graphics tablets. Non-experts, however, have very limited knowledge of keyboard hotkeys or key + pen combos for mode switching. These users rely on mode options available on the GUI and, therefore, can benefit the most from our PoC.

By using our PoC, the user does not need to lift up the pen and move it to change modes, thus saving manual work. Let us say that the user wants to select a different

brush. The brush options can be found by navigating through the hierarchical options tree. Remember that there is no timeout or specific gestures for a selection. Upon fixating on a bar, the user can explore the brush options freely. The last one gazed before moving the eyes back to the painting canvas will indicate to the system that an input change was made. If a wrong brush is selected, recovering from the error is as simple as gazing at the correct brush and focusing back on the painting.

Since GazeBar does not expect a hand-gaze coordination event, our PoC can be easily adapted to any other application where mode switching is frequent. The only requirement is that the application provides the possibility of changing modes via shortcut keys. The expected adaptation effort would be in terms of changing mode icons, defining a new mode hierarchy, and mapping new shortcuts.

6.2.3 Gazebar: comparative design analysis

The literature has mixed results regarding the efficiency of gaze input in contrast with traditional manual input. Some studies revealed that gaze can be significantly faster than mouse input for target acquisition (Sibert and Jacob, 2000; Vertegaal, 2008), though there are also some diverging results (Miniotas, 2000). As Schuetz et al. pointed out, most of these comparative works are backed up by Fitts' Law, which may show some conflicting results depending on how gaze interaction is assessed (Schuetz *et al.*, 2019).

Saccades are ballistic movements (up to 900^o/s) (Bahill *et al.*, 1975), and therefore cannot benefit from online and controlled adjustments during its course. Thus, efficient use of gaze primarily depends on the size of a saccade with respect to a stimulus and the target size (Schuetz *et al.*, 2019). Additionally, eye tracker fidelity has been found to significantly affect this task (Barz *et al.*, 2018), which brings high variability to the index of difficulty according to the Fitts' Law.

To claim a theoretical efficiency of our design over manual mode-switching in GUIs, we assume the following conditions: the user is seated at 50 cm from a 24" display, our eye tracker accuracy lies between 0.5-1^o, and targets have an approximate size of 2^o. Assuming also that the user's primary task is located at the center of the screen, this results in an average of 27^o of horizontal and 16^o of vertical saccadic span to reach one of the bars.

Based on a quadratic approximation of an empirical model (Abrams *et al.*, 1989), this saccade length demands at most 140 ms. Since overshoots are very likely in this span, two extra short corrective saccades of 30 ms are usually necessary for target acquisition (Schuetz *et al.*, 2019). With a variable fixation time on the target of at most 200 ms, a non-optimistic time estimate for mode-switching with GazeBar would be 500 ms. Note that a fixation dwell is assumed only as a consequence of human cognitive processing (Just and Carpenter, 1976), but it is not required by our technique.

Manual input has been demonstrated to take roughly double the time to close the distance to targets, albeit being much more precise than saccades (Vertegaal, 2008). This suggests that using the GUI for switching modes using a mouse, for example, would require 600 ms just for the complete manual movement, not counting the time for selection and visualization feedback.

A more fundamental argument in favor of our technique is to realize that GazeBar leverages the natural scan paths made when switching modes manually. Thus, the manual response is bounded by the user's visual perceptive task and should improve flow. Unless the flow is broken by unintended selections.

To avoid unintended selections, GazeBar's graphical design has considered the low accuracy of gaze trackers, eye jittery, and other factors as described in Section 6.2.1. Yet, if unintended selections occur often, maybe due to the lack of experience of a novice user, the user experience will be damaged. Therefore, GazeBar is probably not appropriate for tasks such as typing but can benefit tasks that require not-so-frequent selections, and where involuntary selections have a low interaction cost since GazeBar allows very fast recovery.

The GazeBar selection mechanism is designed to be experienced as a "Midas touch", but activation is triggered in practice in a similar fashion to context-switching (Morimoto and Amir, 2010) or reverse-crossing (Feng *et al.*, 2014). These gaze-only techniques, as well as other methods such as dwell-time selection (Majaranta and Räihä, 2002), eye gestures (Møllenbach *et al.*, 2010), or motion correlation (Velloso *et al.*, 2017), create additional preventive steps to avoid the MTP, while GazeBar takes advantage of the expected gaze path to eliminate this need.

Though GazeBar might resemble other multimodal gaze-based interaction techniques, such as gaze-touch (Pfeuffer *et al.*, 2014) and gaze-shifting (Pfeuffer *et al.*, 2015), it is important to notice that gaze and manual inputs are choreographed in their case, while gaze is independent of manual control with GazeBar. We compare our own experience of using GazeBar to driving a car with automatic transmission: while other methods require a "clutch" to change gears, GazeBar improves flow by allowing the user to just look at the desired option.

For traversing hierarchical menus, the design of pEYEs (Urbina and Huckauf, 2010) shares similarities to GazeBar as well. With pEYEs, however, selections are performed by navigating through expanding sub-menus, until only one option is available. For m options per menu and N items total, this leads to a minimum time complexity of $\log_m N$ for each selection. Searching options in GazeBar is also bounded by $\log_m N$, but on average faster because, besides the last branch of the tree being always visible, the user does not necessarily have to reach a leaf to make a selection.

6.2.4 Gazebar: discussion and conclusion

In terms of mode-switching efficiency, we showed that GazeBar's selection technique can be interpreted as an approximate theoretical upper bound for manual input performance since manual input depends on the visual perceptive channel to coordinate selections. By modeling GazeBar's selection on mode-switching scan paths, we discard the preventive measures found in gaze-based techniques, effectively embracing a Midas touch-like selection mechanism that does necessarily imposes involuntary mistakes.

Our prototype mitigates involuntary mode switches by design, such as the one-dimensional bar format, the positioning on the screen, and the two-criteria threshold for activation. We are aware though that an empirical study still remains necessary to

verify the effect of these choices and establish an average expected number of wrong selections. Another sensitive aspect that we shall address in a future experiment is the impact of GazeBar on user experience. Some objective aspects, such as saving users' manual effort compared to other methods, are easy to acknowledge, but others, such as a measurable state of flow, can only be assessed in a user study.

Compared to other gaze-only techniques, our prototype shows that GazeBar has a low error recovery cost. That is because a GazeBar selection is fast and requires minimal eye movement. Dwell-time selection, for instance, not only requires the same amount of movement (in the application workflow), but also additional fixation times. Some users might prefer more low-cost mistakes than costly errors that impose a greater penalty on user performance.

The GazeBar design, though, is not as multipurpose as other methods. However, it is fairly safe to say that complex and multimodal scenarios are more likely to be benefited from this technique, since the idea of GazeBar could be easily adapted to other intensive manual applications, such as word processing, video editing, or 3D modeling. And the principle of modeling gaze-based selections on expected eye movements to maximize the state of flow during interaction could be further applied to tasks other than mode-switching.

While many previous techniques have exploited mechanisms such as dwell-time or eye gestures, and combined gaze with other mechanisms such as speech or finger touch to trigger selections, we proposed a just-look-to-select experience that can improve gaze interaction efficiency.

We expect that interaction flow can be improved by eliminating safety interaction steps (such as dwell-time or mouse click) suggested in the literature to avoid the Midas' touch problem. Of course, this improvement can only be achieved while keeping the number of errors small or compatible with other techniques.

6.3 Exploiting the Midas Touch for seamless interaction in 3D

When it comes to 3D gaze interaction, most of the works in the literature have tried to adapt known 2D techniques to VR or AR scenarios (Bâce *et al.*, 2016; Lee *et al.*, 2014; Nilsson *et al.*, 2009; Park *et al.*, 2008). Genuine methods tailored to the 3D world do exist, but, likewise, they impose several safeguard mechanisms that, in practice, can provoke a feeling of "breaking" the interactive flow (Ahn *et al.*, 2020; Kitajima *et al.*, 2015; Mardanbegi *et al.*, 2019; Pai *et al.*, 2016).

Moreover, using gaze interaction to control our surroundings is not so simple. It is important to tell whether the user is actually trying to access an AR interface or just exploring the scene (Vidal *et al.*, 2014). The MTP is particularly relevant in AR because triggering unwarranted digital information can also impair user visibility and safety.

Another relevant issue in this setting is the so-called *AR pollution*, i.e., the cluttering of unwanted augmented content within the user's field of view, sometimes with layers of digital information superimposed (see Figure 6.14. Though there are several ways to address this, neither automatically triggering AR content nor completely hiding it would be satisfactory, since in the first case we could fall into an MTP, whereas in the second one, we would not allow the user to know what is AR-enabled.

Figure 6.14: *Augmented reality clutter (left), even when subjected to sorting and overlay occlusion mitigation, still may be perceived as AR pollution. Using gaze fixation (right), we correlate the relevancy of AR content to user attention, though triggering unwarranted AR content automatically just based on attention can be considered a form of the Midas touch problem.*

Therefore, in this second step of creating a seamless gaze interaction study, we propose a proof of concept in which we take advantage of the extra depth dimension in 3D to create a "just-looking-to-select" mechanism that does not impair the user's ability to explore the AR world while preventing the triggering of unwarranted digital overlays on display.

6.3.1 V-switch: designing seamless interaction with AR-enabled devices

Moving to a wearable setting with AR-enabled devices, we can take advantage of the fact that interaction is happening in 3D to design seamless gaze-based selection

mechanisms. Because in 3D we have the depth dimension, eye vergence becomes naturally present. Thus, we can leverage vergence as a discriminative response to separate what is gaze perception from what is gaze selection intent.

We call this vergence-based triggering mechanism V-Switch. It operates in a similar way to GazeBar, in the sense that the interaction cannot be fully segregated from the interface. In fact, to exploit the MTP, it needs to be designed this way, so that the user might *just-look-to-select* without having to suffer from the ill effects of the Midas touch problem.

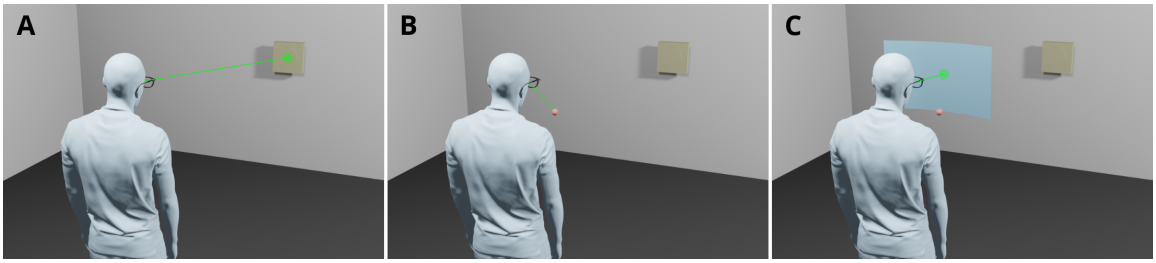

Figure 6.15: *V-Switch Workflow. (A) Eye contact is established with a smart object, at a far distance. (B) A virtual point (the red dot) is rendered near the user, indicating that it is possible to interact with the object. To toggle the object's AR content, the user looks at the sphere. (C) V-Switch detects the vergence movement at the sphere and the interactive AR content is rendered.*

The technique makes two basic assumptions. Firstly, it assumes that all real-world objects that can be enhanced through AR are tagged, i.e., each object has a unique identifier and provides some way to identify it, such as IR lights, AR markers, or RFID tags. Secondly, it assumes that augmentable objects are not within arm's reach from the user, i.e., they are placed, for instance, at 2 m from an individual. Figure 6.15 illustrates the whole interactive workflow, which goes as follows:

1. **target acquisition** — it is performed in the perceptual channel of the user. In other words, the system becomes aware that the user wants to interact with a certain target on gaze contact. Disengaging is accomplished by looking at a different target or at a non-interactive part of the scene for a certain time threshold. There is no MTP here since no change in the state of the target is possible at this point.

2. **associated cue pop-up** — upon gaze contact, one or more visual cues called *V-Dots* are shown at the bottom of the display. The V-Dots are placed closer to the user depthwise and also respect a minimum discriminant distance (MDD) from the target. Because the accuracy of gaze depth estimates deteriorates as we get farther from the user, MDD also varies to accommodate this change.

3. **eye convergence towards the cue** — when the eyes converge towards one of the V-Dots, it can either trigger specific changes of state in the target or toggle an interface to control it. The choice of behavior depends on the task. For micro-interactions (i.e., simple actions such as turning something on or off), there is no need for a management interface.

4. **(optional) target fine control** — if there are multiple options associated with a task or if it is intrinsically complex, the system can show an AR interface near the

selected V-Dot to allow the user to perform a finer control of the target. We can resort to the same principles in GazeBar to maximize the flow, for example, creating a one-dimensional selection bar to perform a "just-by-looking" selection, instead of using a grid interface.

5. **eye divergence** — while gazing at the depth of a V-Dot, changes are yet to be confirmed to the system. The change is committed by performing an eye vergence movement back to the scene or the target. This movement has to cover the MDD to take effect.

6.3.2 V-switch: prototype

Our prototype consists of a Microsoft Hololens (first edition) with a 200 Hz binocular eye tracker from Pupil Labs (Kassner *et al.*, 2014) and an AMD Ryzen 5 2600X PC for processing eye data. The Hololens is capable of tracking and mapping the environment geometry, building its mesh representation in real time. This representation is important because it allows us to know the boundaries and localization of AR-enhanced objects in the scene. With a mesh representation of objects, we are able to determine gaze contact with them. The eye cameras were attached to the Hololens with the frame adapter provided by Pupil Labs and were tethered to the PC (see Figure 6.16).

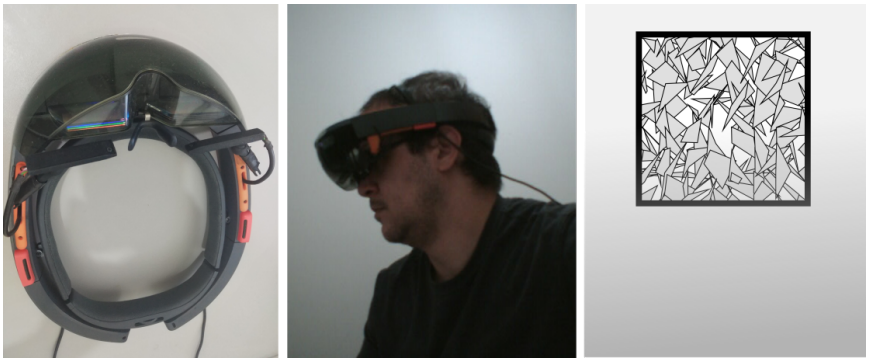

Figure 6.16: *The first two figures on the left show the Hololens 1.0 with the Pupil Labs eye-tracking cameras attached. The eye cameras are tethered to a PC through a USB 2.0 cable. Eye image processing and gaze estimation are done entirely on the PC. The third figure is the fiducial marker used for extended tracking functionality.*

The eye-tracking portion of our software was entirely written in Python 3 and QML. It is cross-platform and runs on a PC. This software is responsible for managing multiple camera streams, tracking pupil center on camera images, performing gaze and depth focus estimation on the head-mounted display (HMD), and establishing a back-and-forth communication with it. QML is used for rendering the graphical user interface (GUI), which provides feedback concerning pupil tracking and gaze estimation quality, as well as letting the user configure camera and network settings.

The AR application running on the Hololens is a Unity 3D project, with its code written in C#, compatible with the Windows Universal Platform. We also use the Vuforia Engine for visual targets. This allows us to define an image on the wall as the ground-zero marker for the Hololens world mesh representation. Thus, even when this marker is no longer

visible, we can have accurate object localization thanks to the HMD positional device tracking.

Communication between the eye-tracking application and the HMD is established via UDP protocol within a local Wi-Fi network. The eye-tracking software updates the HMD app with gaze estimates. Before running the actual app though, a calibration procedure must be performed on the HMD.

For this prototype, we assumed a scenario with three smart appliances that are AR-enabled: a lamp, a thermostat, and a fan (see Figure 6.17. The simplest interaction lies with the lamp: it is just an on/off switch. In the case of the fan, we have a discrete interval with four options: off and speeds 1, 2, and 3. The thermostat also has an on/off switch button and, additionally, a linear temperature gauge. All smart objects can be interacted with using not only V-Switch, but also dwell-time selection, and the head gaze + pinch selection (which is the default method for Hololens applications).

