SAARLAND UNIVERSITY

FACULTY OF NATURAL SCIENCES AND TECHNOLOGY I

DEPARTMENT OF COMPUTER SCIENCE

Master's Thesis

COMPUTATIONAL MODELING AND PREDICTION OF GAZE ESTIMATION ERROR FOR HEAD-MOUNTED EYE TRACKERS

Submitted by

Michael Johannes Barz

on 13.04.2015

Supervisor:

Prof. Dr. Antonio Krüger

Advisors:

Florian Daiber (DFKI)
Dr. Andreas Bulling (MPI-INF)

Reviewers:

Prof. Dr. Antonio Krüger
Dr. Andreas Bulling (MPI-INF)

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

STATEMENTS

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

Einverständniserklärung

Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken,

(Datum/Date) (Unterschrift/Signature)

ABSTRACT

It is evident that there's a strong connection between the gaze of a human and his intentions. For instance you first look at an object before grabbing or manipulating it. That's why eye tracking is a very promising technology for human computer interaction (HCI). It provides insights into the user's mind, which can be used for interaction directly or to enhance it. Especially mobile eye tracking enables a flexible utilization of gaze as an input modality for HCI. Nevertheless eye tracking is still prone to errors constraining the development of gaze-based interfaces depending on accurate point of gaze (POG) estimates, e.g. interfaces where gaze replaces the mouse cursor. Current approaches either ignore this error or try to compensate it. The idea of this work is to head towards error-aware gaze-based interfaces considering the gaze estimation error as an inevitable part of itself and activating its full potential. What's missing is the intermediate part, predicting the gaze estimation error and enabling an adaptive behavior, such as magnifying objects in high error regions or moving small objects to low error regions. With this work I present a computational model capable of predicting the gaze estimation error for head-mounted eye trackers in real-time. On the way to get there I conducted two data recordings targeting at both, the gaze estimation and at the display detection by means of marker detection, which is essential for gaze-based interaction. The resulting data was used to train a support vector regression (SVR) model predicting the gaze estimation error with a root mean squared error of 1.01°.

1 Introduction

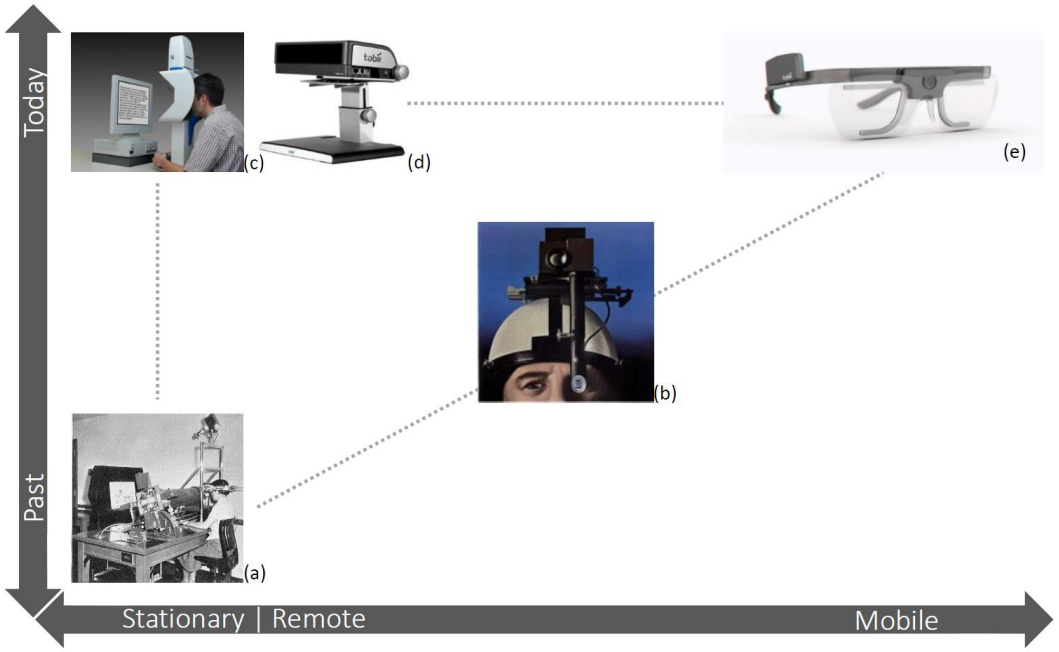

Figure 1: Evolution of eye tracking equipment until today and the scope of available products from stationary and remote systems to mobile ones. (a) 1935: corneal reflection film-based eye tracker [9]; (b) 1968: head-mounted eye tracker as cover image of the "Scientific American" [32]; (c) SMI iViewX Hi-Speed 1250 [26]; (d) Tobii X120 [33]; (e) Tobii Glasses 2 [33].

Starting more than one century ago eye tracking began its evolution as a large and invasive tool to investigate the human behavior based on their gaze. Gradually systems grew smaller, more flexible and in the end mobile by means of head-mounted devices (see Figure 1). Not least these advances are due to video-based techniques, which still play a major role today. Another interesting trend is the drift from expensive, proprietary and closed source solutions to affordable, extensible open source platforms [18].

In addition eye gaze is predestinated for intuitive interaction because our eyes naturally indicate what we are interested in and because they are readily available. The progress of eye tracking equipment and measurement techniques suggest gaze as a compelling modality for interaction with multiple ambient displays. Especially mobile eye tracking facilitates unobtrusive gaze-based interaction in everyday life settings, which is also referred to as pervasive eye tracking [6].

A key problem in context of mobile gaze-based interaction is that the gaze estimation error can vary considerably while the depth towards a fixation plane changes, e.g. towards interactive displays [10,20]. Beside the user's current position and orientation, deciding sources influencing the error are display and marker properties, eye tracker intrinsics as well as parameters of the calibration routine the eye tracker was initially calibrated with.

One solution to the problem can be found in gaze-contingent gaze interaction techniques (see section 1.1.3) that do not require accurate point of gaze (POG) estimates, such as eye gestures [7,34], relative eye movements [38,36] or the user's attention [35] (see section 2.2.4). These approaches are based on pattern matching or correlation, i.e. they are not affected by deviations of the estimated gaze point from the actual one. Selective interaction techniques that do require accurate POG estimates either assume that eye trackers deliver ideal gaze data or they try to deal with inaccuracies a priori, in real-time to improve user-experience or post-hoc (see section 2.2.3). Methods include extending of calibration methods [10,11], filtering of jitter caused by the tracking hardware or eye movements [29] or gaze-to-object mapping algorithms to snap gaze to interactive objects [30,31].