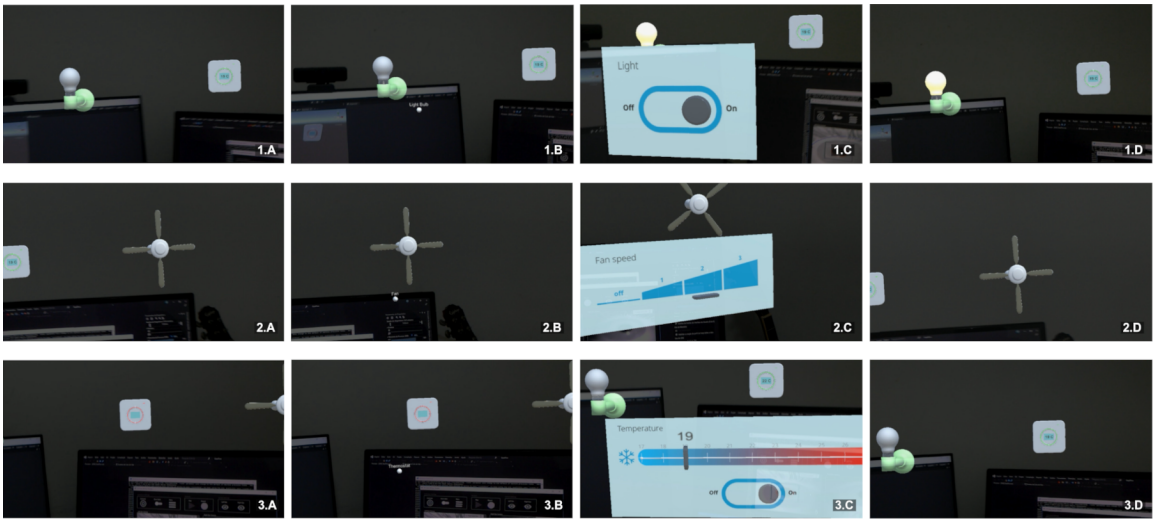

Figure 6.17: *Screen capture of the proof-of-concept from the user standpoint. Each row shows the expected interaction flow with each smart device (1: light bulb, 2: fan, 3: thermostat). Column A depicts the initial scene; column B shows the V-dot popping up in the virtual space, as the user's gaze meets the IoT device; column C displays the device settings UI after the user performed a vergence towards the V-dot; column D shows the moment when the user performs a vergence gesture back to the device, checking its updated state.*

When triggering the interface menu of each appliance using V-Switch, the user is able to set any configuration just by looking at it. In a sense, this follows the GazeBar principles very closely, except that a change of state is only committed when the user performs a divergence, back to the far plane.

Binocular Gaze and Vergence Estimation

To perform binocular gaze estimation, we use reconstructed unit normal vectors from each pupil projected in the left and right eye cameras (Świrski and Dodgson, 2013). Let these vectors be n_L and n_R, respectively. Since the Hololens have an internal 3D coordinate representation for objects rendered in its display, we then proceed to find the left and

right rotation matrices R_L and R_R that will map, respectively, n_L and n_R to this 3D space, as shown in Figure 6.18.

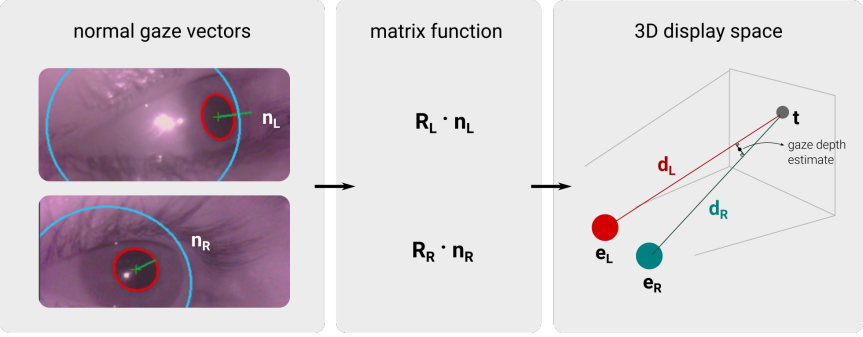

Figure 6.18: *Gaze estimation schematic procedure, including gaze depth estimation as well.*

But first, we place left and right virtual eyeballs e_L and e_R at the origin of the display space. We then separate them symmetrically in the x-axis using the anthropometric interpupillary distance of 63 mm (Gordon *et al.*, 2014). To get coordinate references of this space, the user is asked to look at 3D calibration targets t_i on the display (our prototype uses nine). We then compute the normalized vectors d_L and d_R, which are given by $\frac{t-e_L}{\|t-e_L\|}$ and $\frac{t-e_R}{\|t-e_R\|}$, respectively.

For each target t_i we compute d_{L_i} and d_{R_i}, and collect a few samples of n_{L_i} and n_{R_i}, keeping only the median values for each step i. After collecting all the data, we define the following matrices:

$$N_L = [n_{L_1}, n_{L_2}, ..., n_{L_n}]^T \text{ and } N_R = [n_{R_1}, n_{R_2}, ..., n_{R_n}]^T$$
$$D_L = [d_{L_1}, d_{L_2}, ..., d_{L_n}]^T \text{ and } D_R = [d_{R_1}, d_{R_2}, ..., d_{R_n}]^T$$

From there, all that remains is to solve for R_L and R_R matrices in the following overdetermined linear systems:

$$R_L N_L = D_L$$
$$R_R N_R = D_R$$

Our implementation uses the linear least squares method to find an approximate solution. Once we are able to transform normal gaze vectors from the cameras to the display space, gaze depth estimation is determined in an online fashion by finding the midpoint of the shortest segment between the two rays defined by λd_L and λd_R, as described in Section 5.1.1. Note that to decide whether the user is interacting with a certain object in the scene, V-Switch also takes into account the intersection of both gaze rays with this object as well. Additionally, a smoothing procedure is used to compensate for outliers.

To compute vergence, we first assume that physical and virtual object locations can be defined by 2D planes at different depths in AR. As a simplifying assumption, our V-Switch prototype requires what we call a minimum discriminant depth between the object in the

perceptive space and the user interactive space. Upon eye contact on the object (virtual or not) that can be digitally controlled, a small corresponding V-dot is shown at the bottom of the display, close to the user. To open the interface, the user must look at the V-dot, similar to reaching a remote control. Otherwise, the user can keep exploring the scene as long as desired, with minimal digital clutter other than the small V-dot. Once enabled, the interface remains open while the user is looking at the near interface plane. Error correction when the V-dot is triggered by accident is as simple as looking back at the scene, which causes a divergent eye movement.

6.3.3 V-switch: comparative design analysis

Contrary to a 2D interactive scenario, where manual input is typically considered the preferred way to perform most tasks, in 3D, hand-based interaction becomes much more cumbersome. First, the lack of precision and control is significantly greater; it is often problematic to manipulate distant objects, not within one's reach; and it is tiresome for most people to keep their arms raised for more than a few minutes (Chaconas and Höllerer, 2018; LaViola Jr *et al.*, 2017).

In contrast, one of the earliest comparative studies between manual input and gaze-based interaction in 3D has attested to the advantage of the latter modality in target acquisition (Tanriverdi and Jacob, 2000). Moreover, continuous use of gaze is a natural human state and, thus not prone to provoke fatigue as the "pinch gesture" typically employed in Hololens' manual interactions (Chaconas and Höllerer, 2018). Finally, because the field of view is generally wider in VR/AR, gaze estimation errors can be mitigated with better spacing and sizing of selectable objects (Hansen *et al.*, 2008).

Despite some clear advantages of gaze-based techniques in the 3D domain over manual input, the Midas touch problem remains a constant challenge, regardless of the number of dimensions. Not only that, the introduction of depth, albeit being an opportunity to design new ways of interaction, also adds technical obstacles to the table, such as collinearity of gazed objects, and overall less reliable gaze estimation (Mansouryar *et al.*, 2016).

Still, assuming a sound-calibrated setup, we can, at least theoretically, gauge V-Switch performance against a baseline technique such as dwell-time selection. In the aforementioned scenario of our PoC, let us consider the following interactive journey to establish a benchmark between both of them: the user fixates on a distant object, interacts with it, accesses its configuration menu to alter its state, and finally commits the change, checking the object's visual feedback. The menu is always displayed in an offset position and closer to the user to avoid occlusion of the scene, in case of accidental activations (see Figure 6.17).

Previous studies have demonstrated that eye vergence is a potentially sluggish movement, probably due to the accommodation and pupil interplay. Vergence might have different latency and response times, depending on the depth, if it is symmetrical or not, or whether it is a convergence or a divergence (Ward *et al.*, 2020; Yang *et al.*, 2002), but it is generally acknowledged that it takes, on average 1 s.

In the case of the dwell-time selection, the user would typically have to perform a long fixation at the object to trigger the menu ($\tilde{5}$00 ms), perform a convergence toward

the menu ($\tilde{1}$000 ms), explore the interface ($\tilde{1}$000 ms), commit a change with another long fixation ($\tilde{5}$00 ms), close the UI, again with the dwell-time selection ($\tilde{5}$00 ms), and finally perform a divergence back to the smart device to check its state ($\tilde{1}$000 ms). So, on average, this interactive path would take roughly 4.5 s (discarding saccades altogether).

Compared to the same scenario as before, V-Switch only requires the user to make contact with the device and perform a convergence to access the menu ($\tilde{1}$000 ms), explore the interface ($\tilde{1}$000 ms), and commit the changes by performing a divergence back to the scene ($\tilde{1}$000 ms), taking approximately 3 s to complete the same path. Because "seeing is selecting" only in this sandboxed condition (similarly to GazeBar), we remove the extra latency typically added to gaze-based techniques to circumvent the Midas touch problem. Thus, the interaction time is dominated by the vergence response almost in its entirety.

Another key design difference is that V-Switch does not impose any additional eye movement that would not be already performed if the interaction scenario used manual input. Dwell-time, on the other hand, requires the user to constantly relinquish the natural interplay of saccades and fixations when exploring the environment. This could have a potentially positive effect on user experience. Besides, contrary to previous approaches (Ahn *et al.*, 2020; Kitajima *et al.*, 2015; Vidal *et al.*, 2014), vergence in V-Switch is always performed with virtual visual cues on the display (i.e., the V-dot), which should provide more comfort.

6.3.4 V-switch: discussion and conclusion

It is reasonable to state that both GazeBar and V-Switch share similar advantages and deficiencies. As in the case of the former, V-Switch leverages the expected gaze paths in a certain interaction flow as a means to allow selection by "just looking". Because there is always a disguised two-step mechanism for an actual selection, both techniques are able to provide some degree of safety while apparently preserving user freedom to move the eyes. The major downside, again, lies with an ad-hoc design, i.e., it cannot be as easily generalized and ported to different interactive scenarios as, for example, dwell-time selection.

In the particular case of V-Switch, another negative remark is that error recovery might be costly due to the vergence-accommodation interplay, which is not as fast as a saccade. On the other hand, taking advantage of different depth planes of interaction allows for cleaner management of digital overlays in AR/VR. Additionally, in the case of gaze-only interactive techniques, it creates a logical criterion to separate what is a simple eye surveying of the environment from actual eye-based commands to the interface.

Evidently, not all AR scenarios will be covered by dividing the interaction flow between a near and a far plane. Tasks that strictly require the interaction to be constrained to an approximately fixed plane will not benefit from this design.

Because the interpupillary distance becomes less pronounced beyond 1.0 m, it might not be technically possible to devise an interaction scenario with multiple depth planes, as it would be hard to discriminate between them using only vergence as the main input

feature for gaze depth. Though feasible, this would result in controls with different levels of precision and accuracy depending on the gaze depth.

Despite being theoretically fast in the proposed configuration, V-Switch could also be deployed as a meta-technique, i.e., the vergence-based switching mechanism could be combined with other methods tailored for more specific scenarios, such as text entry.

Finally, V-Switch still needs to be empirically validated through a user study, not only to confirm or not the expected performance improvement in this kind of scenario but especially to assess its impact on user experience compared to other baselines, such as dwell-time interaction.

Chapter 7

Conclusion

Wearable computing is often depicted as one of the prime contexts where gaze interaction could have a great impact on user experience, since the use of gaze seems like an intuitive, fast, effortless, and private way of performing several tasks, particularly for micro-interactions, such as target acquisition.

Despite this assumption, there are several technical and interactive challenges associated with the wearable computing scenario that should not be overlooked. In this thesis, we investigated and discussed a few relevant cases motivated by the following issues:

- **3D gaze interaction** - In the wearable context, gaze interaction is primarily expected to be performed in 3D, rather than in 2D, which demands more refined and complex estimation techniques, since depth-related issues become prominent.
- **context matters** - Gaze interaction mode or technique can change significantly depending on the wearable context, and thus cannot be assumed to remain static. In fact, context switching is likely to happen in short amounts of time, so gaining insight into the user's cognitive state or task engagement is a must, either to provide notifications or to enhance task performance.
- **Midas touch** - This conflict becomes even more apparent in the wearable scenario, as the eyes are frequently elicited to acquire information about constantly changing surroundings, or to help the user navigate in the environment. Therefore, wearable gaze interaction should be as transparent as possible.

Given the aforementioned challenges, we opened different fronts of investigation to either provide solutions to some of such problems or to improve the current state of the art. With time, we have realized that there were many technical gaps that should be addressed to foster gaze interaction in the wearable setting. In other words, not only new interactive techniques but also faster and more lightweight eye pattern recognition algorithms or estimation methods that take gaze depth into consideration.

From a more practical standpoint, we proposed novel and generalized gaze estimation techniques for 3D environments that make minimal assumptions and require very low-entry hardware. We also explored how these 3D techniques can provide means to use vergence as a useful eye pattern in wearable gaze interaction since the extra depth

dimension can be an important ally to determine user attention and at the same time avoid the Midas Touch problem.

We advanced the current state of the art on eye pattern recognition, as it is commonly considered a fundamental gateway to getting insights about the user's cognitive state and the task at hand. In particular, we showed how to achieve very high accurate classification performance in real-time of basic eye patterns, such as fixations, saccades, and smooth pursuits. We showed how we can detect reading patterns in an online fashion, using just commodity hardware and low-power requirements, which is an important consideration in the wearable context.

For the issue of the Midas Touch conflict and wearable gaze interaction per se, we designed and proposed the GazeBar and V-Switch techniques, which are part of a broader idea of modeling gaze input by leveraging the natural eye movement patterns associated with a task. This way, it is expected that gaze interaction becomes more transparent and lightweight cognitively. In this sense, we also investigated how the gaze can be particularly useful in AR scenarios for micro-interactions.

Besides proposing these techniques, to facilitate future research and enhance reproducibility, we also provided several publicly available datasets, which help tackle the problem of data scarcity for this kind of research. We believe that these datasets will not only support future work but also encourage novel research in the field.

With the current advancements, we have now a more clear understanding of what is viable and sensible in the context of gaze-based wearable interaction, as well as what are the more pressing challenges that still need to be handled.

7.1 Significance and limitations

Despite the breadth of this research, it evidently has several limitations. Regarding eye movement understanding, our methods still target only basic eye movement patterns, leaving the problem of classifying a complex and long sequence of eye movements aside. Besides, context awareness involves not only identifying these patterns but also interpreting their relationship with other kinds of inputs, such as environmental sound, scene images, etc. Thus, there might be still a long way towards scene context comprehension using eye movement data alone.

As for 3D gaze estimation, our results show that is still hard to achieve very accurate 3D PoR estimates in the same range of 2D techniques, in which it is fairly common to witness sub-degree accuracy. In part, this can be attributed either to the increased complexity of the problem (since we are adding an extra dimension), as well as to the intrinsic lack of resolution of binocular cues for depth estimation. Nevertheless, there are ways to mitigate these issues, such as: using heuristics related to scene comprehension; calibrating to the 3D virtual space of an augmented reality headset instead of the actual real scene; or using hybrid methods, e.g., 2D calibration with segregated depth estimation.

Finally, it is clear that the interactive methods that we have developed and proposed for the wearable context (i.e., *GazeBar* and *V-Switch*) still need a thorough validation and ample assessment of user experience and usability performance. The empirical evaluation was

very limited, partly due to the COVID-19 pandemic (which subsequently led to increased restrictions and ethical considerations in conducting user studies in the aftermath), but also due to technical limitations of our hardware resources. Additionally, GazeBar and V-Switch do have an ad-hoc design and, despite their potential usefulness and originality, cannot be considered as general as other all-purpose methods.

With all that said, we still provide clear contributions on the aforementioned fronts: novel algorithms and techniques for eye movement pattern recognition, tailored for lightweight and wearable devices; novel techniques for 3D gaze estimation, casting additional light on the issue of understanding PoR in the 3-dimensional space; new approaches to design and perform gaze-based interactions that should be most beneficial in the wearable domain; and open and high-quality data available to support current and future research in these areas.

7.2 Future work

In terms of future work, one of the most critical aspects will be conducting thorough user studies to evaluate the effectiveness and usability of our proposed GazeBar and V-Switch techniques. We aim not only to pursue a short-term evaluation but also longitudinal studies, to give a more detailed understanding of user experience and assess any learning effect. We will also compare our techniques with other traditional gaze-only methods such as dwell-time selection (Majaranta and Räihä, 2002), as well as multimodal, such as gaze-touch (Pfeuffer *et al.*, 2014), and 3D techniques for AR/VR, to determine the most effective and efficient methods for various use cases.

Another important aspect should be investigating user task context detection. This will enable seamless switching between interactive techniques, selecting appropriate ones given a specific scenario. This context detection should take into account not only simple gaze patterns, such as fixations, saccades, and smooth pursuits, but also more complex patterns. Additionally, it should consider what the user is looking at and what task they are performing. By considering these factors, we can develop context detection algorithms that enable more efficient and effective interaction with wearable devices. For instance, if the task at hand involves text entry, we should seamlessly transition to an eye-typing technique instead of persisting in a single all-purpose technique.

Overall, future work will focus on developing new techniques and improving existing ones for gaze-based wearable interaction, leveraging the concept of *flow* that permeates GazeBar and V-Switch, in which the Midas touch problem is addressed by modeling interaction implicitly in the task instead of constraining the user with safety mechanisms. With this paradigm shift in interaction design, along with better scene context and intent detection, we expect to make gaze interaction more effective, efficient, and usable in real-world scenarios.

Published work

The following is a list of research published during my time as a Ph.D. candidate:

1. **Elmadjian, Carlos**; Gonzales, Candy; Costa, Rodrigo Lima da; Morimoto, Carlos H. *Online eye-movement classification with temporal convolutional networks*. Behavior Research Methods, v. 1, p. 1-19, 2022.