However they lack the possibility to embrace the inevitable gaze estimation error in interaction design, because they alleviate the symptoms only. The vision of this work is to proactively deal with inaccuracies by affording real-time error prediction and guidelines for gaze-based interaction. As a consequence this paves the way to adapt interfaces during runtime, e.g. by magnifying objects in high error regions or by moving them to low-error regions of the display. As foundation for this idea I present a computational model of gaze estimation error for monocular head-mounted eye trackers, general guidelines aiming at the context of gaze-based interaction and interaction prototypes leveraging the aforementioned gains.

In order to enable a better understanding my thesis continues with basic concepts of eye tracking and marker detection followed by related work. Subsequently I introduce the theoretical part of the presented error model in chapter 3 and information about the applied eye tracking hardware and software extensions in chapter 4. Chapter 5 provides detailed information on how I gathered data to actually generate the computational model, which is then evaluated in chapter 6. A discussion can be found in chapter 7 offering guidelines derived from my findings and presenting thoughts on possible applications by means of low-fidelity interaction prototypes in. Eventually I conclude my work in chapter 8 and I depict what present limitations are and how they can be solved by means of future work.

1.1 Eye Tracking

Exploring offers of leading eye tracking hardware manufacturers, e.g. Tobii Technology [33] and SensoMotoric Instruments (SMI) [26], it is obvious that there are mainly two groups of video-based eye tracking devices, namely stationary and mobile eye trackers (see Figure 1). Abstracting from that fact Hansen and Ji presented a general workflow for video-oculography (VOG) indicating two major fields of research: eye detection (see section 1.1.1) and gaze estimation (see section 1.1.2) [14]. They further stated that VOG devices utilizing active infrared illumination dominate the development. That is why I concentrate on this kind of devices in this work. Additionally I have chosen to address head-mounted eye trackers, because I believe that they are more flexible in context of human-computer interaction. Further I present fields of eye tracking applications in section 1.1.3.

1.1.1 Eye Detection

Eye detection is the first challenge when it comes to eye tracking and is essential for the second part of it, namely gaze estimation (see section 1.1.2). Commonly the target of eye detection is to identify the eye position, e.g. in terms of the pupil center. Hansen and Ji suggest a classification of related techniques into shape-based, appearance-based and hybrid methods [14] as illustrated in Figure 2.

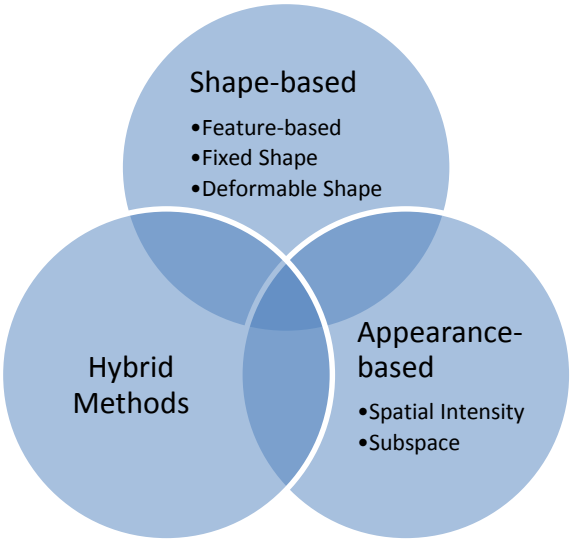

Figure 2: Classification of eye detection techniques according to [14].

Shape-based methods usually take into account image features like points or contours to fit a geometric eye model for each frame. A similarity measure decides, if the model could be applied successfully or not. The more complex models try to match fixed or deformable shape templates by minimizing an associated energy or error function. Others simply provide distinctive features such as the pupil contour and its center, also referred to as feature-based shape methods. One example is the widespread dark pupil detection driven by active infrared illumination. More holistic approaches are appearance-based methods

relying on image template matching. They either facilitate the spatial intensity information of an image or base upon properties of a certain subspace, e.g. its Fourier transform, to build an eye detection model. But a huge base of training data is required to account for possible variances in appearance. Combinations of two or more different techniques in one algorithm are denoted as hybrid-methods.

1.1.2 Gaze Estimation

Gaze estimation, i.e. determining gaze in terms of gaze direction or POG, is the second stage of eye tracking. Next to raw gaze, important classes of eye movements include saccades, fixations and smooth pursuit movements [14]. A fixation summarizes a couple of gaze samples close to each other in space and time. They are connected by fast, ballistic jumps, known as saccades. Smooth pursuits describe slower, but continuous movements of the eye, e.g. when following a moving object. In general the process of gaze estimation is based on correspondences between eye features as introduced in section 1.1.1 and the user's fixations. The most common approaches to model gaze are feature-based. They comprise feature-based shape methods for eye detection and either model-based or regression-based approaches to find a mapping. Finally gaze is predicted and reported as gaze direction in a respective scene coordinate system (3D vector) in case of model-based methods or as POG (2D point) in case of regression-based methods.

1.1.3 Applications of Eye Tracking

Today eye tracking maintains a wide variety of applications. Duchowski [12] already performed a study on that topic in 2002 and found two main categories, diagnostic and interactive systems. First, diagnostic systems were used to investigate the human behavior, e.g. during perception of arts and later to investigate how people look at advertisement or how they use an airplane cockpit. Due to advances in eye tracking technology and its strong link to the human behavior, gaze became an interesting modality for interaction. Interactive systems can further be split in two application subtypes, namely selective and gaze-contingent ones. While selective systems use the point of gaze for direct input, gaze-contingent systems take advantage of knowledge from the user's gaze. The model presented in this work is intended to improve on selective interactive systems. Examples for gaze-contingent interactive systems can be found in the related work section 2.2.4.

2 Related Work

My thesis is related to previous publications on mobile gaze-based interaction, especially approaches to find the point of regard relative to interactive areas of interest. Further I refer to studies dealing with the measurement and compensation – or avoidance – of gaze estimation error.

2.1 Mobile Gaze-based Interaction

Mobile eye-based input receives great attention in ongoing research. An essential part and key problem is the identification of interactive areas and the mapping of gaze points to these regions.

Yu and Eizenmann proposed the use of 2D-features to accomplish offline gaze to surface mapping [37]. First objects within the area of interest had to be labeled manually for one frame of the recording – the reference frame. With 2D-feature detection point correspondences and resulting homographies between each pair of frames were detected. This allowed them to transfer gaze data from all frames to the reference frame. Further the authors suggested to use at least four point correspondences radially and symmetrically arranged around the area of interest. For an ideal and a typical experimental setup they reached an accuracy such that 95% of the deviations have been lower than 0.32° and 0.9° respectively.