2. **Elmadjian, Carlos**; Gonzales, Candy; Morimoto, Carlos H. *Eye Movement Classification with Temporal Convolutional Networks*. Lecture Notes in Computer Science. 1ed.: Springer International Publishing, 2021, v., p. 390-404.

3. **Elmadjian, Carlos**; Morimoto, Carlos H. *GazeBar: Exploiting the Midas Touch in Gaze Interaction*. In: CHI '21: CHI Conference on Human Factors in Computing Systems, 2021, Yokohama Japan. Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems. New York: ACM. p. 1.

4. Freire, Fernando; Rosa, Thatiane; Feulo, Guilherme; **Elmadjian, Carlos**; Cordeiro, Renato; Moura, Shayenne; Andrade, Acácio; de Omena, Lucy Anne; Vicente, Augusto; Marques, Felipe; Sheffer, Aléxia; Hideki, Otávio; Nascimento, Patrícia; Cordeiro, Daniel; Goldman, Alfredo. *Toward Development of A.D.A. - Advanced Distributed Assistant*. In: XXI Simpósio em Sistemas Computacionais de Alto Desempenho, 2020, Brasil. Anais do XXI Simpósio em Sistemas Computacionais de Alto Desempenho (WSCAD 2020). p. 203.

5. Shukla, Pushkar; **Elmadjian, Carlos**; Sharan, Richika; Kulkarni, Vivek; Turk, Matthew; Wang, William Yang. *What Should I Ask? Using Conversationally Informative Rewards for Goal-oriented Visual Dialog*. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, Florence. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2019. p. 6442.

6. Yamashita, Cintia; de Mello e Sousa, Silvia Helena; Kaminski, Michael A.; Alves Martins, Maria Virgínia; **Elmadjian, Carlos Eduardo Leão**; Nagai, Renata Hanae; Yamamoto, Naira Tiemi; Koutsoukos, Eduardo Apostolos Machado; Figueira, Rubens Cesar Lopes. *Description, distribution and ecology of living Reophax pyriformis n. sp.* (Campos Basin, South Atlantic Ocean). REVUE DE MICROPALÉONTOLOGIE, v. 64, p. 100360, 2019.

7. Carneiro, Alex Torquato S.; **Elmadjian, Carlos Eduardo L.**; Gonzales, Candy; Coutinho, Flavio L.; Morimoto, Carlos H. *PursuitPass: A Visual Pursuit-Based User*

Authentication System. In: 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2019, Rio de Janeiro. 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2019. p. 226.

8. **Elmadjian, Carlos**; Shukla, Pushkar; Tula, Antonio Diaz; Morimoto, Carlos H. *3D gaze estimation in the scene volume with a head-mounted eye tracker*. In: 2018 ACM Symposium on Eye Tracking Research and Applications, 2018, Warsaw. Proceedings of the Workshop on Communication by Gaze Interaction - COGAIN '18, 2018. p. 1.

9. Shukla, Pushkar; Sadana, Hemant; Bansal, Apaar; Verma, Deepak; **Elmadjian, Carlos**; Raman, Balasubramanian; Turk, Matthew. *Automatic Cricket Highlight Generation Using Event-Driven and Excitement-Based Features*. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018, Salt Lake City. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018. p. 1881.

10. **Elmadjian, C. E. L.**; Tula, A. D.; Aluani, F. O.; Morimoto, C. H. Gaze interaction using low-resolution images at 5 FPS. In: 19th European Conference on Eye Movements, 2017, Wuppertal. 2017 COGAIN Symposium, 2017.

11. **Elmadjian, C. E. L.**; Kurauchi, A. T. N.; Morimoto, C. H. *Recognizing reading behavior in low-power conditions: a step further towards wearable computing*. In: XV Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais, 2016, São Paulo. Workshop on Eye Gaze Applications, 2016.

Bibliography

Abbott and Faisal(2012) William Welby Abbott and Aldo Ahmed Faisal. Ultra-low-cost 3d gaze estimation: an intuitive high information throughput compliment to direct brain–machine interfaces. *Journal of neural engineering*, 9(4):046016. (Cited on pgs. 21, 22, and 58)

Abdrabou *et al.*(2021) Yasmeen Abdrabou, Ahmed Shams, Mohamed Omar Mantawy, Anam Ahmad Khan, Mohamed Khamis, Florian Alt and Yomna Abdelrahman. Gazemeter: Exploring the usage of gaze behaviour to enhance password assessments. In *ACM Symposium on Eye Tracking Research and Applications*, New York, NY, USA. Association for Computing Machinery. ISBN 9781450383448. URL: https://doi.org/10.1145/3448017.3457384. (Cited on pg. 17)

Abowd *et al.*(1999) Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith and Pete Steggles. Towards a better understanding of context and context-awareness. In Hans-Werner Gellersen, editor, *Handheld and Ubiquitous Computing, First International Symposium, HUC'99, Karlsruhe, Germany, September 27-29, 1999, Proceedings*, volume 1707 of *Lecture Notes in Computer Science*, pgs. 304–307. Springer. doi: 10.1007/3-540-48157-5_29. URL: https://doi.org/10.1007/3-540-48157-5_29. (Cited on pg. 28)

Abrams *et al.*(1989) Richard A Abrams, David E Meyer and Sylvan Kornblum. Speed and accuracy of saccadic eye movements: characteristics of impulse variability in the oculomotor system. *Journal of Experimental Psychology: Human Perception and Performance*, 15(3):529. (Cited on pg. 84)

Agtzidis *et al.*(2016) I. Agtzidis, M. Startsev and M. Dorr. In the pursuit of (ground) truth: a hand-labelling tool for eye movements recorded during dynamic scene viewing. In *2016 IEEE Second Workshop on Eye Tracking and Visualization (ETVIS)*, pgs. 65–68. (Cited on pg. 18)

Agustin *et al.*(2009) Javier San Agustin, Julio C. Mateo, John Paulin Hansen and Arantxa Villanueva. Evaluation of the potential of gaze input for game interaction. *PsychNology J.*, 7(2):213–236. URL: http://www.psychnology.org/File/PNJ7%282%29/PSYCHNOLOGY_JOURNAL_7_2_SANAGUSTIN.pdf. (Cited on pg. 24)

Ahn *et al.*(2020) Sunggeun Ahn, Jeongmin Son, Sangyoon Lee and Geehyuk Lee. Verge-it: Gaze interaction for a binocular head-worn display using modulated disparity vergence eye movement. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, pg. 1–7, New York, NY, USA. Association for Computing Machinery. ISBN 9781450368193. doi: 10.1145/3334480.3382908. URL: https://doi.org/10.1145/3334480.3382908. (Cited on pgs. 28, 87, and 93)

Ajanki *et al.***(2011)** Antti Ajanki, Mark Billinghurst, Hannes Gamper, Toni Järvenpää, Melih Kandemir, Samuel Kaski, Markus Koskela, Mikko Kurimo, Jorma Laaksonen, Kai Puolamäki, Teemu Ruokolainen and Timo Tossavainen. An augmented reality interface to contextual information. *Virtual Real.*, 15(2-3):161–173. doi: 10.1007/s10055-010-0183-5. URL: https://doi.org/10.1007/s10055-010-0183-5. (Cited on pgs. 25, 28, and 29)

Akkil *et al.***(2016)** Deepak Akkil, Andrés Lucero, Jari Kangas, Tero Jokela, Marja Salmimaa and Roope Raisamo. User expectations of everyday gaze interaction on smartglasses. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction, Gothenburg, Sweden, October 23 - 27, 2016*, pg. 24. ACM. doi: 10.1145/2971485.2971496. URL: https://doi.org/10.1145/2971485.2971496. (Cited on pgs. 1 and 26)

Arabadzhiyska *et al.***(2017)** Elena Arabadzhiyska, Okan Tarhan Tursun, Karol Myszkowski, Hans-Peter Seidel and Piotr Didyk. Saccade landing position prediction for gaze-contingent rendering. *ACM Trans. Graph.*, 36(4):50:1–50:12. doi: 10.1145/3072959.3073642. URL: https://doi.org/10.1145/3072959.3073642. (Cited on pg. 27)

Ashbrook(2010) Daniel L. Ashbrook. *Enabling Mobile Microinteractions.* Tese de Doutorado, Georgia Institute of Technology, Atlanta, GA, USA. AAI3414437. (Cited on pg. 69)

Bâce *et al.***(2016)** Mihai Bâce, Teemu Leppänen, David Gil de Gomez and Argenis Ramirez Gomez. ubigaze: ubiquitous augmented reality messaging using gaze gestures. In Steven Zhiying Zhou and Börje Karlsson, editors, *SIGGRAPH ASIA 2016, Macao, December 5-8, 2016 - Mobile Graphics and Interactive Applications*, pgs. 11:1–11:5. ACM. doi: 10.1145/2999508.2999530. URL: https://doi.org/10.1145/2999508.2999530. (Cited on pgs. 27 and 87)

Bahill *et al.***(1975)** A. Terry Bahill, Michael R. Clark and Lawrence Stark. The main sequence, a tool for studying human eye movements. *Mathematical biosciences*, 24(3-4): 191–204. (Cited on pg. 84)

Bai *et al.***(2018)** Shaojie Bai, J. Zico Kolter and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271. URL: http://arxiv.org/abs/1803.01271. (Cited on pgs. 15, 19, and 39)

Barea *et al.***(2002)** Rafael Barea, Luciano Boquete, Manuel Mazo and Elena López. System for assisted mobility using eye movements based on electrooculography. *IEEE transactions on neural systems and rehabilitation engineering*, 10(4):209–218. (Cited on pg. 8)

Barz *et al.***(2018)** Michael Barz, Florian Daiber, Daniel Sonntag and Andreas Bulling. Error-aware gaze-based interfaces for robust mobile gaze interaction. In Bonita Sharif and Krzysztof Krejtz, editors, *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, ETRA 2018, Warsaw, Poland, June 14-17, 2018*, pgs. 24:1–24:10. ACM. doi: 10.1145/3204493.3204536. URL: https://doi.org/10.1145/3204493.3204536. (Cited on pg. 84)

Bayat and Pomplun(2017) Akram Bayat and Marc Pomplun. Biometric identification through eye-movement patterns. In Daniel N. Cassenti, editor, *Advances in Human Factors in Simulation and Modeling - Proceedings of the AHFE 2017 International Conference on Human Factors in Simulation and Modeling, July 17-21, 2017, The Westin Bonaventure*

Hotel, Los Angeles, California, USA, volume 591 of *Advances in Intelligent Systems and Computing*, pgs. 583–594. Springer. doi: 10.1007/978-3-319-60591-3_53. URL: https://doi.org/10.1007/978-3-319-60591-3_53. (Cited on pg. 17)

Bektas(2020) Kenan Bektas. Toward A pervasive gaze-contingent assistance system: Attention and context-awareness in augmented reality. In Andreas Bulling, Anke Huckauf, Eakta Jain, Ralph Radach and Daniel Weiskopf, editors, *ETRA '20 Adjunct: 2020 Symposium on Eye Tracking Research and Applications, Stuttgart, Germany, June 2-5, 2020, Adjunct Volume*, pgs. 36:1–36:3. ACM. doi: 10.1145/3379157.3391657. URL: https://doi.org/10.1145/3379157.3391657. (Cited on pg. 3)

Berg *et al.***(2009)** David J. Berg, Susan E. Boehnke, Robert A. Marino, Douglas P. Munoz and Laurent Itti. Free viewing of dynamic stimuli by humans and monkeys. *Journal of Vision*, 9(5):19–19. ISSN 1534-7362. doi: 10.1167/9.5.19. (Cited on pg. 18)

Berndt *et al.***(2019)** Samuel-Hunter Berndt, Douglas Kirkpatrick, Timothy Taviano and Oleg Komogortsev. Tertiary eye movement classification by a hybrid algorithm. *CoRR*, abs/1904.10085. URL: http://arxiv.org/abs/1904.10085. (Cited on pgs. 17 and 32)

Beymer and Flickner(2003) David Beymer and Myron Flickner. Eye gaze tracking using an active stereo head. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA*, pgs. 451–458. IEEE Computer Society. doi: 10.1109/CVPR.2003.1211502. URL: https://doi.org/10.1109/CVPR.2003.1211502. (Cited on pg. 20)

Biedert *et al.***(2012)** Ralf Biedert, Jörn Hees, Andreas Dengel and Georg Buscher. A robust realtime reading-skimming classifier. In Carlos Hitoshi Morimoto, Howell O. Istance, Stephen N. Spencer, Jeffrey B. Mulligan and Pernilla Qvarfordt, editors, *Proceedings of the 2012 Symposium on Eye-Tracking Research and Applications, ETRA 2012, Santa Barbara, CA, USA, March 28-30, 2012*, pgs. 123–130. ACM. doi: 10.1145/2168556.2168575. URL: https://doi.org/10.1145/2168556.2168575. (Cited on pg. 18)

Blattgerste *et al.***(2018)** Jonas Blattgerste, Patrick Renner and Thies Pfeiffer. Advantages of eye-gaze over head-gaze-based selection in virtual and augmented reality under varying field of views. In Carlos Morimoto and Thies Pfeiffer, editors, *Proceedings of the Workshop on Communication by Gaze Interaction, COGAIN@ETRA 2018, Warsaw, Poland, June 15, 2018*, pgs. 1:1–1:9. ACM. doi: 10.1145/3206343.3206349. URL: https://doi.org/10.1145/3206343.3206349. (Cited on pg. 26)

Bolt(1981) Richard A Bolt. Gaze-orchestrated dynamic windows. In *ACM SIGGRAPH Computer Graphics*, volume 15, pgs. 109–119. ACM. (Cited on pg. 23)

Brolly and Mulligan(2004) Xavier L. C. Brolly and Jeffrey B. Mulligan. Implicit calibration of a remote gaze tracker. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2004, Washington, DC, USA, June 27 - July 2, 2004*, pg. 134. IEEE Computer Society. doi: 10.1109/CVPR.2004.366. URL: https://doi.org/10.1109/CVPR.2004.366. (Cited on pg. 20)

Bulling and Gellersen(2010) Andreas Bulling and Hans Gellersen. Toward mobile eye-based human-computer interaction. *IEEE Pervasive Computing*, 9(4):8–12. (Cited on pg. 1)

Bulling *et al.***(2011)** Andreas Bulling, Daniel Roggen and Gerhard Tröster. What's in the eyes for context-awareness? *IEEE Pervasive Comput.*, 10(2):48–57. doi: 10.1109/MPRV.2010.49. URL: https://doi.org/10.1109/MPRV.2010.49. (Cited on pg. 1)

Burch *et al.***(2019)** Michael Burch, Ayush Kumar and Neil Timmermans. An interactive web-based visual analytics tool for detecting strategic eye movement patterns. In Krzysztof Krejtz and Bonita Sharif, editors, *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, ETRA 2019, Denver , CO, USA, June 25-28, 2019*, pgs. 93:1–93:5. ACM. doi: 10.1145/3317960.3321615. URL: https://doi.org/10.1145/3317960.3321615. (Cited on pg. 17)

Buscher *et al.***(2008)** Georg Buscher, Andreas Dengel and Ludger van Elst. Eye movements as implicit relevance feedback. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pgs. 2991–2996, New York, NY, USA. ACM. ISBN 978-1-60558-012-8. (Cited on pgs. 49, 52, and 55)

Calhoun and McMillan(1998) G.L. Calhoun and G.R. McMillan. Hands-free input devices for wearable computers. In *Proceedings Fourth Annual Symposium on Human Interaction with Complex Systems*, pgs. 118–123. doi: 10.1109/HUICS.1998.659965. (Cited on pg. 1)

Campbell and Maglio(2001a) Christopher S. Campbell and Paul P. Maglio. A robust algorithm for reading detection. In *Proceedings of the 2001 Workshop on Perceptive User Interfaces*, PUI '01, pgs. 1–7, New York, NY, USA. ACM. (Cited on pgs. 49 and 52)

Campbell and Maglio(2001b) Christopher S. Campbell and Paul P. Maglio. A robust algorithm for reading detection. In *Proceedings of the 2001 workshop on Perceptive user interfaces, PUI '01, Orlando, Florida, USA, November 15-16, 2001*, pgs. 3:1–3:7. ACM. doi: 10.1145/971478.971503. URL: https://doi.org/10.1145/971478.971503. (Cited on pgs. 18 and 19)

Carneiro *et al.***(2019)** Alex Torquato S. Carneiro, Carlos Eduardo L. Elmadjian, Candy Gonzales, Flavio Luiz Coutinho and Carlos H. Morimoto. Pursuitpass: A visual pursuit-based user authentication system. In *32nd SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2019, Rio de Janeiro, Brazil, October 28-30, 2019*, pgs. 226–233. IEEE. doi: 10.1109/SIBGRAPI.2019.00038. URL: https://doi.org/10.1109/SIBGRAPI.2019.00038. (Cited on pgs. 18, 19, and 24)

Castellina and Corno(2008) Emiliano Castellina and Fulvio Corno. Multimodal gaze interaction in 3d virtual environments. *COGAIN*, 8:33–37. (Cited on pg. 25)

Cazzato *et al.***(2020)** Dario Cazzato, Marco Leo, Cosimo Distante and Holger Voos. When I look into your eyes: A survey on computer vision contributions for human gaze estimation and tracking. *Sensors*, 20(13):3739. doi: 10.3390/s20133739. URL: https://doi.org/10.3390/s20133739. (Cited on pgs. 10, 19, and 20)

Chaconas and Höllerer(2018) Nikolas Chaconas and Tobias Höllerer. An evaluation of bimanual gestures on the microsoft hololens. In Kiyoshi Kiyokawa, Frank Steinicke, Bruce H. Thomas and Greg Welch, editors, *2018 IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2018, Tuebingen/Reutlingen, Germany, 18-22 March 2018*, pgs. 33–40. IEEE Computer Society. doi: 10.1109/VR.2018.8446320. URL: https://doi.org/10.1109/VR.2018.8446320. (Cited on pg. 92)

Chennamma and Yuan(2013) HR Chennamma and Xiaohui Yuan. A survey on eye-gaze tracking techniques. *arXiv preprint arXiv:1312.6410*. (Cited on pg. 8)

Chitty(2013) Nell Chitty. User fatigue and eye controlled technology. Dissertação de Mestrado, OCAD university, Toronto, Canada. (Cited on pg. 26)

Cho *et al.*(2014) Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pgs. 1724–1734. ACL. doi: 10.3115/v1/d14-1179. URL: https://doi.org/10.3115/v1/d14-1179. (Cited on pg. 13)

Clark and Chalmers(1998) Andy Clark and David Chalmers. The extended mind. *analysis*, pgs. 7–19. (Cited on pg. 3)

Cournia *et al.*(2003) Nathan Cournia, John D. Smith and Andrew T. Duchowski. Gaze- vs. hand-based pointing in virtual environments. In Gilbert Cockton and Panu Korhonen, editors, *Extended abstracts of the 2003 Conference on Human Factors in Computing Systems, CHI 2003, Ft. Lauderdale, Florida, USA, April 5-10, 2003*, pgs. 772–773. ACM. doi: 10.1145/765891.765982. URL: https://doi.org/10.1145/765891.765982. (Cited on pg. 25)

Coutinho and Morimoto(2006) Flavio Luiz Coutinho and Carlos Hitoshi Morimoto. Free head motion eye gaze tracking using a single camera and multiple light sources. In *19th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2006), 8-11 October 2006, Manaus, Amazonas, Brazil*, pgs. 171–178. IEEE Computer Society. doi: 10.1109/SIBGRAPI.2006.21. URL: https://doi.org/10.1109/SIBGRAPI.2006.21. (Cited on pg. 20)

de Greef *et al.*(2009) Tjerk de Greef, Harmen Lafeber, Herre van Oostendorp and Jasper Lindenberg. Eye movement as indicators of mental workload to trigger adaptive automation. In Dylan Schmorrow, Ivy V. Estabrooke and Marc Grootjen, editors, *Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience, 5th International Conference, FAC 2009 Held as Part of HCI International 2009 San Diego, CA, USA, July 19-24, 2009, Proceedings*, volume 5638 of *Lecture Notes in Computer Science*, pgs. 219–228. Springer. doi: 10.1007/978-3-642-02812-0_26. URL: https://doi.org/10.1007/978-3-642-02812-0_26. (Cited on pg. 17)

DeVaul(2004) R.W. DeVaul. *The memory glasses: wearable computing for just-in-time memory support*. Tese de Doutorado, Massachusetts Institute of Technology. (Cited on pg. 1)

Diaz-Tula and Morimoto(2015) Antonio Diaz-Tula and Carlos H Morimoto. Dynamic and Meta-Context Switching for Gaze-Based Interaction. *SBC Journal on Interactive Systems*, 6(1):66–75. (Cited on pg. 74)

Dietrich(2004) Arne Dietrich. Neurocognitive mechanisms underlying the experience of flow. *Consciousness and Cognition*, 13(4):746 – 761. ISSN 1053-8100. doi: https://doi.org/10.1016/j.concog.2004.07.002. URL: http://www.sciencedirect.com/science/article/pii/S1053810004000583. (Cited on pg. 28)

Dorr *et al.*(2009) Michael Dorr, Laura Pomârjanschi and Erhardt Barth. Gaze beats mouse: A case study on a gaze-controlled breakout. *PsychNology J.*, 7(2):197–211. URL: http://www.psychnology.org/File/PNJ7%282%29/PSYCHNOLOGY_JOURNAL_7_2_DORR.pdf. (Cited on pgs. xiii, 24, 32, and 39)

Dorr *et al.*(2010) Michael Dorr, Thomas Martinetz, Karl R. Gegenfurtner and Erhardt Barth. Variability of eye movements when viewing dynamic natural scenes. *Journal of Vision*, 10(10):28–28. ISSN 1534-7362. doi: 10.1167/10.10.28. URL: https://doi.org/10.1167/10.10.28. (Cited on pgs. 18 and 32)

Drewes and Schmidt(2009) Heiko Drewes and Albrecht Schmidt. The magic touch: Combining magic-pointing with a touch-sensitive mouse. In *Human-Computer Interaction–INTERACT 2009: 12th IFIP TC 13 International Conference, Uppsala, Sweden, August 24-28, 2009, Proceedings, Part II 12*, pgs. 415–428. Springer. (Cited on pgs. 29 and 80)

Drewes *et al.*(2007) Heiko Drewes, Alexander De Luca and Albrecht Schmidt. Eye-gaze interaction for mobile phones. In Peter Han Joo Chong and Adrian David Cheok, editors, *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology, Mobility Conference 2007, Singapore, September 10-12, 2007*, pgs. 364–371. ACM. doi: 10.1145/1378063.1378122. URL: https://doi.org/10.1145/1378063.1378122. (Cited on pg. 24)

Duchowski(2017) Andrew T. Duchowski. *Eye tracking methodology: Theory and practice*. Springer. (Cited on pg. 8)

Duchowski(2018) Andrew T. Duchowski. Gaze-based interaction: A 30 year retrospective. *Comput. Graph.*, 73:59–69. doi: 10.1016/j.cag.2018.04.002. URL: https://doi.org/10.1016/j.cag.2018.04.002. (Cited on pgs. 10 and 24)

Duchowski and Çöltekin(2007) Andrew T. Duchowski and Arzu Çöltekin. Foveated gaze-contingent displays for peripheral LOD management, 3d visualization, and stereo imaging. *ACM Trans. Multim. Comput. Commun. Appl.*, 3(4):6:1–6:18. doi: 10.1145/1314303.1314309. URL: https://doi.org/10.1145/1314303.1314309. (Cited on pg. 1)

Duchowski *et al.*(2011) Andrew T. Duchowski, Brandon Pelfrey, Donald H. House and Rui I. Wang. Measuring gaze depth with an eye tracker during stereoscopic display. In Rachel McDonnell, Simon J. Thorpe, Stephen N. Spencer, Diego Gutierrez and Martin A. Giese, editors, *Proceedings of the 8th Symposium on Applied Perception in Graphics and Visualization, APGV 2011, Toulouse, France, August 27-28, 2011*, pgs. 15–22. ACM. doi: 10.1145/2077451.2077454. URL: https://doi.org/10.1145/2077451.2077454. (Cited on pg. 22)

Duchowski *et al.*(2014) Andrew T. Duchowski, Donald H. House, Jordan Gestring, Robert Congdon, Lech Swirski, Neil A. Dodgson, Krzysztof Krejtz and Izabela Krejtz. Comparing estimated gaze depth in virtual and physical environments. In Pernilla Qvarfordt and Dan Witzner Hansen, editors, *Eye Tracking Research and Applications, ETRA '14, Safety Harbor, FL, USA, March 26-28, 2014*, pgs. 103–110. ACM. doi: 10.1145/2578153.2578168. URL: https://doi.org/10.1145/2578153.2578168. (Cited on pg. 22)

Edwards(1998) Gregory W. Edwards. A tool for creating eye-aware applications that adapt to changes in user behaviors. In Meera Blattner and Arthur I. Karshmer, editors,

Proceedings of the Third International ACM Conference on Assistive Technologies, ASSETS 1998, Marina del Rey, CA, USA, April 15-17, 1998, pgs. 67–74. ACM. doi: 10.1145/274497.274511. URL: https://doi.org/10.1145/274497.274511. (Cited on pg. 17)

Elmadjian *et al.*(2016) Carlos Elmadjian, Andrew Kurauchi and Carlos Hitoshi Morimoto. Recognizing reading behavior in low-power conditions: a step further towards wearable computing. In *Extended proceedings of IHC'16, Brazilian Symposium on Human Factors in Computing Systems. Workshop of Eye Gaze Applications - WEGA*, pgs. 222–226, São Paulo. Brazilian Computer Society. URL: http://www.lbd.dcc.ufmg.br/curadoria/minger/recentinsertion/ihc/2016/pdf/043.pdf. (Cited on pg. 49)

Elmadjian *et al.*(2022) Carlos Elmadjian, Candy Gonzales, Rodrigo Lima da Costa and Carlos H Morimoto. Online eye-movement classification with temporal convolutional networks. *Behavior Research Methods*, pgs. 1–19. doi: https://doi.org/10.3758/s13428-022-01978-2. (Cited on pgs. 2 and 39)

Elmadjian and Morimoto(2021) Carlos E. L. Elmadjian and Carlos H. Morimoto. Gazebar: Exploiting the midas touch in gaze interaction. In Yoshifumi Kitamura, Aaron Quigley, Katherine Isbister and Takeo Igarashi, editors, *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama Japan, May 8-13, 2021, Extended Abstracts*, pgs. 248:1–248:7. ACM. doi: 10.1145/3411763.3451703. URL: https://doi.org/10.1145/3411763.3451703. (Cited on pg. 69)

Elmadjian *et al.*(2017) Carlos E. L. Elmadjian, Antonio Diaz-Tula, Fernando O. Aluani and Carlos H. Morimoto. Gaze interaction using low-resolution images at 5 fps. URL: http://cogain2017.cogain.org/camready/talk5-Elmadjian.pdf. (Cited on pg. 49)

Elmadjian *et al.*(2018) Carlos E. L. Elmadjian, Pushkar Shukla, Antonio Diaz Tula and Carlos H. Morimoto. 3d gaze estimation in the scene volume with a head-mounted eye tracker. In Carlos Morimoto and Thies Pfeiffer, editors, *Proceedings of the Workshop on Communication by Gaze Interaction, COGAIN@ETRA 2018, Warsaw, Poland, June 15, 2018*, pgs. 3:1–3:9. ACM. doi: 10.1145/3206343.3206351. URL: https://doi.org/10.1145/3206343.3206351. (Cited on pg. 57)

Elmadjian *et al.*(2020) Carlos E. L. Elmadjian, Candy Gonzales and Carlos H. Morimoto. Eye movement classification with temporal convolutional networks. In Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, Hugo Jair Escalante and Roberto Vezzani, editors, *Pattern Recognition. ICPR International Workshops and Challenges - Virtual Event, January 10-15, 2021, Proceedings, Part III*, volume 12663 of *Lecture Notes in Computer Science*, pgs. 390–404. Springer. doi: 10.1007/978-3-030-68796-0_28. URL: https://doi.org/10.1007/978-3-030-68796-0_28. (Cited on pg. 32)

Essig *et al.*(2006) Kai Essig, Marc Pomplun and Helge J. Ritter. A neural network for 3d gaze recording with binocular eye trackers. *IJPEDS*, 21(2):79–95. doi: 10.1080/17445760500354440. URL: https://doi.org/10.1080/17445760500354440. (Cited on pg. 22)

Esteves *et al.*(2015) Augusto Esteves, Eduardo Velloso, Andreas Bulling and Hans Gellersen. Orbits: Gaze interaction for smart watches using smooth pursuit eye movements. In Celine Latulipe, Bjoern Hartmann and Tovi Grossman, editors, *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, UIST 2015, Charlotte,*

NC, USA, November 8-11, 2015, pgs. 457–466. ACM. doi: 10.1145/2807442.2807499. URL: https://doi.org/10.1145/2807442.2807499. (Cited on pgs. 19, 24, and 29)

Esteves *et al.*(2017) Augusto Esteves, David Verweij, Liza Suraiya, Md. Rasel Islam, Youryang Lee and Ian Oakley. Smoothmoves: Smooth pursuits head movements for augmented reality. In Krzysztof Gajos, Jennifer Mankoff and Chris Harrison, editors, *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, UIST 2017, Quebec City, QC, Canada, October 22 - 25, 2017*, pgs. 167–178. ACM. doi: 10.1145/3126594.3126616. URL: https://doi.org/10.1145/3126594.3126616. (Cited on pg. 25)

Feng *et al.*(2014) Wenxin Feng, Ming Chen and Margrit Betke. Target reverse crossing: a selection method for camera-based mouse-replacement systems. In Fillia Makedon, Mark Clements, Catherine Pelachaud, Vana Kalogeraki and Ilias Maglogiannis, editors, *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments, PETRA 2014, Island of Rhodes, Greece, May 27 - 30, 2014*, pgs. 39:1–39:4. ACM. doi: 10.1145/2674396.2674443. URL: https://doi.org/10.1145/2674396.2674443. (Cited on pgs. 81 and 85)

Feng *et al.*(2021) Wenxin Feng, Jiangnan Zou, Andrew Kurauchi, Carlos H Morimoto and Margrit Betke. Hgaze typing: Head-gesture assisted gaze typing. In *ACM Symposium on Eye Tracking Research and Applications*, New York, NY, USA. Association for Computing Machinery. ISBN 9781450383448. URL: https://doi.org/10.1145/3448017.3457379. (Cited on pg. 17)

Fiala(2005) Mark Fiala. Artag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pgs. 590–596. IEEE Computer Society. doi: 10.1109/CVPR.2005.74. URL: https://doi.org/10.1109/CVPR.2005.74. (Cited on pg. 60)

Foundation(2020) Krita Foundation, Aug 2020. URL: http://krita.org/. (Cited on pg. 83)

Fuhl *et al.*(2016) Wolfgang Fuhl, Marc Tonsen, Andreas Bulling and Enkelejda Kasneci. Pupil detection for head-mounted eye tracking in the wild: an evaluation of the state of the art. *Mach. Vis. Appl.*, 27(8):1275–1288. doi: 10.1007/s00138-016-0776-4. URL: https://doi.org/10.1007/s00138-016-0776-4. (Cited on pg. 2)

Garau *et al.*(2003) Maia Garau, Mel Slater, Vinoba Vinayagamoorthy, Andrea Brogni, Anthony Steed and Martina Angela Sasse. The impact of avatar realism and eye gaze control on perceived quality of communication in a shared immersive virtual environment. In Gilbert Cockton and Panu Korhonen, editors, *Proceedings of the 2003 Conference on Human Factors in Computing Systems, CHI 2003, Ft. Lauderdale, Florida, USA, April 5-10, 2003*, pgs. 529–536. ACM. doi: 10.1145/642611.642703. URL: https://doi.org/10.1145/642611.642703. (Cited on pg. 25)

George and Routray(2016) Anjith George and Aurobinda Routray. A score level fusion method for eye movement biometrics. *Pattern Recognit. Lett.*, 82:207–215. doi: 10.1016/j.patrec.2015.11.020. URL: https://doi.org/10.1016/j.patrec.2015.11.020. (Cited on pg. 17)

Goldberg and Kotval(1999) Joseph H. Goldberg and Xerxes P. Kotval. Computer interface evaluation using eye movements: methods and constructs. *International Journal of Industrial Ergonomics*, 24(6):631–645. (Cited on pg. 49)

Gordon *et al.***(2014)** Claire C Gordon, Cynthia L Blackwell, Bruce Bradtmiller, Joseph L Parham, Patricia Barrientos, Stephen P Paquette, Brian D Corner, Jeremy M Carson, Joseph C Venezia, Belva M Rockwell *et al.* 2012 anthropometric survey of us army personnel: methods and summary statistics. Relatório técnico, Army Natick Soldier Research Development and Engineering Center MA. (Cited on pg. 91)

Grubert *et al.***(2017)** Jens Grubert, Tobias Langlotz, Stefanie Zollmann and Holger Regenbrecht. Towards pervasive augmented reality: Context-awareness in augmented reality. *IEEE Trans. Vis. Comput. Graph.*, 23(6):1706–1724. doi: 10.1109/TVCG.2016.2543720. URL: https://doi.org/10.1109/TVCG.2016.2543720. (Cited on pg. 28)

Guenter *et al.***(2012)** Brian K. Guenter, Mark Finch, Steven Mark Drucker, Desney S. Tan and John Snyder. Foveated 3d graphics. *ACM Trans. Graph.*, 31(6):164:1–164:10. doi: 10.1145/2366145.2366183. URL: https://doi.org/10.1145/2366145.2366183. (Cited on pg. 27)

Guestrin and Eizenman(2006) Elias Daniel Guestrin and Moshe Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on biomedical engineering*, 53(6):1124–1133. (Cited on pgs. 5, 20, 21, and 58)

Hansen and Ji(2010) Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(3):478–500. doi: 10.1109/TPAMI.2009.30. URL: https://doi.org/10.1109/TPAMI.2009.30. (Cited on pgs. 2, 10, and 20)

Hansen and Pece(2005) Dan Witzner Hansen and Arthur E. C. Pece. Eye tracking in the wild. *Comput. Vis. Image Underst.*, 98(1):155–181. doi: 10.1016/j.cviu.2004.07.013. URL: https://doi.org/10.1016/j.cviu.2004.07.013. (Cited on pg. 2)

Hansen *et al.***(2002)** Dan Witzner Hansen, John Paulin Hansen, Mads Nielsen, Anders Sewerin Johansen and Mikkel 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*, pgs. 132–136. IEEE Computer Society. doi: 10.1109/ACV.2002.1182170. URL: https://doi.org/10.1109/ACV.2002.1182170. (Cited on pg. 24)