Bardins et al. propounded a setup embracing a binocular head-mounted eye tracker with infrared LEDs attached to it as well as a stereo camera statically aligned to one or more areas of interest in 2008 [3]. The fixed alignment and the LEDs enabled the stereo camera to estimate the tracker's pose in 3D scene coordinate space and allowed a calibration with respect to that space, i.e. the gaze was reported in terms of two 3D vectors. Eventually they received gaze points in real-time relative to one of the interaction regions with an average accuracy of 0.61°. One mentioned application was to augment an interface with information guiding the user, e.g. during visual search.

More recently Mardanbegi and Hansen presented an approach for mobile gaze-based interaction with multiple ambient displays [19]. Their algorithm detects display quadrilaterals in scene camera images by changes in luminance and identifies them by means of QR-codes. Gaze is then mapped to the screen with the aid of corresponding homographies. In an attendant qualitative evaluation of their prototype – facilitating the control of objects in a home environment – the authors found that the gaze estimation error is dependent on the position of the user, i.e. its distance and angle to the screen. In a similar setup Breuninger et al. [5] employed visual markers to detect one display with the objective of controlling household appliances such as TV sets or music players.

Kassner et al. presented Pupil – “an accessible, affordable, and extensible open source platform for mobile eye tracking and gaze-based interaction” [18]. They reported an accuracy of 0.6° and a precision of 0.08° under ideal conditions for their device. Even though Pupil supports marker-based detection of multiple surfaces and gaze-mapping by means of homographies in real-time, they did no evaluate that part nor its application for gaze-based interaction with ambient displays.

2.2 Gaze Estimation Error

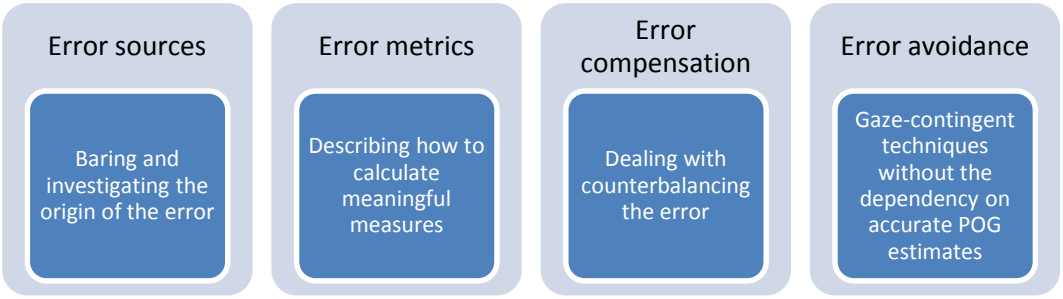

Figure 3: Topics which are related to the emergence and the treatment of the gaze estimation error.

There are mainly four topics regarding the gaze estimation error, namely error sources, error metrics, error compensation and error avoidance as shown in Figure 3. In the following sections I shortly describe each topic and present related publications that are concerned about it.

2.2.1 Error Sources

Understanding and dealing with an error premises knowledge about its sources. Although the importance of gaze estimation error has long been acknowledged, most works investigating its origin were published rather recently. Holmqvist et al. [15] suggested the following non-disjoint influencing categories, having eye tracking studies in mind: participants, operators, tasks, recording environment, geometry and eye tracker design. For gaze-based interaction all of them but operators are interesting, because interaction is not thought to be supervised. Below I delve into the remaining categories as shown in Figure 4 to motivate the scope of my work.

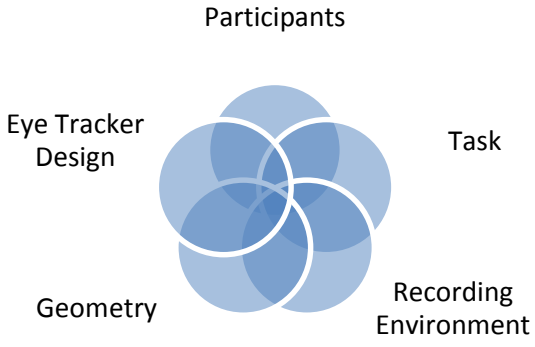

Figure 4: Error sources of VOG according to Holmqvist et al. [15] and adapted to the context of human-computer interaction.

2.2.1.1 Participants

The term participants describes the group of people using a device or its interface during a study instructed and monitored by an observer. Due to discrepancies in appearance, such as the eye physiology, or behavior of the single user the device and its algorithms might react in different ways. Nyström et al. conducted an eye tracking study with 149 participants, where one objective was to investigate the impact of these user-specific properties on the gaze estimation error [23]. They discovered several significant criteria, namely the presence of visual aids, the direction of eyelashes, the eye diameter and color, the wearing of mascara and if the measured eye was the dominant one. Beyond that Holmqvist et al. stated issues like differing ability and varying neurology and psychology [15]. For interaction scenarios this group might rather be called users with the difference that no observers are present.

2.2.1.2 Task

Regarding studies, a task describes a problem introduced by the observer, which has to be solved by all participants. Tasks are essential for the outcome and the expressiveness of a study and therefor have to be designed properly and non-biasing. In context of eye tracking Holmqvist et al. especially addressed tasks that involve moving around frequently as source for gaze estimation error [15]. Mardanbegi and Hansen came to an analog conclusion and referred to this error resulting from altering depths during gaze estimation as parallax error (see section 2.1) [19]. Subsequently they began to analyze the characteristics of the parallax error that they delineated by the epipolar geometry of monocular head-mounted eye trackers [20]. The authors reported that - for a constant calibration distance – the gaze estimation error increases with changing fixation distance. However the parallax error is not restricted to mobile settings, but also plays a role for remote eye trackers. Cerrolaza et al. [10] found a strong influence using a remote setting and introduced both, a new calibration method including depth as input and a device-specific mathematical approach for error compensation.

In an early work Hornof and Halverson observed the matter of increasing gaze estimation error against time of an eye tracking task. They proposed automatically triggered re-calibration to account for this issue [16]. Nyström et al. could confirm the coherence of time and error as a part of their study mentioned above [23]. Both used remote eye trackers for their investigations. John et al. put their focus on head-mounted devices and identified displacement, i.e. the device drifted from its original position, it had during calibration, and destroyed the underlying geometry (see section 2.2.1.4), as a major cause of the time dependent error [17]. By implication the presented partitioning is not entirely disjoint.

Since motion and time are actual factors of mobile gaze-based interaction, it is evident that tasks are a central source of errors to it.

2.2.1.3 Recording Environment

Another source of errors is the recording environment incorporating e.g. ambient infrared emitters or vibrations [15]. Most video-based eye trackers rely on active infrared illumination to cope with changing lightning conditions, i.e. the sun or other strong infrared emitters might emerge artifacts raising gaze estimation error. Moreover vibrations maintain the displacement of eye tracking devices and consequently affect the geometry (see section 2.2.1.4). Further Drewes et al. found changes in pupil size stemming from altering room brightness to be another related factor [11].

2.2.1.4 Geometry

Holmqvist et al. specified geometry as the alignment of the eye camera, the participant and the stimulus, i.e. the intended fixation target, to each other [15]. If a part of this geometry misbehaves according to the applied gaze estimation model an error is provoked. One example is the violation of geometry due to the displacement of a head-mounted eye tracker as stated by John et al. [17]. By this means the foundation of the calibration, which is meant to be static, is changed and a systematic error is induced.

2.2.1.5 Eye Tracker Design

The last category is due to the eye tracker embracing its hardware and software components as potential error sources [15]. Hardware includes factors like camera resolution, camera image quality, sampling rate and eye illumination are stated. Kassner et al. presented their open source eye tracking platform Pupil and suggested the detection routine, the gaze mapping model and the model calibration procedure as key points considering the software [18].

2.2.2 Error Metrics

The quality of gaze data obtained from a certain eye tracking system can be defined by different measures. The most prominent ones are spatial accuracy and spatial precision [18,15,23,16,17,4,2] as well as robustness [23,4,2] (see Figure 5). In general accuracy is a measure of central tendency whereas precision is a measure of statistical dispersion. Tracking robustness represents the ratio between the amount of valid samples and the total amount of processed camera frames. Altogether they enable the comparison of different eye tracking devices and studies.

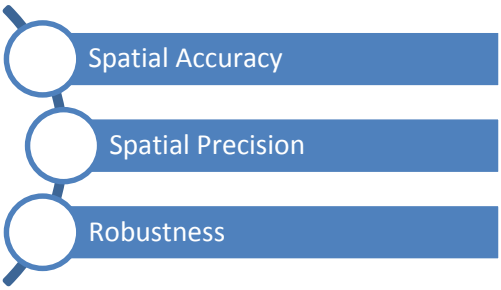

Figure 5: The most prominent metrics of the gaze estimation error for VOG: spatial accuracy, spatial precision and robustness.

Holmqvist et al. defined the terms accuracy and precision for spatial as well as for temporal deviation between the actual and the measured gaze [15]. The deviation for spatial and temporal aspects was measured in degrees of visual angle and time respectively. Invalid samples are mentioned, but not further regarded for evaluation. Additionally it was proposed to compute the values separately for horizontal and vertical dimensions. With TrackStick [4] and TraQuMe [2] there are two similar tools which determine gaze data quality for stationary eye trackers in the context of user studies. They aim to provide a consistent and comparable measuring approach across studies and – in case of TraQuMe – also across the eye tracking devices. Both works use the same definition of accuracy and precision as Holmqvist et al. [15]. In addition they consider tracking robustness as a third measure. You can find an overview with definitions taken from related work in Table 1.

Spatial Accuracy

$A = \frac{1}{n} \cdot \sum_{i=1}^{n} \theta_i$ *[15]*	Spatial accuracy A is calculated as the average angular offset θ_i between n measured fixations and the corresponding fixation targets.

Spatial Precision

$P = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} \theta_i^2}$ *[15]*	Spatial precision P is calculated as the Root Mean Square (RMS) of the angular distance θ_i between successive samples $(x_i, y_i) \rightarrow (x_{i+1}, y_{i+1})$.

Robustness

$R = \frac{n_{valid}}{n_{max}}$ *[23]*	Robustness is calculated as the ratio between valid samples n_{valid} and the total amount of captured samples n_{max}.

Table 1: Definitions of the three common measures applied to describe gaze estimation error: spatial accuracy, spatial precision and robustness.

2.2.3 Error Compensation

Methods to deal with gaze estimation error can be applied at different points in time, i.e. avoiding the error before it occurs (a priori), compensating the error during runtime (real-time) and balancing it after a recording was finished (post-hoc) as illustrated in Figure 6. For gaze-based interaction a priori and real-time methods are suitable, because they can actually impact the performance of an interactive system. Post-hoc methods take effect after the interaction already happened and thus are appropriate for studies only. However, the applied ideas might inform future a priori or real-time methods.

Figure 6: Compensation of the gaze estimation error can happen at several points in time: a priori, in real-time or post-hoc.

2.2.3.1 A Priori

Dealing with the gaze estimation error – a priori – means to compensate it beforehand by means of decisions as to the hardware design or the corresponding algorithms. At this point I clearly want to exclude techniques avoiding the error by not relying on accurate POG estimates, because they don't allow for sophisticated gaze-based interaction. I review them separately in section 2.2.4.

Regarding the hardware of video-based eye trackers, an improvement on the parallax issue can be reached, when the projection center of the scene camera is coincident with the eye ball center [20]. However, this requires a half mirror in front of the eye to not completely distract the human FOV.

On the part of the software Drewes et al. suggested to use two separate calibrations, one for a dark and one for a bright surrounding [11]. They aimed at balancing the error stemming from changing pupil size by dynamically weighting the estimates of the two resulting mapping functions. Similarly Cerrolaza et al. incorporated several depths for calibration to account for the parallax error [10]. Blignaut et al. investigated the impact of the chosen calibration method in general on the gaze estimation error and suggested to individually decide for one for each participant. Alongside they doubted that the additional effort was worthwhile.

2.2.3.2 Real-time

To cope with the gaze estimation error in real-time implies that some source already entailed noise, but the POG was not continued to use yet. Gaze-to-object mappings and eye movement filters are two common approaches to overcome the symptoms of noise. Recently Špakov provided a comparison of available methods in [29] and [30].