Hansen *et al.***(2008)** Dan Witzner Hansen, Henrik H. T. Skovsgaard, John Paulin Hansen and Emilie Møllenbach. Noise tolerant selection by gaze-controlled pan and zoom in 3d. In Kari-Jouko Räihä and Andrew T. Duchowski, editors, *Proceedings of the Eye Tracking Research & Application Symposium, ETRA 2008, Savannah, Georgia, USA, March 26-28, 2008*, pgs. 205–212. ACM. doi: 10.1145/1344471.1344521. URL: https://doi.org/10.1145/1344471.1344521. (Cited on pg. 92)

Hansen *et al.***(2018)** John Paulin Hansen, Vijay Rajanna, I. Scott MacKenzie and Per Bækgaard. A fitts' law study of click and dwell interaction by gaze, head and mouse with a head-mounted display. In Carlos Morimoto and Thies Pfeiffer, editors, *Proceedings of the Workshop on Communication by Gaze Interaction, COGAIN@ETRA 2018, Warsaw, Poland, June 15, 2018*, pgs. 7:1–7:5. ACM. doi: 10.1145/3206343.3206344. URL: https://doi.org/10.1145/3206343.3206344. (Cited on pgs. xi and 82)

Hearst *et al.***(1998)** Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13 (4):18–28. (Cited on pg. 11)

Hedeshy *et al.***(2021)** Ramin Hedeshy, Chandan Kumar, Raphael Menges and Steffen Staab. Hummer: Text entry by gaze and hum. In Yoshifumi Kitamura, Aaron Quigley, Katherine Isbister, Takeo Igarashi, Pernille Bjørn and Steven Mark Drucker, editors, *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, pgs. 741:1–741:11. ACM. doi: 10.1145/3411764.3445501. URL: https://doi.org/10.1145/3411764.3445501. (Cited on pg. 24)

Hennessey and Lawrence(2009) Craig Hennessey and Peter D. Lawrence. Noncontact binocular eye-gaze tracking for point-of-gaze estimation in three dimensions. *IEEE Trans. Biomed. Engineering*, 56(3):790–799. doi: 10.1109/TBME.2008.2005943. URL: https://doi.org/10.1109/TBME.2008.2005943. (Cited on pgs. 21, 22, and 58)

Hennessey *et al.***(2006)** Craig Hennessey, Borna Noureddin and Peter D. Lawrence. A single camera eye-gaze tracking system with free head motion. In Kari-Jouko Räihä and Andrew T. Duchowski, editors, *Proceedings of the Eye Tracking Research & Application Symposium, ETRA 2006, San Diego, California, USA, March 27-29, 2006*, pgs. 87–94. ACM. doi: 10.1145/1117309.1117349. URL: https://doi.org/10.1145/1117309.1117349. (Cited on pg. 20)

Hirzle *et al.***(2018)** Teresa Hirzle, Jan Gugenheimer, Florian Geiselhart, Andreas Bulling and Enrico Rukzio. Towards a symbiotic human-machine depth sensor: Exploring 3d gaze for object reconstruction. In Patrick Baudisch, Albrecht Schmidt and Andy Wilson, editors, *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings, UIST 2018, Berlin, Germany, October 14-17, 2018*, pgs. 114–116. ACM. doi: 10.1145/3266037.3266119. URL: https://doi.org/10.1145/3266037.3266119. (Cited on pg. 28)

Hirzle *et al.***(2019)** Teresa Hirzle, Jan Gugenheimer, Florian Geiselhart, Andreas Bulling and Enrico Rukzio. A design space for gaze interaction on head-mounted displays. In Stephen A. Brewster, Geraldine Fitzpatrick, Anna L. Cox and Vassilis Kostakos, editors, *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*, pg. 625. ACM. doi: 10.1145/3290605.3300855. URL: https://doi.org/10.1145/3290605.3300855. (Cited on pgs. 27 and 28)

Hochreiter and Schmidhuber(1997) Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780. doi: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735. (Cited on pg. 13)

Holland and Komogortsev(2013) Corey D Holland and Oleg V Komogortsev. Complex eye movement pattern biometrics: Analyzing fixations and saccades. In *2013 International conference on biometrics (ICB)*, pgs. 1–8. IEEE. (Cited on pg. 18)

Höllerer and Feiner(2004) Tobias Höllerer and Steve Feiner. Mobile augmented reality. *Telegeoinformatics: Location-based computing and services*, 21. (Cited on pg. 26)

Hooge *et al.***(2018)** Ignace Hooge, Diederick C Niehorster, Marcus Nyström, Richard Andersson and Roy S Hessels. Is human classification by experienced untrained observers a gold standard in fixation detection? *Behavior Research Methods*, 50(5):1864–1881. ISSN 1554-3528. doi: 10.3758/s13428-017-0955-x. Copyright 2017. Published by Elsevier Ltd. (Cited on pgs. 35 and 43)

Hoppe and Bulling(2016) Sabrina Hoppe and Andreas Bulling. End-to-end eye movement detection using convolutional neural networks. *CoRR*, abs/1609.02452. URL: http://arxiv.org/abs/1609.02452. (Cited on pgs. 18 and 43)

Howard and Rogers(1995) Ian P Howard and Brian J Rogers. *Binocular vision and stereopsis*. Oxford University Press, USA. (Cited on pg. 5)

Huang *et al.*(2019) Michael Xuelin Huang, Jiajia Li, Grace Ngai, Hong Va Leong and Andreas Bulling. Moment-to-moment detection of internal thought during video viewing from eye vergence behavior. In Laurent Amsaleg, Benoit Huet, Martha A. Larson, Guillaume Gravier, Hayley Hung, Chong-Wah Ngo and Wei Tsang Ooi, editors, *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*, pgs. 2254–2262. ACM. doi: 10.1145/3343031.3350573. URL: https://doi.org/10.1145/3343031.3350573. (Cited on pg. 28)

Hutchinson *et al.*(1989) Thomas E Hutchinson, K Preston White, Worthy N Martin, Kelly C Reichert and Lisa A Frey. Human-computer interaction using eye-gaze input. *IEEE Transactions on systems, man, and cybernetics*, 19(6):1527–1534. (Cited on pg. 24)

Ishiguro *et al.*(2010) Yoshio Ishiguro, Adiyan Mujibiya, Takashi Miyaki and Jun Rekimoto. Aided eyes: eye activity sensing for daily life. In *Proceedings of the 1st Augmented Human International Conference*, AH '10, pgs. 25:1–25:7, New York, NY, USA. ACM. ISBN 978-1-60558-825-4. doi: 10.1145/1785455.1785480. URL: http://doi.acm.org/10.1145/1785455.1785480. (Cited on pgs. 1 and 25)

Ishii and Ullmer(1997) Hiroshi Ishii and Brygg Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In Steven Pemberton, editor, *Human Factors in Computing Systems, CHI '97 Conference Proceedings, Atlanta, Georgia, USA, March 22-27, 1997*, pgs. 234–241. ACM/Addison-Wesley. doi: 10.1145/258549.258715. URL: https://doi.org/10.1145/258549.258715. (Cited on pg. 29)

Isokoski(2000) Poika Isokoski. Text input methods for eye trackers using off-screen targets. In Andrew T. Duchowski, editor, *Proceedings of the Eye Tracking Research & Application Symposium, ETRA 2000, Palm Beach Gardens, Florida, USA, November 6-8, 2000*, pgs. 15–21. ACM. doi: 10.1145/355017.355020. URL: https://doi.org/10.1145/355017.355020. (Cited on pg. 11)

Isokoski and Martin(2006) Poika Isokoski and Benoît Martin. Eye tracker input in first person shooter games. In *Proceedings of the 2nd Conference on Communication by Gaze Interaction: Communication by Gaze Interaction-COGAIN 2006: Gazing into the Future*, pgs. 78–81. (Cited on pg. 24)

Istance *et al.*(2010) Howell O. Istance, Aulikki Hyrskykari, Lauri Immonen, Santtu Mansikkamaa and Stephen Vickers. Designing gaze gestures for gaming: an investigation of performance. In Carlos Hitoshi Morimoto, Howell O. Istance, Aulikki Hyrskykari and Qiang Ji, editors, *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA 2010, Austin, Texas, USA, March 22-24, 2010*, pgs. 323–330. ACM. doi: 10.1145/1743666.1743740. URL: https://doi.org/10.1145/1743666.1743740. (Cited on pgs. 3 and 24)

Itoh and Klinker(2014) Yuta Itoh and Gudrun Klinker. Interaction-free calibration for optical see-through head-mounted displays based on 3d eye localization. In *IEEE*

Symposium on 3D User Interfaces, 3DUI 2014, Minneapolis, MN, USA, March 29-30, 2014, pgs. 75–82. doi: 10.1109/3DUI.2014.6798846. URL: https://doi.org/10.1109/3DUI.2014.6798846. (Cited on pg. 22)

Jacob and Karn(2003) RJ Jacob and Keith S Karn. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *Mind*, 2(3):4. (Cited on pg. 23)

Jacob(1991) Robert J. K. Jacob. The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Trans. Inf. Syst.*, 9(2):152–169. doi: 10.1145/123078.128728. URL: http://doi.acm.org/10.1145/123078.128728. (Cited on pgs. 1, 11, 23, 24, 28, 72, and 80)

Jacob(1995) Robert JK Jacob. Eye tracking in advanced interface design. *Virtual environments and advanced interface design*, pgs. 258–288. (Cited on pg. 23)

Ji and Zhu(2002) Qiang Ji and Zhiwei Zhu. Eye and gaze tracking for interactive graphic display. In Andreas Butz, Antonio Krüger, Patrick Olivier, Stefan Schlechtweg and Michelle X. Zhou, editors, *Proceedings of the 2nd International Symposium on Smart Graphics, Smart Graphics 2002, Hawthorne, New York, USA, June 11-13, 2002*, pgs. 79–85. ACM. doi: 10.1145/569005.569017. URL: https://doi.org/10.1145/569005.569017. (Cited on pg. 20)

Just and Carpenter(1976) Marcel Adam Just and Patricia A Carpenter. The role of eye-fixation research in cognitive psychology. *Behavior Research Methods & Instrumentation*, 8(2):139–143. (Cited on pg. 84)

Kassner *et al.*(2014) Moritz Kassner, William Patera and Andreas Bulling. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication*, pgs. 1151–1160. (Cited on pgs. 28, 40, 52, and 89)

Kellnhofer *et al.*(2016) Petr Kellnhofer, Piotr Didyk, Karol Myszkowski, Mohamed Hefeeda, Hans-Peter Seidel and Wojciech Matusik. Gazestereo3d: seamless disparity manipulations. *ACM Trans. Graph.*, 35(4):68:1–68:13. doi: 10.1145/2897824.2925866. URL: https://doi.org/10.1145/2897824.2925866. (Cited on pg. 27)

Khamis *et al.*(2018) Mohamed Khamis, Carl Oechsner, Florian Alt and Andreas Bulling. Vrpursuits: interaction in virtual reality using smooth pursuit eye movements. In Tiziana Catarci, Kent L. Norman and Massimo Mecella, editors, *Proceedings of the 2018 International Conference on Advanced Visual Interfaces, AVI 2018, Castiglione della Pescaia, Italy, May 29 - June 01, 2018*, pgs. 18:1–18:8. ACM. doi: 10.1145/3206505.3206522. URL: https://doi.org/10.1145/3206505.3206522. (Cited on pg. 26)

Kingma and Ba(2015) Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. URL: http://arxiv.org/abs/1412.6980. (Cited on pg. 33)

Kirst and Bulling(2016) Dominik Kirst and Andreas Bulling. On the verge: Voluntary convergences for accurate and precise timing of gaze input. In *Proceedings of the 2016*

CHI Conference Extended Abstracts on Human Factors in Computing Systems, CHI EA '16, pg. 1519–1525, New York, NY, USA. Association for Computing Machinery. ISBN 9781450340823. doi: 10.1145/2851581.2892307. URL: https://doi.org/10.1145/2851581.2892307. (Cited on pg. 28)

Kitajima *et al.*(2015) Yuki Kitajima, Sei Ikeda and Kosuke Sato. Vergence-based ax-ray vision. In Christian Sandor, Robert W. Lindeman, Walterio W. Mayol-Cuevas, Nobuchika Sakata, Richard A. Newcombe and Veronica Teichrieb, editors, *2015 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2015, Fukuoka, Japan, September 29 - Oct. 3, 2015*, pgs. 188–189. IEEE Computer Society. doi: 10.1109/ISMAR.2015.58. URL: https://doi.org/10.1109/ISMAR.2015.58. (Cited on pgs. 25, 28, 87, and 93)

Kollmorgen and Holmqvist(2007) Sepp Kollmorgen and Kenneth Holmqvist. Automatically detecting reading in eye tracking data. *Lund University Cognitive Studies*. (Cited on pg. 49)

Komogortsev and Karpov(2013) Oleg V. Komogortsev and Alex Karpov. Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades. *Behavior Research Methods*, 45(1):203–215. ISSN 1554-3528. doi: 10.3758/s13428-012-0234-9. URL: https://doi.org/10.3758/s13428-012-0234-9. (Cited on pg. 18)

Komogortsev and Khan(2007) Oleg V. Komogortsev and Javed I. Khan. Kalman filtering in the design of eye-gaze-guided computer interfaces. In *Proceedings of the 12th International Conference on Human-computer Interaction: Intelligent Multimodal Interaction Environments*, HCI'07, pgs. 679–689, Berlin, Heidelberg. Springer-Verlag. ISBN 978-3-540-73108-5. URL: http://dl.acm.org/citation.cfm?id=1769590.1769667. (Cited on pg. 18)

Komogortsev and Khan(2009) Oleg V. Komogortsev and Javed I. Khan. Eye movement prediction by oculomotor plant kalman filter with brainstem control. *Journal of Control Theory and Applications*, 7(1):14–22. ISSN 1993-0623. doi: 10.1007/s11768-009-7218-z. URL: https://doi.org/10.1007/s11768-009-7218-z. (Cited on pg. 18)

Konrad *et al.*(2020) Robert Konrad, Anastasios Angelopoulos and Gordon Wetzstein. Gaze-contingent ocular parallax rendering for virtual reality. *ACM Trans. Graph.*, 39(2): 10:1–10:12. doi: 10.1145/3361330. URL: https://doi.org/10.1145/3361330. (Cited on pg. 27)

Koochaki and Najafizadeh(2018) Fatemeh Koochaki and Laleh Najafizadeh. Predicting intention through eye gaze patterns. In *2018 IEEE Biomedical Circuits and Systems Conference, BioCAS 2018, Cleveland, OH, USA, October 17-19, 2018*, pgs. 1–4. IEEE. doi: 10.1109/BIOCAS.2018.8584665. URL: https://doi.org/10.1109/BIOCAS.2018.8584665. (Cited on pg. 17)

Krafka *et al.*(2016) Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra M. Bhandarkar, Wojciech Matusik and Antonio Torralba. Eye tracking for everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pgs. 2176–2184. IEEE Computer Society. doi: 10.1109/CVPR.2016.239. URL: https://doi.org/10.1109/CVPR.2016.239. (Cited on pgs. 9 and 21)

Krajancich *et al.*(2020) Brooke Krajancich, Petr Kellnhofer and Gordon Wetzstein. Optimizing depth perception in virtual and augmented reality through gaze-contingent

stereo rendering. *ACM Trans. Graph.*, 39(6):269:1–269:10. doi: 10.1145/3414685.3417820. URL: https://doi.org/10.1145/3414685.3417820. (Cited on pg. 27)

Kudo *et al.*(2013) Shinya Kudo, Hiroyuki Okabe, Taku Hachisu, Michi Sato, Shogo Fukushima and Hiroyuki Kajimoto. Input method using divergence eye movement. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pg. 1335–1340, New York, NY, USA. Association for Computing Machinery. ISBN 9781450319522. doi: 10.1145/2468356.2468594. URL: https://doi.org/10.1145/2468356.2468594. (Cited on pg. 28)

Kumar and Winograd(2007) Manu Kumar and Terry Winograd. Guide: gaze-enhanced UI design. In Mary Beth Rosson and David J. Gilmore, editors, *Extended Abstracts Proceedings of the 2007 Conference on Human Factors in Computing Systems, CHI 2007, San Jose, California, USA, April 28 - May 3, 2007*, pgs. 1977–1982. ACM. doi: 10.1145/1240866.1240935. URL: https://doi.org/10.1145/1240866.1240935. (Cited on pg. 24)

Kunze *et al.*(2013) Kai Kunze, Yuzuko Utsumi, Yuki Shiga, Koichi Kise and Andreas Bulling. I know what you are reading: recognition of document types using mobile eye tracking. In Kristof Van Laerhoven, Daniel Roggen, Daniel Gatica-Perez and Masaaki Fukumoto, editors, *Proceedings of the 17th Annual International Symposium on Wearable Computers. ISWC 2013, Zurich, Switzerland, September 8-12, 2013*, pgs. 113–116. ACM. doi: 10.1145/2493988.2494354. URL: https://doi.org/10.1145/2493988.2494354. (Cited on pgs. 1 and 18)

Kurauchi *et al.*(2016) Andrew T. N. Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto and Margrit Betke. Eyeswipe: Dwell-free text entry using gaze paths. In Jofish Kaye, Allison Druin, Cliff Lampe, Dan Morris and Juan Pablo Hourcade, editors, *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016*, pgs. 1952–1956. ACM. doi: 10.1145/2858036.2858335. URL: https://doi.org/10.1145/2858036.2858335. (Cited on pg. 24)

Kurauchi and Morimoto(2013) Andrew Toshiaki Kurauchi and Carlos Hitoshi Morimoto. Towards wearable gaze supported augmented cognition. In *Proc. CHI*, pgs. 91–96. (Cited on pg. 25)

Kuroda *et al.*(2010) Yoshihiro Kuroda, Yuta Morishita, Yasushi Masuda, Tomohiro Kuroda and Osamu Oshiro. Error reduction in 3d gaze point estimation for advanced medical annotations. In Olaf Dössel and Wolfgang C. Schlegel, editors, *World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany*, pgs. 2117–2119, Berlin, Heidelberg. Springer Berlin Heidelberg. ISBN 978-3-642-03882-2. (Cited on pg. 28)

Kwon *et al.*(2006) Yong-Moo Kwon, Kyeong-Won Jeon, Jeongseok Ki, Qonita M. Shahab, Sangwoo Jo and Sung-Kyu Kim. 3d gaze estimation and interaction to stereo dispaly. *IJVR*, 5(3):41–45. URL: http://www.ijvr.org/sub/issues/issue3/16-1394-KIST-YMKWON-20061021.pdf. (Cited on pgs. 22 and 64)

Kytö *et al.*(2018) Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee and Mark Billinghurst. Pinpointing: Precise head- and eye-based target selection for augmented reality. In Regan L. Mandryk, Mark Hancock, Mark Perry and Anna L. Cox,

editors, *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, Montreal, QC, Canada, April 21-26, 2018*, pg. 81. ACM. doi: 10.1145/3173574.3173655. URL: https://doi.org/10.1145/3173574.3173655. (Cited on pgs. 3 and 26)

Lander *et al.*(2015) Christian Lander, Sven Gehring, Antonio Krüger, Sebastian Boring and Andreas 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, UIST 2015, Charlotte, NC, USA, November 8-11, 2015*, pgs. 395–404. doi: 10.1145/2807442.2807479. URL: http://doi.acm.org/10.1145/2807442.2807479. (Cited on pg. 22)

Larsson *et al.*(2015) Linnéa Larsson, Marcus Nyström, Richard Andersson and Martin Stridh. Detection of fixations and smooth pursuit movements in high-speed eye-tracking data. *Biomedical Signal Processing and Control*, 18:145 – 152. ISSN 1746-8094. doi: 10.1016/j.bspc.2014.12.008. URL: http://www.sciencedirect.com/science/article/pii/S1746809414002031. (Cited on pg. 18)

LaViola Jr *et al.*(2017) Joseph J LaViola Jr, Ernst Kruijff, Ryan P McMahan, Doug Bowman and Ivan P Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley Professional. (Cited on pg. 92)

Lee(1999) Choongkil Lee. Eye and head coordination in reading: roles of head movement and cognitive control. *Vision Research*, 39(22):3761–3768. (Cited on pg. 49)

Lee *et al.*(2014) Jae-Young Lee, Hyung-Min Park, Seok-Han Lee, Soon-Ho Shin, Tae-eun Kim and Jong-Soo Choi. Design and implementation of an augmented reality system using gaze interaction. *Multimedia Tools Appl.*, 68(2):265–280. doi: 10.1007/s11042-011-0944-5. URL: https://doi.org/10.1007/s11042-011-0944-5. (Cited on pg. 87)

Lee *et al.*(2017) Youngho Lee, Choonsung Shin, Alexander Plopski, Yuta Itoh, Thammathip Piumsomboon, Arindam Dey, Gun A. Lee, Seungwon Kim and Mark Billinghurst. Estimating gaze depth using multi-layer perceptron. In *2017 International Symposium on Ubiquitous Virtual Reality, ISUVR 2017, Nara, Japan, June 27-29, 2017*, pgs. 26–29. IEEE. doi: 10.1109/ISUVR.2017.13. URL: https://doi.org/10.1109/ISUVR.2017.13. (Cited on pgs. 22 and 66)

Li and Li(2014) Jianfeng Li and Shigang Li. Eye-model-based gaze estimation by RGB-D camera. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2014, Columbus, OH, USA, June 23-28, 2014*, pgs. 606–610. doi: 10.1109/CVPRW.2014.93. URL: https://doi.org/10.1109/CVPRW.2014.93. (Cited on pg. 22)

Li *et al.*(2019) Yuan Li, Feiyu Lu, Wallace Santos Lages and Doug A. Bowman. Gaze direction visualization techniques for collaborative wide-area model-free augmented reality. In Christoph W. Borst, Arun K. Kulshreshth, Gerd Bruder, Stefania Serafin, Christian Sandor, Kyle Johnsen, Jinwei Ye, Daniel Roth and Sungchul Jung, editors, *Symposium on Spatial User Interaction, SUI 2019, New Orleans, LA, USA, October 19-20, 2019*, pgs. 11:1–11:11. ACM. doi: 10.1145/3357251.3357583. URL: https://doi.org/10.1145/3357251.3357583. (Cited on pg. 27)

Lidegaard *et al.*(2014) Morten Lidegaard, Dan Witzner Hansen and Norbert Krüger. Head mounted device for point-of-gaze estimation in three dimensions. In Pernilla Qvarfordt

and Dan Witzner Hansen, editors, *Eye Tracking Research and Applications, ETRA '14, Safety Harbor, FL, USA, March 26-28, 2014*, pgs. 83–86. ACM. doi: 10.1145/2578153.2578163. URL: https://doi.org/10.1145/2578153.2578163. (Cited on pgs. 21 and 58)

Luca *et al.*(2007) Alexander De Luca, Roman Weiss and Heiko Drewes. Evaluation of eye-gaze interaction methods for security enhanced pin-entry. In Bruce Thomas, editor, *Proceedings of the 2007 Australasian Computer-Human Interaction Conference, OZCHI 2007, Adelaide, Australia, November 28-30, 2007*, volume 251 of *ACM International Conference Proceeding Series*, pgs. 199–202. ACM. doi: 10.1145/1324892.1324932. URL: https://doi.org/10.1145/1324892.1324932. (Cited on pg. 24)

Luebke and Hallen(2001) David P. Luebke and Benjamin Hallen. Perceptually-driven simplification for interactive rendering. In Steven J. Gortler and Karol Myszkowski, editors, *Proceedings of the 12th Eurographics Workshop on Rendering Techniques, London, UK, June 25-27, 2001*, Eurographics, pgs. 223–234. Springer. doi: 10.1007/978-3-7091-6242-2_21. URL: https://doi.org/10.1007/978-3-7091-6242-2_21. (Cited on pg. 24)

Lystbæk *et al.*(2022a) Mathias N. Lystbæk, Ken Pfeuffer, Jens Emil Sloth Grønbæk and Hans Gellersen. Exploring gaze for assisting freehand selection-based text entry in AR. *Proc. ACM Hum. Comput. Interact.*, 6(ETRA):141:1–141:16. doi: 10.1145/3530882. URL: https://doi.org/10.1145/3530882. (Cited on pg. 24)

Lystbæk *et al.*(2022b) Mathias N. Lystbæk, Peter Rosenberg, Ken Pfeuffer, Jens Emil Grønbæk and Hans Gellersen. Gaze-hand alignment: Combining eye gaze and mid-air pointing for interacting with menus in augmented reality. *Proc. ACM Hum. Comput. Interact.*, 6(ETRA):145:1–145:18. doi: 10.1145/3530886. URL: https://doi.org/10.1145/3530886. (Cited on pg. 26)

MacKenzie and Zhang(2008) I. Scott MacKenzie and Xuang Zhang. Eye typing using word and letter prediction and a fixation algorithm. In Kari-Jouko Räihä and Andrew T. Duchowski, editors, *Proceedings of the Eye Tracking Research & Application Symposium, ETRA 2008, Savannah, Georgia, USA, March 26-28, 2008*, pgs. 55–58. ACM. doi: 10.1145/1344471.1344484. URL: https://doi.org/10.1145/1344471.1344484. (Cited on pgs. 17 and 24)

Majaranta and Räihä(2002) Päivi Majaranta and Kari-Jouko Räihä. Twenty years of eye typing: systems and design issues. In Andrew T. Duchowski, Roel Vertegaal and John W. Senders, editors, *Proceedings of the Eye Tracking Research & Application Symposium, ETRA 2002, New Orleans, Louisiana, USA, March 25-27, 2002*, pgs. 15–22. ACM. doi: 10.1145/507072.507076. URL: https://doi.org/10.1145/507072.507076. (Cited on pgs. 3, 11, 24, 29, 69, 85, and 97)

Mann(1997a) Steve Mann. Wearable computing: A first step toward personal imaging. *IEEE Computer*, 30(2):25–32. doi: 10.1109/2.566147. URL: http://dx.doi.org/10.1109/2.566147. (Cited on pg. 1)

Mann(1997b) Steve Mann. Introduction: On the bandwagon or beyond wearable computing? *Personal Technologies*, 1(4):203–207. (Cited on pg. 3)

Mansouryar *et al.*(2016) Mohsen Mansouryar, Julian Steil, Yusuke Sugano and Andreas Bulling. 3d gaze estimation from 2d pupil positions on monocular head-mounted eye trackers. In Pernilla Qvarfordt and Dan Witzner Hansen, editors, *Proceedings of the*

Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, ETRA 2016, Charleston, SC, USA, March 14-17, 2016, pgs. 197–200. ACM. doi: 10.1145/2857491.2857530. URL: https://doi.org/10.1145/2857491.2857530. (Cited on pg. 92)

Mardanbegi *et al.***(2012)** D. Mardanbegi, D. W. Hansen and T. Pederson. Eye-based head gestures. In *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA'12*, pgs. 139–146, Santa Barbara, CA. ACM Press. (Cited on pgs. 24 and 29)

Mardanbegi *et al.***(2019)** D. Mardanbegi, B. Mayer, K. Pfeuffer, S. Jalaliniya, H. Gellersen and A. Perzl. Eyeseethrough: Unifying tool selection and application in virtual environments. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pgs. 474–483. (Cited on pg. 87)

Mardanbegi and Hansen(2011) Diako Mardanbegi and Dan Witzner Hansen. Mobile gaze-based screen interaction in 3d environments. In Veronica Sundstedt and Charlotte C. Sennersten, editors, *NGCA 2011, First Conference on Novel Gaze-Controlled Applications, Karlskrona, Sweden, May 26 - 27, 2011*, pg. 2. ACM. doi: 10.1145/1983302.1983304. URL: https://doi.org/10.1145/1983302.1983304. (Cited on pg. 21)

Mardanbegi and Hansen(2012) Diako Mardanbegi and Dan Witzner Hansen. Parallax error in the monocular head-mounted eye trackers. In Anind K. Dey, Hao-Hua Chu and Gillian R. Hayes, editors, *The 2012 ACM Conference on Ubiquitous Computing, Ubicomp '12, Pittsburgh, PA, USA, September 5-8, 2012*, pgs. 689–694. ACM. doi: 10.1145/2370216.2370366. URL: https://doi.org/10.1145/2370216.2370366. (Cited on pgs. 2, 20, and 22)

Maruyama *et al.***(2016)** Hirotaka Maruyama, Yuta Saito and Mitsuho Yamada. An analysis of changes in attention based on miniature eye movements. In *11th International Conference on Computer Science & Education, ICCSE 2016, Nagoya, Japan, August 23-25, 2016*, pgs. 539–543. IEEE. doi: 10.1109/ICCSE.2016.7581638. URL: https://doi.org/10.1109/ICCSE.2016.7581638. (Cited on pg. 17)

Matin(1974) Ethel Matin. Saccadic suppression: a review and an analysis. *Psychological bulletin*, 81(12):899. (Cited on pg. 6)

Mauderer *et al.***(2014)** Michael Mauderer, Simone Conte, Miguel A. Nacenta and Dhanraj Vishwanath. Depth perception with gaze-contingent depth of field. In Matt Jones, Philippe A. Palanque, Albrecht Schmidt and Tovi Grossman, editors, *CHI Conference on Human Factors in Computing Systems, CHI'14, Toronto, ON, Canada - April 26 - May 01, 2014*, pgs. 217–226. ACM. doi: 10.1145/2556288.2557089. URL: https://doi.org/10.1145/2556288.2557089. (Cited on pg. 27)

McMurrough *et al.***(2012)** Christopher McMurrough, Christopher Conly, Vassilis Athitsos and Fillia Makedon. 3d point of gaze estimation using head-mounted RGB-D cameras. In *The 14th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS '12, Boulder, CO, USA, October 22 - 24, 2012*, pgs. 283–284. doi: 10.1145/2384916.2384994. URL: http://doi.acm.org/10.1145/2384916.2384994. (Cited on pgs. 20 and 22)

Meyer *et al.***(2022)** Johannes Meyer, Tobias Wilm, Reinhold Fiess, Thomas Schlebusch, Wilhelm Stork and Enkelejda Kasneci. A holographic single-pixel stereo camera sensor for calibration-free eye-tracking in retinal projection augmented reality glasses. In Frederick Shic, Enkelejda Kasneci, Mohamed Khamis, Hans Gellersen, Krzysztof Krejtz,

Daniel Weiskopf, Tanja Blascheck, Jessica Bradshaw, Hana Vrzakova, Kamran Binaee, Michael Burch, Peter Kiefer, Roman Bednarik, Diako Mardanbegi, Christopher Clarke, Rakshit Sunil Kothari, Vijay Rajanna, Sampath Jayarathna, Arantxa Villanueva, Adham Atyabi and Shahram Eivazi, editors, *ETRA 2022: Symposium on Eye Tracking Research and Applications, Seattle, WA, USA, June 8 - 11, 2022*, pgs. 20:1–20:7. ACM. doi: 10.1145/3517031.3529616. URL: https://doi.org/10.1145/3517031.3529616. (Cited on pg. 22)

Miller(1968) Robert B. Miller. Response time in man-computer conversational transactions. In *American Federation of Information Processing Societies: Proceedings of the AFIPS '68 Fall Joint Computer Conference, December 9-11, 1968, San Francisco, California, USA - Part I*, volume 33 of *AFIPS Conference Proceedings*, pgs. 267–277. AFIPS / ACM / Thomson Book Company, Washington D.C. doi: 10.1145/1476589.1476628. URL: https://doi.org/10.1145/1476589.1476628. (Cited on pg. 42)

Miniotas(2000) Darius Miniotas. Application of fitts' law to eye gaze interaction. In Marilyn Tremaine, editor, *CHI '00 Extended Abstracts on Human Factors in Computing Systems, CHI Extended Abstracts '00, The Hague, The Netherlands, April 1-6, 2000*, pgs. 339–340. ACM. doi: 10.1145/633292.633496. URL: https://doi.org/10.1145/633292.633496. (Cited on pg. 84)

Miniotas *et al.***(2004)** Darius Miniotas, Oleg Spakov and I. Scott MacKenzie. Eye gaze interaction with expanding targets. In Elizabeth Dykstra-Erickson and Manfred Tscheligi, editors, *Extended abstracts of the 2004 Conference on Human Factors in Computing Systems, CHI 2004, Vienna, Austria, April 24 - 29, 2004*, pgs. 1255–1258. ACM. doi: 10.1145/985921.986037. URL: https://doi.org/10.1145/985921.986037. (Cited on pg. 24)

Miniotas *et al.***(2006)** Darius Miniotas, Oleg Spakov, Ivan Tugoy and I. Scott MacKenzie. Speech-augmented eye gaze interaction with small closely spaced targets. In Kari-Jouko Räihä and Andrew T. Duchowski, editors, *Proceedings of the Eye Tracking Research & Application Symposium, ETRA 2006, San Diego, California, USA, March 27-29, 2006*, pgs. 67–72. ACM. doi: 10.1145/1117309.1117345. URL: https://doi.org/10.1145/1117309.1117345. (Cited on pg. 24)

Mitsugami *et al.***(2003)** Ikuhisa Mitsugami, Norimichi Ukita and Masatsugu Kidode. Estimation of 3d gazed position using view lines. In *12th International Conference on Image Analysis and Processing (ICIAP 2003), 17-19 September 2003, Mantova, Italy*, pgs. 466–471. doi: 10.1109/ICIAP.2003.1234094. URL: https://doi.org/10.1109/ICIAP.2003.1234094. (Cited on pg. 22)

Molitor *et al.***(2015)** Robert J Molitor, Philip C Ko and Brandon A Ally. Eye movements in alzheimer's disease. *Journal of Alzheimer's disease: JAD*, 44(1):1. (Cited on pg. 18)

Møllenbach *et al.***(2010)** Emilie Møllenbach, Martin Lillholm, Alastair G. Gail and John Paulin Hansen. Single gaze gestures. In Carlos Hitoshi Morimoto, Howell O. Istance, Aulikki Hyrskykari and Qiang Ji, editors, *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA 2010, Austin, Texas, USA, March 22-24, 2010*, pgs. 177–180. ACM. doi: 10.1145/1743666.1743710. URL: https://doi.org/10.1145/1743666.1743710. (Cited on pg. 85)

Mora and Odobez(2014) Kenneth Alberto Funes Mora and Jean-Marc Odobez. Geometric generative gaze estimation (G3E) for remote RGB-D cameras. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pgs. 1773–1780. doi: 10.1109/CVPR.2014.229. URL: https://doi.org/10.1109/CVPR.2014.229. (Cited on pgs. 21 and 22)

Morales *et al.*(2018) Aythami Morales, Francisco M Costela, Ruben Tolosana and Russell L Woods. Saccade landing point prediction: A novel approach based on recurrent neural networks. In *Proceedings of the 2018 International Conference on Machine Learning Technologies*, pgs. 1–5. (Cited on pg. 27)