One approach for gaze-to-object mapping goes back to 2004, then Miniotas et al. suggested expanding targets in combination with their grab-and-hold algorithm to enhance pointing speed and selection accuracy [21]. The grab-and-hold algorithm was mainly a two stage dwell time approach to map gaze to an object. First, an item had to be

“grabbed” with a shortened dwell time of 200ms. Second, further gaze points were manipulated to “hold” the fixation on the object with a higher probability. One year later in 2005 Monden et al. introduced another gaze-to-object mapping, especially aiming at general WIMP user interfaces as known from Microsoft Windows and Mac OS [22]. The authors combined eye gaze with a mouse and associated the current POG with the nearest object when a click event was recognized. In 2008 Zhang et al. came up with three different algorithms concentrating gaze on an object, i.e. to prevent the dwell time from a reset [41]. The most recent work is by Špakov and Gizatdinova from 2014 [31]. They proposed a probabilistic mapping approach based on a growing set of gaze deviations extracted with so called required fixation locations as reference points. Originally required fixation locations were introduced by Hornof and Halverson [16]. A more unconcerned compensation can be achieved with eye movement filters, which are similar to usual noise filters known from signal processing. Špakov summarized and compared several filters reaching from the basic averaging of gaze samples to more sophisticated functions such as finite-impulse response filters [29].

2.2.3.3 Post-hoc

Post-hoc error compensation takes place after a recording has finished, e.g. within the scope of a study. In general these methods are similar to real-time compensation as they alleviate the symptoms of the gaze estimation error only. Though being executed post-hoc implies the availability of all data points compared to those of a small time window only.

Hornof and Halverson (2002) discovered that repeatedly measured individual deviations of the gaze point partially reveals a systematic error and can be used to compensate it [16]. They defined required fixation locations, i.e. on-screen locations which are certainly fixated at a particular point in time, for the measurement of gaze deviations. In 2011 Zhang and Hornof suggested to take the disparities between fixations and their nearest objects – that are more general required fixation locations – into account [40]. Their system moved all detected offset vectors to the origin of a 2D coordinate system and calculated the mode of these data points by means of density-based clustering. Subsequently the gaze points were corrected according to the resulting mode of disparities vector. John et al. improved on both works in 2012 by eliminating the need for required fixation locations at all, what makes their method applicable for all contents without prior knowledge about them [17]. The authors forged an algorithm, which is capable of finding a function for error compensation that minimizes an entropy-based error term. Recently, in 2014, Zhang and Hornof published the prosecution of their previous work, revising some major issues [39]. In the first instance they extended the required fixation locations by defining probable fixation locations to become more independent from the content. Nevertheless some knowledge about it still has to be available beforehand. Another restriction they claimed to be resolved is the static behavior of prior post-hoc methods, i.e. gaze has been corrected dependent on the user and on the device but not on the time and the spatial

position of the fixation target. Eventually the authors introduced a regression-based approach incorporating gaze deviations, time and gaze target positions to compensate the gaze estimation error.

2.2.4 Error Avoidance

Another solution to cope with the problem of gaze estimation errors can be found in gaze-contingent interaction techniques that do not require accurate POG values (see section 1.1.3). This implies that the error measures spatial accuracy and spatial precision as introduced above are not applicable, but because they don't play a role. Commonly these techniques are used for calibration-free and spontaneous interaction methods that do not afford high-fidelity input.

One example are Attentive User Interfaces (AUIs) which are sensitive to the user's attention and adapt their behavior upon that information. In 2003 Vertegaal suggested eye contact as determining factor for attention, i.e. the user's gaze reveals his intentions [35]. In that same year Shell et al. introduced EyePliances, small devices that are capable of sensing the user's visual attention on them and reacting to it [27]. Two years later in 2005 Smith et al. presented ViewPointer, a head-mounted successor, recognizing eye contact by means of infrared tags attached to ubiquitous interfaces and causing reflections on the eye ball [28].

The limitation of AUIs is that only two states can be differentiated per object, namely the user is looking at it or not. Eye gestures, which are based on relative eye movements, allow a more sophisticated distinction in this regard. With EyeMote Bulling et al. (2008) proposed a method to use electrooculography (EOG) signals for gesture detection in the context of gaming [7]. They were able to recognize a set of 8 distinct gestures with accuracies between 83% and 93%. Built upon the previously introduced EOG signal processing Bulling et al. presented an approach for eye-based activity recognition of 6 activity classes in 2009 [8]. The authors reported an average precision of 76.1% and a recall of 70.5%. Similar to AUIs this data can be incorporated as context information for user interaction and inform interface behavior.

Similar to eye gestures there are methods directly comprising relative eye movements for interaction. With SideWays Zhang et al. presented a solution for spontaneous gaze-based interaction with public displays in 2013 [38]. They detect both eye corners and the pupil center of each eye with an RGB camera. Depending on the distance of each eye center to the corresponding eye corners they determine if the user looks left, right or centered on the horizontal axis without prior calibration. Another work in 2013 is Pursuits of Vidal et al. [36]. They proposed a system correlating the eye movements gathered by a remote eye tracker with objects dynamically moving on an interface. Their approach relies on Pearson's product moment correlation coefficient to calculate the similarity between these movements.

3 Modeling of Gaze Estimation Error

Looking towards the idea of error-aware gaze-based interfaces the first step is to find a proper model to predict that error. This chapter starts with conveying a solid background about applied error metrics for monocular head-mounted eye trackers in section 3.1 and concerned error sources in section 3.2. Second, based on these considerations, I propose an error model covering the full processing pipeline for mobile gaze estimation in context of gaze-based interaction, namely mapping of pupil positions to scene camera coordinates, detection of ambient displays and gaze mapping to these displays (see Figure 7).

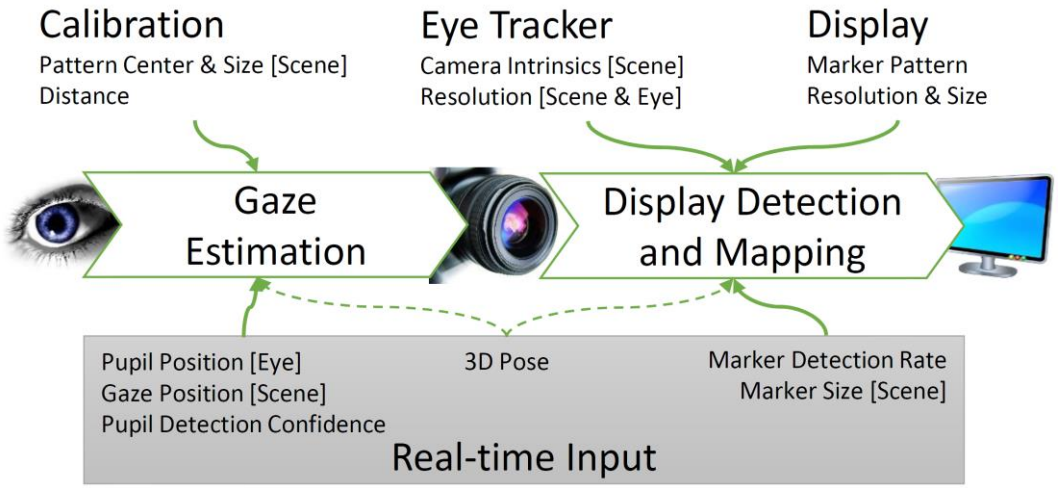

Figure 7: Gaze estimation error model for head-mounted eye trackers comprises two components – Gaze Estimation and Display Detection & Mapping. Model inputs include parameters for Calibration, Eye Tracker and Display as well as certain real-time information.