Morimoto and Amir(2010) Carlos Hitoshi Morimoto and Arnon Amir. Context switching for fast key selection in text entry applications. In Carlos Hitoshi Morimoto, Howell O. Istance, Aulikki Hyrskykari and Qiang Ji, editors, *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA 2010, Austin, Texas, USA, March 22-24, 2010*, pgs. 271–274. ACM. doi: 10.1145/1743666.1743730. URL: https://doi.org/10.1145/1743666.1743730. (Cited on pg. 85)

Morimoto and Mimica(2005) Carlos Hitoshi Morimoto and Marcio R. M. Mimica. Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98(1):4–24. doi: 10.1016/j.cviu.2004.07.010. URL: https://doi.org/10.1016/j.cviu.2004.07.010. (Cited on pg. 20)

Munn and Pelz(2008) Susan M. Munn and Jeff B. Pelz. 3d point-of-regard, position and head orientation from a portable monocular video-based eye tracker. In Kari-Jouko Räihä and Andrew T. Duchowski, editors, *Proceedings of the Eye Tracking Research & Application Symposium, ETRA 2008, Savannah, Georgia, USA, March 26-28, 2008*, pgs. 181–188. ACM. doi: 10.1145/1344471.1344517. URL: https://doi.org/10.1145/1344471.1344517. (Cited on pgs. 20 and 22)

Murphy *et al.*(2001) Paul J Murphy, Anne L Duncan, Alastair J Glennie and Paul C Knox. The effect of scleral search coil lens wear on the eye. *British journal of ophthalmology*, 85(3):332–335. (Cited on pg. 8)

Nacke *et al.*(2010) Lennart E Nacke, Sophie Stellmach, Dennis Sasse and Craig A Lindley. Gameplay experience in a gaze interaction game. *arXiv preprint arXiv:1004.0259*. (Cited on pg. 25)

Nakamura and Csikszentmihalyi(2014) J. Nakamura and M. Csikszentmihalyi. *Flow and the Foundations of Positive Psychology*, chapter The Concept of Flow. Springer. doi: https://doi.org/10.1007/978-94-017-9088-8_16. (Cited on pg. 80)

Nilsson *et al.*(2009) Susanna Nilsson, Torbjorn Gustafsson and Per Carleberg. Hands free interaction with virtual information in a real environment: Eye gaze as an interaction tool in an augmented reality system. *PsychNology J.*, 7(2):175–196. URL: http://www.psychnology.org/File/PNJ7%282%29/PSYCHNOLOGY_JOURNAL_7_2_NILSSON.pdf. (Cited on pgs. 25 and 87)

Nyström and Holmqvist(2010) Marcus Nyström and Kenneth Holmqvist. An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior research methods*, 42(1):188–204. (Cited on pg. 17)

Ohno(1998) Takehiko Ohno. Features of eye gaze interface for selection tasks. In *Third Asian Pacific Computer and Human Interaction, July 15-17, 1998, Kangawa, Japan, Proceedings*, pgs. 176–182. IEEE Computer Society. doi: 10.1109/APCHI.1998.704190. URL: https://doi.org/10.1109/APCHI.1998.704190. (Cited on pg. 74)

Orlosky *et al.***(2014)** Jason Orlosky, Takumi Toyama, Daniel Sonntag and Kiyoshi Kiyokawa. Using eye-gaze and visualization to augment memory - A framework for improving context recognition and recall. In Norbert A. Streitz and Panos Markopoulos, editors, *Distributed, Ambient, and Pervasive Interactions - Second International Conference, DAPI 2014, Held as Part of HCI Interational 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings*, volume 8530 of *Lecture Notes in Computer Science*, pgs. 282–291. Springer. doi: 10.1007/978-3-319-07788-8_27. URL: https://doi.org/10.1007/978-3-319-07788-8_27. (Cited on pg. 3)

Pai *et al.***(2016)** Yun Suen Pai, Benjamin Outram, Noriyasu Vontin and Kai Kunze. Transparent reality: Using eye gaze focus depth as interaction modality. In Jun Rekimoto, Takeo Igarashi, Jacob O. Wobbrock and Daniel Avrahami, editors, *Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST 2016 Adjunct Volume, Tokyo, Japan, October 16 - 19, 2016*, pgs. 171–172. ACM. doi: 10.1145/2984751.2984754. URL: https://doi.org/10.1145/2984751.2984754. (Cited on pgs. 25, 28, and 87)

Paletta *et al.***(2013)** Lucas Paletta, Katrin Santner, Gerald Fritz, Heinz Mayer and Johann Schrammel. 3d attention: measurement of visual saliency using eye tracking glasses. In Wendy E. Mackay, Stephen A. Brewster and Susanne Bødker, editors, *2013 ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '13, Paris, France, April 27 - May 2, 2013, Extended Abstracts*, pgs. 199–204. ACM. doi: 10.1145/2468356.2468393. URL: https://doi.org/10.1145/2468356.2468393. (Cited on pg. 22)

Park *et al.***(2011)** Gie-seo Park, Jong-gil Ahn and Gerard Jounghyun Kim. Gaze-directed hands-free interface for mobile interaction. In Julie A. Jacko, editor, *Human-Computer Interaction. Interaction Techniques and Environments - 14th International Conference, HCI International 2011, Orlando, FL, USA, July 9-14, 2011, Proceedings, Part II*, volume 6762 of *Lecture Notes in Computer Science*, pgs. 304–313. Springer. doi: 10.1007/978-3-642-21605-3_34. URL: https://doi.org/10.1007/978-3-642-21605-3_34. (Cited on pg. 24)

Park *et al.***(2008)** Hyung-Min Park, Seok-Han Lee and Jong-Soo Choi. Wearable augmented reality system using gaze interaction. In *7th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2008, Cambridge, UK, 15-18th September 2008*, pgs. 175–176. IEEE Computer Society. doi: 10.1109/ISMAR.2008.4637353. URL: https://doi.org/10.1109/ISMAR.2008.4637353. (Cited on pgs. 25 and 87)

Pathmanathan *et al.***(2020)** Nelusa Pathmanathan, Michael Becher, Nils Rodrigues, Guido Reina, Thomas Ertl, Daniel Weiskopf and Michael Sedlmair. Eye vs. head: Comparing gaze methods for interaction in augmented reality. In Andreas Bulling, Anke Huckauf, Eakta Jain, Ralph Radach and Daniel Weiskopf, editors, *ETRA '20: 2020 Symposium on Eye Tracking Research and Applications, Short Papers, Stuttgart, Germany, June 2-5, 2020*, pgs. 50:1–50:5. ACM. doi: 10.1145/3379156.3391829. URL: https://doi.org/10.1145/3379156.3391829. (Cited on pg. 26)

Penkar *et al.***(2012)** Abdul Moiz Penkar, Christof Lutteroth and Gerald Weber. Designing for the eye: design parameters for dwell in gaze interaction. In Vivienne Farrell, Graham Farrell, Caslon Chua, Weidong Huang, Rajesh Vasa and Clinton Woodward, editors, *The 24th Australian Computer-Human Interaction Conference, OzCHI '12, Melbourne, VIC, Australia - November 26 - 30, 2012*, pgs. 479–488. ACM. doi: 10.1145/2414536.2414609. URL: https://doi.org/10.1145/2414536.2414609. (Cited on pgs. 24 and 29)

Pfeiffer and Renner(2014) Thies Pfeiffer and Patrick Renner. Eyesee3d: a low-cost approach for analyzing mobile 3d eye tracking data using computer vision and augmented reality technology. In *Eye Tracking Research and Applications, ETRA '14, Safety Harbor, FL, USA, March 26-28, 2014*, pgs. 195–202. doi: 10.1145/2578153.2578183. URL: http://doi.acm.org/10.1145/2578153.2578183. (Cited on pg. 22)

Pfeiffer *et al.***(2008)** Thies Pfeiffer, Marc Erich Latoschik and Ipke Wachsmuth. Evaluation of binocular eye trackers and algorithms for 3d gaze interaction in virtual reality environments. *JVRB*, 5. URL: http://www.jvrb.org/past-issues/5.2008/1660. (Cited on pg. 28)

Pfeuffer *et al.***(2014)** Ken Pfeuffer, Jason Alexander, Ming Ki Chong and Hans Gellersen. Gaze-touch: combining gaze with multi-touch for interaction on the same surface. In Hrvoje Benko, Mira Dontcheva and Daniel Wigdor, editors, *The 27th Annual ACM Symposium on User Interface Software and Technology, UIST '14, Honolulu, HI, USA, October 5-8, 2014*, pgs. 509–518. ACM. doi: 10.1145/2642918.2647397. URL: https://doi.org/10.1145/2642918.2647397. (Cited on pgs. 24, 85, and 97)

Pfeuffer *et al.***(2015)** Ken Pfeuffer, Jason Alexander, Ming Ki Chong, Yanxia Zhang and Hans Gellersen. Gaze-shifting: Direct-indirect input with pen and touch modulated by gaze. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, UIST '15, pg. 373–383, New York, NY, USA. Association for Computing Machinery. ISBN 9781450337793. doi: 10.1145/2807442.2807460. URL: https://doi.org/10.1145/2807442.2807460. (Cited on pg. 85)

Pfeuffer *et al.***(2021)** Ken Pfeuffer, Yasmeen Abdrabou, Augusto Esteves, Radiah Rivu, Yomna Abdelrahman, Stefanie Meitner, Amr Saadi and Florian Alt. Artention: A design space for gaze-adaptive user interfaces in augmented reality. *Computers & Graphics*. (Cited on pg. 27)

Pirri *et al.***(2011)** Fiora Pirri, Matia Pizzoli and Alessandro Rudi. A general method for the point of regard estimation in 3d space. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pgs. 921–928. doi: 10.1109/CVPR.2011.5995634. URL: https://doi.org/10.1109/CVPR.2011.5995634. (Cited on pg. 22)

Piumsomboon *et al.***(2017)** Thammathip Piumsomboon, Gun A. Lee, Robert W. Lindeman and Mark Billinghurst. Exploring natural eye-gaze-based interaction for immersive virtual reality. In Maud Marchal, Robert J. Teather and Bruce H. Thomas, editors, *2017 IEEE Symposium on 3D User Interfaces, 3DUI 2017, Los Angeles, CA, USA, March 18-19, 2017*, pgs. 36–39. IEEE Computer Society. doi: 10.1109/3DUI.2017.7893315. URL: https://doi.org/10.1109/3DUI.2017.7893315. (Cited on pgs. 3 and 27)

Pouwelse *et al.***(2001)** Johan Pouwelse, Koen Langendoen and Henk Sips. Dynamic voltage scaling on a low-power microprocessor. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pgs. 251–259. ACM. (Cited on pg. 49)

Purves(2004) D. Purves. *Neuroscience*. The Japanese classic collection. Sinauer Associates. ISBN 9780878937257. URL: https://books.google.com.br/books?id=YZGDbwAACAAJ. (Cited on pgs. 5, 6, 7, and 9)

Qian and Teather(2017) Yuan Yuan Qian and Robert J. Teather. The eyes don't have it: an empirical comparison of head-based and eye-based selection in virtual reality. In Adalberto L. Simeone, Kyle Johnsen, Robert J. Teather and Christian Sandor, editors, *Proceedings of the 5th Symposium on Spatial User Interaction, SUI 2017, Brighton, United Kingdom, October 16 - 17, 2017*, pgs. 91–98. ACM. doi: 10.1145/3131277.3132182. URL: https://doi.org/10.1145/3131277.3132182. (Cited on pg. 26)

Ravanelli *et al.***(2018)** Mirco Ravanelli, Philemon Brakel, Maurizio Omologo and Yoshua Bengio. Light gated recurrent units for speech recognition. *IEEE Trans. Emerg. Top. Comput. Intell.*, 2(2):92–102. doi: 10.1109/TETCI.2017.2762739. URL: https://doi.org/10.1109/TETCI.2017.2762739. (Cited on pg. 14)

Rayner(1998) Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372. (Cited on pg. 6)

Rayner *et al.***(2001)** Keith Rayner, Barbara R Foorman, Charles A Perfetti, David Pesetsky and Mark S Seidenberg. How psychological science informs the teaching of reading. *Psychological science in the public interest*, 2(2):31–74. (Cited on pg. 49)

Reichle *et al.***(1998)** Erik D. Reichle, Alexander Pollatsek, Donald L. Fisher and Keith Rayner. Toward a model of eye movement control in reading. *Psychological review*, 105 (1):125. (Cited on pg. 49)

Rivu *et al.***(2020)** Radiah Rivu, Yasmeen Abdrabou, Ken Pfeuffer, Augusto Esteves, Stefanie Meitner and Florian Alt. Stare: Gaze-assisted face-to-face communication in augmented reality. In Andreas Bulling, Anke Huckauf, Eakta Jain, Ralph Radach and Daniel Weiskopf, editors, *ETRA '20 Adjunct: 2020 Symposium on Eye Tracking Research and Applications, Stuttgart, Germany, June 2-5, 2020, Adjunct Volume*, pgs. 14:1–14:5. ACM. doi: 10.1145/3379157.3388930. URL: https://doi.org/10.1145/3379157.3388930. (Cited on pgs. 26, 28, and 29)

Ruan *et al.***(2018)** Lingyan Ruan, Bin Chen and Miu-Ling Lam. Human-computer interaction by voluntary vergence control. In *SIGGRAPH Asia 2018 Posters*, SA '18, New York, NY, USA. Association for Computing Machinery. ISBN 9781450360630. doi: 10.1145/3283289.3283356. URL: https://doi.org/10.1145/3283289.3283356. (Cited on pg. 28)

Saffer(2013) Dan Saffer. *Microinteractions: Designing with Details*. O'Reilly Media, Inc. ISBN 144934268X, 9781449342685. (Cited on pg. 69)

Salvucci and Goldberg(2000) Dario D. Salvucci and Joseph H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ETRA '00, pgs. 71–78, New York, NY, USA.

ACM. ISBN 1-58113-280-8. doi: 10.1145/355017.355028. URL: http://doi.acm.org/10.1145/355017.355028. (Cited on pg. 17)

Sanches *et al.***(2017)** Charles Lima Sanches, Olivier Augereau and Koichi Kise. Using the eye gaze to predict document reading subjective understanding. In *1st International Workshop on Human-Document Interaction, 14th IAPR International Conference on Document Analysis and Recognition, HDI@ICDAR 2017, Kyoto, Japan, November 9-15, 2017*, pgs. 28–31. IEEE. doi: 10.1109/ICDAR.2017.377. URL: https://doi.org/10.1109/ICDAR.2017.377. (Cited on pg. 17)

Santini *et al.***(2016)** Thiago Santini, Wolfgang Fuhl, Thomas Kübler and Enkelejda Kasneci. Bayesian identification of fixations, saccades, and smooth pursuits. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ETRA '16, pgs. 163–170, New York, NY, USA. ACM. ISBN 978-1-4503-4125-7. doi: 10.1145/2857491.2857512. URL: http://doi.acm.org/10.1145/2857491.2857512. (Cited on pgs. 18, 19, and 42)

Sattar *et al.***(2015)** Hosnieh Sattar, Sabine Müller, Mario Fritz and Andreas Bulling. Prediction of search targets from fixations in open-world settings. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pgs. 981–990. IEEE Computer Society. doi: 10.1109/CVPR.2015.7298700. URL: https://doi.org/10.1109/CVPR.2015.7298700. (Cited on pg. 18)

Sauter *et al.***(1991)** D. Sauter, B. J. 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*, 29(1):63–69. ISSN 1741-0444. doi: 10.1007/BF02446297. URL: https://doi.org/10.1007/BF02446297. (Cited on pg. 18)

Schuetz *et al.***(2019)** Immo Schuetz, T. Scott Murdison, Kevin J. MacKenzie and Marina Zannoli. An explanation of fitts' law-like performance in gaze-based selection tasks using a psychophysics approach. In Stephen A. Brewster, Geraldine Fitzpatrick, Anna L. Cox and Vassilis Kostakos, editors, *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*, pg. 535. ACM. doi: 10.1145/3290605.3300765. URL: https://doi.org/10.1145/3290605.3300765. (Cited on pg. 84)

Sesma-Sanchez and Hansen(2018) Laura Sesma-Sanchez and Dan Witzner Hansen. Binocular model-based gaze estimation with a camera and a single infrared light source. In Bonita Sharif and Krzysztof Krejtz, editors, *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, ETRA 2018, Warsaw, Poland, June 14-17, 2018*, pgs. 47:1–47:5. ACM. doi: 10.1145/3204493.3204557. URL: https://doi.org/10.1145/3204493.3204557. (Cited on pg. 2)

Sesma-Sanchez *et al.***(2016)** Laura Sesma-Sanchez, Yanxia Zhang, Andreas Bulling and Hans Gellersen. Gaussian processes as an alternative to polynomial gaze estimation functions. In Pernilla Qvarfordt and Dan Witzner Hansen, editors, *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, ETRA 2016, Charleston, SC, USA, March 14-17, 2016*, pgs. 229–232. ACM. doi: 10.1145/2857491.2857509. URL: https://doi.org/10.1145/2857491.2857509. (Cited on pgs. 20 and 64)

Shih *et al.*(2000) Sheng-Wen Shih, Yu-Te Wu and Jin Liu. A calibration-free gaze tracking technique. In *15th International Conference on Pattern Recognition, ICPR'00, Barcelona, Spain, September 3-8, 2000*, pgs. 4201–4204. IEEE Computer Society. doi: 10.1109/ICPR.2000.902895. URL: https://doi.org/10.1109/ICPR.2000.902895. (Cited on pg. 20)