3.1 Applied Error Metrics

Monocular head-mounted eye trackers are typically equipped with two cameras: a scene camera that captures part of the user's field of view (FOV) and an eye camera that records a close-up video of the user's pupil position and eye movements [18]. For gaze estimation it is an essential task to find a function mapping 2D pupil position in the eye camera coordinate system to 2D gaze positions in the scene camera coordinate system. For that purpose the user is asked to look at predefined gaze targets of which the scene camera coordinate is known or detected, e.g. with the aid of a fiducial marker. Alongside the user's pupil positions are being tracked resulting in pairs of 2D coordinates. Finally they are associated with each other using a first or second order polynomial – a process known as calibration. If gaze estimates are to be used for interacting with one or more ambient displays the mapping has to be extended to the corresponding display coordinate system, e.g. by using visual markers attached to the display [37] or by detecting the display itself [19]. As stated in section 2.2.2 there are three common measures to describe the estimation error originating from that process: spatial accuracy, spatial precision and robustness. Both spatial aspects are defined in terms of visual degrees. Since this work deals with interface design as a central part, the definitions are rephrased using pixel units. I decided

to report performance in pixels instead of degrees, because this measure is directly linked to the distance to the display and its resolution. Both factors vary considerably for mobile gaze interaction settings using head-mounted eye trackers and are therefore important parameters need to be included as part of the error model. Later on this enables a more intuitive understanding of the error, e.g. when predicting it in real-time or when using the output for interaction design. Section 3.1.4 illustrates how these technical terms can be interpreted in practice.

3.1.1 Spatial Accuracy

Spatial accuracy A is defined as the average Euclidean distance (unit: pixel) between n fixations φ and the corresponding fixation target χ.

$$A = \frac{1}{n} \cdot \sum_{I=1}^{n} \|\varphi_i - \chi_i\|$$

3.1.2 Spatial Precision

Spatial Precision P for one fixation is defined as the Root Mean Square (RMS) of the Euclidean distances (unit: pixel) between subsequent samples v_i and v_{i+1} with n samples available.

$$P = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^{n-1} \|v_i - v_{i+1}\|^2}$$

3.1.3 Robustness

Robustness R is defined as the ratio between the amount of valid samples n_{valid} and the number of theoretically reachable valid samples n_{max}. Samples are counted as invalid if the eye detection does not find eye features, which means that no gaze estimation is possible. Additionally samples are invalid if the visual angle between the estimated gaze point and the corresponding fixation target is greater than 5 degrees in order to account for blinks or similar.

$$\mathrm{R} = \frac{\mathrm{n_{valid}}}{\mathrm{n_{max}}}$$

3.1.4 Interpretation

Figure 8 shows four exemplary measurements in order to clarify what good and bad spatial accuracy and precision actually denotes. Hereby the orange squares represent samples of one fixation, the gray triangle indicates the centroid of that fixation and the blue diamond shows the fixation target point. Starting at the lower left, both accuracy and precision are low. As you can see the fixation centroid has a large offset against the fixation target (low accuracy) and there is a high degree of scattering of the individual gaze samples (low precision). An improvement in accuracy is reached on the upper left, where the fixation centroid and the target are nearby each other. Nonetheless the samples are

still spread. Examining the right half of the figure, it is obvious that the degree of dispersion is lower and thus the precision higher. All in all the best condition can be found on the upper right, where accuracy and precision are high.

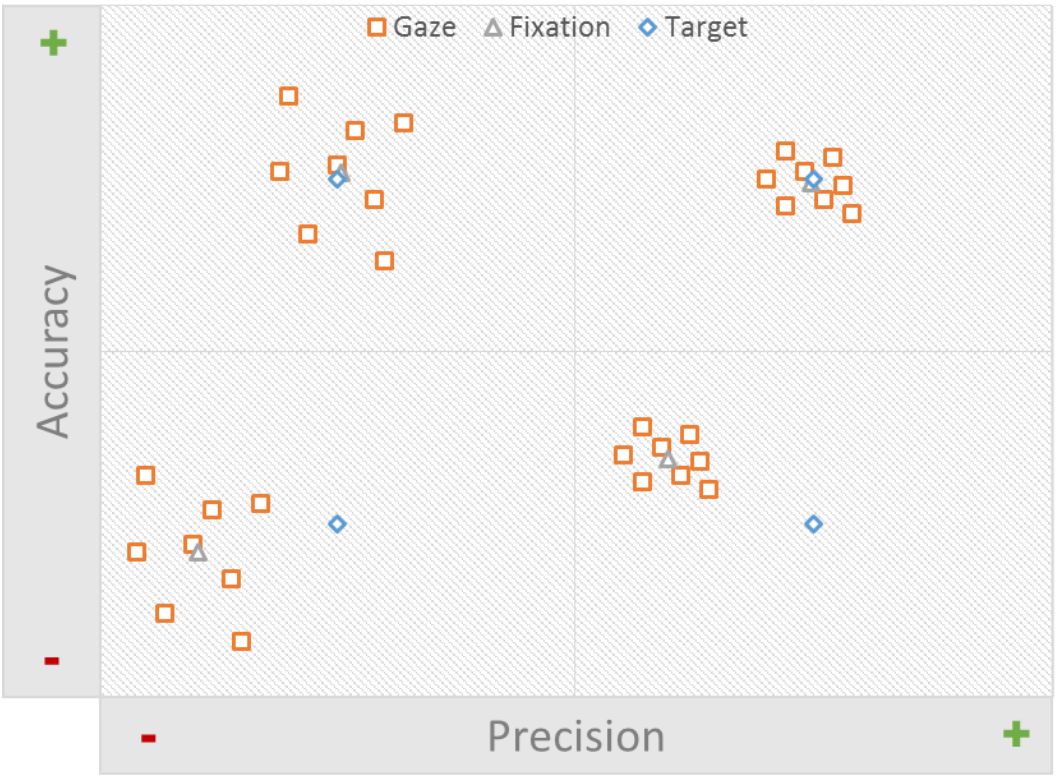

Figure 8: Illustration of high and low degree of spatial accuracy and spatial precision.