Sibert and Jacob(2000) Linda E. Sibert and Robert J. K. Jacob. Evaluation of eye gaze interaction. In Thea Turner and Gerd Szwillus, editors, *Proceedings of the CHI 2000 Conference on Human factors in computing systems, The Hague, The Netherlands, April 1-6, 2000*, pgs. 281–288. ACM. doi: 10.1145/332040.332445. URL: https://doi.org/10.1145/332040.332445. (Cited on pgs. 72 and 84)

Siewiorek(2002) Daniel P Siewiorek. New frontiers of application design. *Communications of the ACM*, 45(12):79–82. (Cited on pgs. 1 and 3)

Smith and Graham(2006) J. David Smith and T. C. Nicholas Graham. Use of eye movements for video game control. In Hiroshi Ishii, Newton Lee, Stéphane Natkin and Katsuhide Tsushima, editors, *Proceedings of the International Conference on Advances in Computer Entertainment Technology, ACE 2006, Hollywood, California, USA, June 14-16, 2006*, pg. 20. ACM. doi: 10.1145/1178823.1178847. URL: https://doi.org/10.1145/1178823.1178847. (Cited on pg. 24)

Spakov and Majaranta(2012) Oleg Spakov and Päivi Majaranta. Enhanced gaze interaction using simple head gestures. In Anind K. Dey, Hao-Hua Chu and Gillian R. Hayes, editors, *The 2012 ACM Conference on Ubiquitous Computing, Ubicomp '12, Pittsburgh, PA, USA, September 5-8, 2012*, pgs. 705–710. ACM. doi: 10.1145/2370216.2370369. URL: https://doi.org/10.1145/2370216.2370369. (Cited on pgs. 24 and 29)

St. John *et al.*(2004) Mark St. John, David A Kobus, Jeffrey G Morrison and Dylan Schmorrow. Overview of the darpa augmented cognition technical integration experiment. *International Journal of Human-Computer Interaction*, 17(2):131–149. (Cited on pgs. 24 and 69)

Stampe(1993) Dave M Stampe. Heuristic filtering and reliable calibration methods for video-based pupil-tracking systems. *Behavior Research Methods, Instruments, & Computers*, 25(2):137–142. (Cited on pg. 20)

Starner(1996) Thad Starner. Human-powered wearable computing. *IBM Syst. J.*, 35(3/4): 618–629. doi: 10.1147/sj.353.0618. URL: https://doi.org/10.1147/sj.353.0618. (Cited on pg. 2)

Starner(2001) Thad Starner. The challenges of wearable computing: Part 1. *Ieee Micro*, 21 (4):44–52. (Cited on pg. 2)

Startsev *et al.*(2019a) Mikhail Startsev, Ioannis Agtzidis and Michael Dorr. Characterizing and automatically detecting smooth pursuit in a large-scale ground-truth data set of dynamic natural scenes. *Journal of Vision*, 19(14):10–10. ISSN 1534-7362. doi: 10.1167/19.14.10. URL: https://doi.org/10.1167/19.14.10. (Cited on pg. 42)

Startsev *et al.*(2019b) Mikhail Startsev, Ioannis Agtzidis and Michael Dorr. 1d cnn with blstm for automated classification of fixations, saccades, and smooth pursuits. *Behavior Research Methods*, 51(2):556–572. ISSN 1554-3528. doi: 10.3758/s13428-018-1144-2. URL: https://doi.org/10.3758/s13428-018-1144-2. (Cited on pgs. vii, viii, 18, 32, 34, 35, and 41)

Startsev *et al.***(2019c)** Mikhail Startsev, Ioannis Agtzidis and Michael Dorr. Sequence-to-sequence deep learning for eye movement classification. In *PERCEPTION*, volume 48, pgs. 200–200. SAGE PUBLICATIONS LTD 1 OLIVERS YARD, 55 CITY ROAD, LONDON EC1Y 1SP, ENGLAND. (Cited on pg. 35)

Stengel *et al.***(2016)** Michael Stengel, Steve Grogorick, Martin Eisemann and Marcus A. Magnor. Adaptive image-space sampling for gaze-contingent real-time rendering. *Comput. Graph. Forum*, 35(4):129–139. doi: 10.1111/cgf.12956. URL: https://doi.org/10.1111/cgf.12956. (Cited on pg. 27)

Steptoe *et al.***(2009)** William Steptoe, Oyewole Oyekoya, Alessio Murgia, Robin Wolff, John Rae, Estefania Guimaraes, David J. Roberts and Anthony Steed. Eye tracking for avatar eye gaze control during object-focused multiparty interaction in immersive collaborative virtual environments. In *IEEE Virtual Reality Conference 2009 (VR 2009), 14-18 March 2009, Lafayette, Louisiana, USA, Proceedings*, pgs. 83–90. IEEE Computer Society. doi: 10.1109/VR.2009.4811003. URL: https://doi.org/10.1109/VR.2009.4811003. (Cited on pg. 25)

Sugano *et al.***(2013)** Yusuke Sugano, Yasuyuki Matsushita and Yoichi Sato. Appearance-based gaze estimation using visual saliency. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35 (2):329–341. doi: 10.1109/TPAMI.2012.101. URL: https://doi.org/10.1109/TPAMI.2012.101. (Cited on pg. 64)

Świrski and Dodgson(2013) Lech Świrski and Neil A. Dodgson. A fully-automatic, temporal approach to single camera, glint-free 3d eye model fitting [abstract]. In *Proceedings of ECEM 2013*. URL: http://www.cl.cam.ac.uk/research/rainbow/projects/eyemodelfit/. (Cited on pgs. 22, 61, and 90)

Takemura *et al.***(2010)** Kentaro Takemura, Yuji Kohashi, Tsuyoshi Suenaga, Jun Takamatsu and Tsukasa Ogasawara. Estimating 3d point-of-regard and visualizing gaze trajectories under natural head movements. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA 2010, Austin, Texas, USA, March 22-24, 2010*, pgs. 157–160. doi: 10.1145/1743666.1743705. URL: http://doi.acm.org/10.1145/1743666.1743705. (Cited on pgs. 20 and 22)

Taketomi *et al.***(2017)** Takafumi Taketomi, Hideaki Uchiyama and Sei Ikeda. Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Trans. Comput. Vis. Appl.*, 9:16. doi: 10.1186/s41074-017-0027-2. URL: https://doi.org/10.1186/s41074-017-0027-2. (Cited on pg. 23)

Tanriverdi and Jacob(2000) Vildan Tanriverdi and Robert J. K. Jacob. Interacting with eye movements in virtual environments. In Thea Turner and Gerd Szwillus, editors, *Proceedings of the CHI 2000 Conference on Human factors in computing systems, The Hague, The Netherlands, April 1-6, 2000*, pgs. 265–272. ACM. doi: 10.1145/332040.332443. URL: https://doi.org/10.1145/332040.332443. (Cited on pgs. 1, 25, and 92)

Toet(2006) Alexander Toet. Gaze directed displays as an enabling technology for attention aware systems. *Comput. Hum. Behav.*, 22(4):615–647. doi: 10.1016/j.chb.2005.12.010. URL: https://doi.org/10.1016/j.chb.2005.12.010. (Cited on pgs. 28 and 29)

Toyama *et al.***(2014)** Takumi Toyama, Daniel Sonntag, Andreas Dengel, Takahiro Matsuda, Masakazu Iwamura and Koichi Kise. A mixed reality head-mounted text translation

system using eye gaze input. In Tsvi Kuflik, Oliviero Stock, Joyce Yue Chai and Antonio Krüger, editors, *19th International Conference on Intelligent User Interfaces, IUI 2014, Haifa, Israel, February 24-27, 2014*, pgs. 329–334. ACM. doi: 10.1145/2557500.2557528. URL: https://doi.org/10.1145/2557500.2557528. (Cited on pg. 25)

Tula and Morimoto(2016a) Antonio Diaz Tula and Carlos H. Morimoto. Augkey: Increasing foveal throughput in eye typing with augmented keys. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pgs. 3533–3544, New York, NY, USA. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858517. URL: http://doi.acm.org/10.1145/2858036.2858517. (Cited on pg. 17)

Tula and Morimoto(2016b) Antonio Diaz Tula and Carlos Hitoshi Morimoto. Augkey: Increasing foveal throughput in eye typing with augmented keys. In Jofish Kaye, Allison Druin, Cliff Lampe, Dan Morris and Juan Pablo Hourcade, editors, *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016*, pgs. 3533–3544. ACM. doi: 10.1145/2858036.2858517. URL: https://doi.org/10.1145/2858036.2858517. (Cited on pg. 24)

Urbina and Huckauf(2010) Mario H. Urbina and Anke Huckauf. Alternatives to single character entry and dwell time selection on eye typing. In Carlos Hitoshi Morimoto, Howell O. Istance, Aulikki Hyrskykari and Qiang Ji, editors, *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA 2010, Austin, Texas, USA, March 22-24, 2010*, pgs. 315–322. ACM. doi: 10.1145/1743666.1743738. URL: https://doi.org/10.1145/1743666.1743738. (Cited on pgs. 81 and 85)

Velloso *et al.*(2017) Eduardo Velloso, Marcus Carter, Joshua Newn, Augusto Esteves, Christopher Clarke and Hans Gellersen. Motion correlation: Selecting objects by matching their movement. *ACM Trans. Comput. Hum. Interact.*, 24(3):22:1–22:35. doi: 10.1145/3064937. URL: https://doi.org/10.1145/3064937. (Cited on pgs. 19, 29, and 85)

Vertegaal(2008) Roel Vertegaal. A fitts law comparison of eye tracking and manual input in the selection of visual targets. In Vassilios Digalakis, Alexandros Potamianos, Matthew A. Turk, Roberto Pieraccini and Yuri Ivanov, editors, *Proceedings of the 10th International Conference on Multimodal Interfaces, ICMI 2008, Chania, Crete, Greece, October 20-22, 2008*, pgs. 241–248. ACM. doi: 10.1145/1452392.1452443. URL: https://doi.org/10.1145/1452392.1452443. (Cited on pg. 84)

Vidal *et al.*(2012) Mélodie Vidal, Andreas Bulling and Hans Gellersen. Detection of smooth pursuits using eye movement shape features. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pgs. 177–180, New York, NY, USA. ACM. ISBN 978-1-4503-1221-9. doi: 10.1145/2168556.2168586. URL: http://doi.acm.org/10.1145/2168556.2168586. (Cited on pg. 18)

Vidal *et al.*(2013) Mélodie Vidal, Andreas Bulling and Hans Gellersen. Pursuits: spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In Friedemann Mattern, Silvia Santini, John F. Canny, Marc Langheinrich and Jun Rekimoto, editors, *The 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '13, Zurich, Switzerland, September 8-12, 2013*, pgs. 439–448. ACM. doi: 10.1145/2493432.2493477. URL: https://doi.org/10.1145/2493432.2493477. (Cited on pgs. 24 and 25)

Vidal *et al.*(2014) Mélodie Vidal, David H. Nguyen and Kent Lyons. Looking at or through?: using eye tracking to infer attention location for wearable transparent displays. In Lucy E. Dunne, Tom Martin and Michael Beigl, editors, *ISWC'14, Proceedings of the 2014 ACM International Symposium on Wearable Computers, Seattle, WA, USA, September 13-17, 2014*, pgs. 87–90. ACM. doi: 10.1145/2634317.2634344. URL: https://doi.org/10.1145/2634317.2634344. (Cited on pgs. 28, 87, and 93)

Villanueva *et al.*(2006) Arantxa Villanueva, Rafael Cabeza and Sonia Porta. Eye tracking: Pupil orientation geometrical modeling. *Image Vis. Comput.*, 24(7):663–679. doi: 10.1016/j.imavis.2005.06.001. URL: https://doi.org/10.1016/j.imavis.2005.06.001. (Cited on pg. 20)

Vinnikov and Allison(2014) Margarita Vinnikov and Robert S. Allison. Gaze-contingent depth of field in realistic scenes: the user experience. In Pernilla Qvarfordt and Dan Witzner Hansen, editors, *Eye Tracking Research and Applications, ETRA '14, Safety Harbor, FL, USA, March 26-28, 2014*, pgs. 119–126. ACM. doi: 10.1145/2578153.2578170. URL: https://doi.org/10.1145/2578153.2578170. (Cited on pg. 27)

Wang and Hung(2019) Chun-Chia Wang and Jason C. Hung. Comparative analysis of advertising attention to facebook social network: Evidence from eye-movement data. *Comput. Hum. Behav.*, 100:192–208. doi: 10.1016/j.chb.2018.08.007. URL: https://doi.org/10.1016/j.chb.2018.08.007. (Cited on pg. 17)

Ward *et al.*(2020) Lindsey M Ward, Chrystal Gaertner, Lucrezia Olivier, Layla Ajrezo and Zoï Kapoula. Vergence and accommodation disorders in children with vertigo: A need for evidence-based diagnosis. *EClinicalMedicine*, 21:100323. (Cited on pg. 92)

Ware and Mikaelian(1987) Colin Ware and Harutune H Mikaelian. An evaluation of an eye tracker as a device for computer input2. In *ACM Sigchi Bulletin*, volume 17, pgs. 183–188. ACM. (Cited on pg. 23)

Weiser(1998) Mark Weiser. The invisible interface: Increasing the power of the environment through calm technology. In Norbert A. Streitz, Shin'ichi Konomi and Heinz Jürgen Burkhardt, editors, *Cooperative Buildings, Integrating Information, Organization, and Architecture, First International Workshop, CoBuild'98, Darmstadt, Germany, February 1998, Proceedings*, volume 1370 of *Lecture Notes in Computer Science*, pg. 1. Springer. doi: 10.1007/3-540-69706-3_1. URL: https://doi.org/10.1007/3-540-69706-3_1. (Cited on pg. 29)

Xiong *et al.*(2014) Xuehan Xiong, Qin Cai, Zicheng Liu and Zhengyou Zhang. Eye gaze tracking using an RGBD camera: a comparison with a RGB solution. In *The 2014 ACM Conference on Ubiquitous Computing, UbiComp '14 Adjunct, Seattle, WA, USA - September 13 - 17, 2014*, pgs. 1113–1121. doi: 10.1145/2638728.2641694. URL: http://doi.acm.org/10.1145/2638728.2641694. (Cited on pg. 22)

Yang *et al.*(2002) Qing Yang, Maria Pia Bucci and Zoï Kapoula. The latency of saccades, vergence, and combined eye movements in children and in adults. *Investigative Ophthalmology & Visual Science*, 43(9):2939–2949. (Cited on pg. 92)

Yoo and Chung(2005) Dong Hyun Yoo and Myung Jin Chung. A novel non-intrusive eye gaze estimation using cross-ratio under large head motion. *Comput. Vis. Image Underst.*, 98(1):25–51. doi: 10.1016/j.cviu.2004.07.011. URL: https://doi.org/10.1016/j.cviu.2004.07.011. (Cited on pg. 20)

Zemblys *et al.***(2018)** Raimondas Zemblys, Diederick C Niehorster, Oleg Komogortsev and Kenneth Holmqvist. Using machine learning to detect events in eye-tracking data. *Behavior research methods*, 50(1):160–181. (Cited on pg. 18)

Zhai *et al.***(1999)** Shumin Zhai, Carlos Morimoto and Steven Ihde. Manual and gaze input cascaded (MAGIC) pointing. In Marian G. Williams and Mark W. Altom, editors, *Proceeding of the CHI '99 Conference on Human Factors in Computing Systems: The CHI is the Limit, Pittsburgh, PA, USA, May 15-20, 1999*, pgs. 246–253. ACM. doi: 10.1145/302979.303053. URL: https://doi.org/10.1145/302979.303053. (Cited on pgs. 24 and 80)

Zhang *et al.***(2015)** Yanxia Zhang, Ming Ki Chong, Jörg Müller, Andreas Bulling and Hans Gellersen. Eye tracking for public displays in the wild. *Pers. Ubiquitous Comput.*, 19(5-6):967–981. doi: 10.1007/s00779-015-0866-8. URL: https://doi.org/10.1007/s00779-015-0866-8. (Cited on pg. 2)

Zhang and Zhou(2006) Yaoxue Zhang and Yue-Zhi Zhou. Transparent computing: A new paradigm for pervasive computing. In Jianhua Ma, Hai Jin, Laurence Tianruo Yang and Jeffrey J. P. Tsai, editors, *Ubiquitous Intelligence and Computing, Third International Conference, UIC 2006, Wuhan, China, September 3-6, 2006, Proceedings*, volume 4159 of *Lecture Notes in Computer Science*, pgs. 1–11. Springer. doi: 10.1007/11833529_1. URL: https://doi.org/10.1007/11833529_1. (Cited on pg. 72)

Zhou *et al.***(2008)** Feng Zhou, Henry Been-Lirn Duh and Mark Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In *7th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2008, Cambridge, UK, 15-18th September 2008*, pgs. 193–202. IEEE Computer Society. doi: 10.1109/ISMAR.2008.4637362. URL: https://doi.org/10.1109/ISMAR.2008.4637362. (Cited on pg. 25)

Zhu *et al.***(2006)** Zhiwei Zhu, Qiang Ji and Kristin P. Bennett. Nonlinear eye gaze mapping function estimation via support vector regression. In *18th International Conference on Pattern Recognition (ICPR 2006), 20-24 August 2006, Hong Kong, China*, pgs. 1132–1135. IEEE Computer Society. doi: 10.1109/ICPR.2006.864. URL: https://doi.org/10.1109/ICPR.2006.864. (Cited on pg. 20)