3.2 Concerned Error Sources

In this work I concentrate on the gaze estimation error stemming from interaction with mobile eye trackers. Error sources of study tasks as mentioned in section 2.2.1 are a good point to start at as they coincide with the major elements of interaction. Related sources are the parallax error and eye tracker displacement. Even if I see displacement as a serious error source, I exclude it from my model for two reasons. Firstly, there's recent work on automatic and real-time compensation by Špakov and Gizatdinova from 2014 [31] and secondly an inclusion would go beyond the scope of my work. Another error relevant for mobile gaze-based interaction arises, because the interaction distance and thus the scene camera's FOV coverage of the calibration pattern continuously varies. This implies that the calibration won't account for the whole FOV of the scene camera all the time. In this case extrapolation has to be applied for gaze estimation in outer areas, causing the extrapolation error [25]. Another important error source is the display detection and mapping itself. In the following I explain the parallax error, the extrapolation error and the error stemming from the display detection in detail and how the error model accounts for them.

3.2.1 Parallax Error

Calibration is typically performed for a fixed distance between the user and the fixation plane used for calibration (calibration plane), such as a display. At this distance the eyes are aligned to each other and to the eye camera in a specific way. Varying the depth of the fixation plane afterwards results in a different alignment and consequently an error, the so-called parallax error.

Mardanbegi et al. investigated the parallax error for monocular head-mounted eye trackers [20]. To convey a better understanding they simplified the setting by regarding two dimensions only and illustrated the cause similar to Figure 9. The scene camera is represented as a pinhole camera. Assume that the shown system is calibrated at distance d_c with respect to the calibration plane, i.e. when looking at point X_1 (the visual axis intersects X_1) the gaze estimate should be X_1' in scene camera coordinates. However when focusing X_2 on the further fixation plane the visual axis intersects both points X_1 and X_2 and the gaze estimate will again be X_1' instead of X_2', because the system was calibrated at distance d_c. The distance e between the actual and the aspired gaze estimate $(X_1' - X_2')$ represents the parallax error. For a non-simplified system the parallax error would be a 2D or 3D vector, depending on the gaze model (see section 1.1.2).

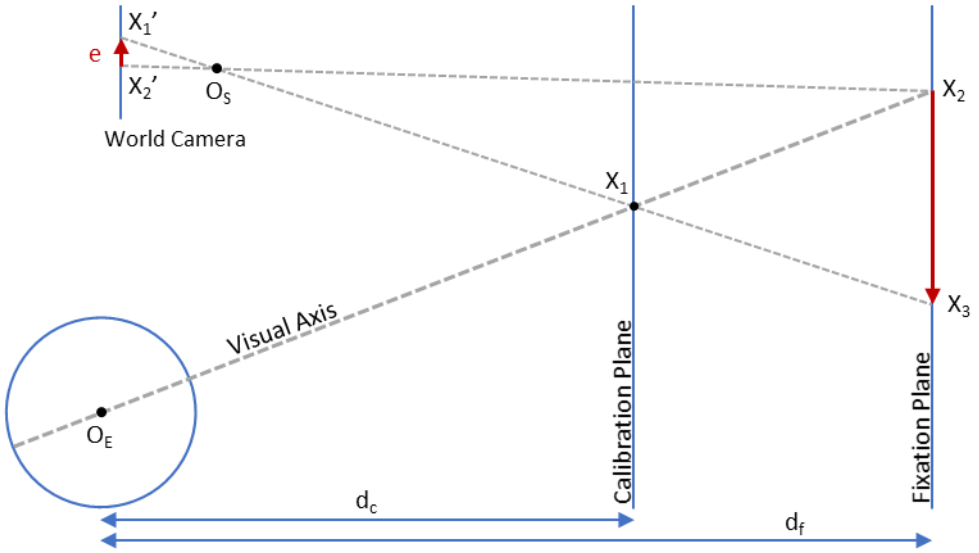

Figure 9: Illustration of parallax error and its cause in 2D [20]

3.2.2 Extrapolation Error

The calibration routine of an eye tracker often relies on regression-based techniques as introduced in section 1.1.2, i.e. calibrating a system means to collect tuples of 2D eye features and corresponding positions in scene camera coordinate space. Subsequently a function according to these samples is generated and used to estimate gaze. With a high probability there are no value tuples for all regions of the scene camera's FOV as pointed out in Figure 10. Eventually the algorithm can interpolate between those points, but has to extrapolate for the non-calibrated outer area, which causes the extrapolation error.

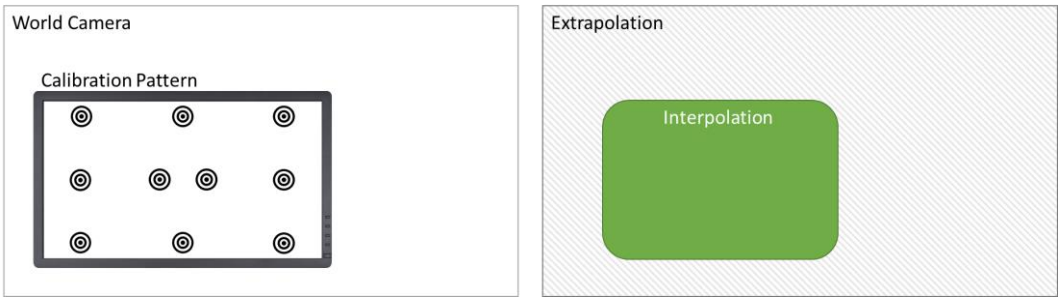

Figure 10: Calibration patterns used with head-mounted eye trackers do not cover the full FOV of the scene camera or of the user wearing it (left). This results in two logical regions for gaze estimation, an interpolation and an extrapolation region. While the use of interpolation is intended by the gaze estimation algorithm, extrapolation rather is an artifact to account for gaze in non-calibrated areas.

3.2.3 Display Detection and Mapping

For this work I applied a marker-based display detection and 2D homographies for mapping gaze similar to [19] and [5]. The model proposed in this work shall incorporate the error stemming from that component, because it is essential for gaze-based interaction with displays. Marker detection is a widespread technology in the field of augmented reality, where accuracy plays a great role as well as for eye tracking. Thus there already have been attempts to investigate the error of marker detection and tracking. Abawi et al. [1] investigated the accuracy of marker detection of the ARToolKit for certain distances and rotations around the y-axis resulting in an error function (see Figure 11).

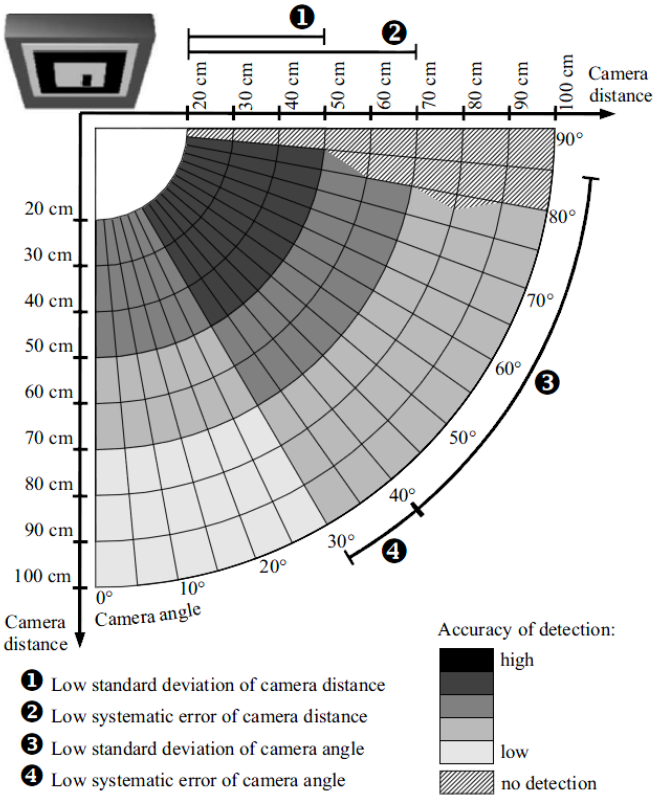

Figure 11: Accuracy the detection and tracking of visual markers as a function of camera distance and camera angle [1].

3.3 Error Model

Parameter	*Description*
P_x	Normalized pupil position [%]
P_y	Normalized pupil position [%]
T_x	Normalized scene target position [%]
T_y	Normalized scene target position [%]
S_p	Relative calibration pattern size [%]
d_t^x	Scene target to calibration pattern relation [%]
d_t^y	Scene target to calibration pattern relation [%]
d_p^{rel}	Relative difference to calibration distance [%]
C	Pupil detection confidence [%]
d	Distance between user and display [cm]
α	Rotation around x-axis (pitch) [°]
β	Rotation around y-axis (yaw) [°]
M	Marker detection rate [%]
S_{marker}	Marker size [px]

Table 2: Input parameters of the gaze estimation error model consisting of parameters covering scene camera mapping error (top) and the display mapping error (bottom).

The error model I propose in this work consists of two components: one component for mapping 2D pupil positions in eye camera coordinates to 2D scene camera coordinates (Gaze Estimation Error) and a second error component for detecting interactive displays in the environment and for mapping gaze from scene camera coordinates to display co-ordinates (Display Detection and Mapping Error). Together they cover the full processing pipeline for mobile gaze-based interaction as illustrated in Figure 7 with a special regard for the error sources mentioned earlier, the parallax error, the extrapolation error and erroneous conditions for marker detection (see section 3.2). The model is intended to predict the gaze estimation error in terms of spatial accuracy and for that purpose takes a number of input parameters that are summarized in Table 2. The upper half of the table lists parameters for the first error component and the lower half for the second one. As proposed by Holmqvist et al. I use separate models for the error in x and y direction [15].

3.3.1 Input Parameters

The first four parameters are the pupil position in eye camera coordinates (P_x and P_y) as well as target positions in scene camera coordinates (T_x and T_y), respectively normalized with the camera and display resolution. As targets one can either use the current gaze location as an approximation – for real-time estimation of the error around the point of gaze – or pre-defined targets, e.g. for simulating the to-be-expected gaze error for a given display setting.

Although the calibration pattern size is fixed regarding display coordinates, head movements can influence the center position and the relative size of the pattern with regard to the scene camera. I propose two novel measures that are robust to this variability and calculated from the calibration data. I compute S_p as ratio between the calibrated area in scene camera space and the total scene camera area. Normalized by S_p I define d_t^x (and d_t^y accordingly) as difference between scene target T_x and calibration center C_x in scene camera coordinate space.

$$S_p = \frac{calibratedArea}{res_{scene}^x \cdot res_{scene}^y}$$

$$d_t^x = \frac{|C_x - T_x|}{\sqrt{S_p} \cdot \frac{1}{2} \cdot res_{scene}^x}$$

For the human eye a 50cm movement perpendicular to the display is not the same starting either at 100cm or at 1000cm. This is caused by the alignment of the eyes to each other and to the eye camera (see the discussion of the parallax error above, section 3.2.1). I therefore introduce another measure d_p^{rel} describing the difference between recording and calibration distance, normalized by the squared calibration distance.

$$d_p^{rel} = \sqrt{\frac{|d_{rec} - d_{cal}|}{d_{cal}^2}}$$

The pupil detection confidence C is a custom measure of the Pupil eye tracker by Kassner et al. [18] indicating the quality of the detected eye features, here the pupil ellipse. Most eye trackers of other brands report a similar value, which can replace Pupil's confidence.

The model has additional input parameters for the display detection and gaze mapping. These describe the 3D pose of the eye tracker relative to the display -- comprising the distance between user and display d, the pitch and yaw rotation of the eye tracker α and β, as well as the marker detection rate M and size S_{marker}.

The eye tracker and display parameters only have to be measured once or are even provided by the device manufacturer, such as camera intrinsics and resolution as well as display resolution and size. For real-time estimation of the error, other parameters have to be measured on the fly. These include the pupil position in eye camera and the corresponding gaze position in scene camera coordinates, the 3D head pose, as well as marker detection parameters (marked in grey in Figure 7).

3.3.2 Hypotheses

Following I pose research questions and according hypotheses regarding the error behavior, which are relevant for the components of the suggested error model and its functionality. I conducted a series of measurements to check if these assumptions are valid and to gather training data to finally build a working model (see chapter 5).

For the first error component – gaze estimation – the question is, how the parallax error and the extrapolation error do contribute to the overall gaze estimation error? Even though the error propagates with display detection and mapping, gaze estimation does not cause a supplementary error for the second component. This allows for separate recordings later on, one for each component. My hypothesis concerning the parallax error is that

H1: *The larger the offset between calibration and fixation plane,*
the higher is the impact of the parallax issue and
the higher is the gaze estimation error.

For the extrapolation error my hypothesis is that

H2: *The smaller the calibration region,*
the lower is the overall spatial accuracy &
the farther a fixation target is from the calibrated area,
the higher is the gaze estimation error.

For the second error component – display detection and mapping – the question is: which impact has the distance and angle of the scene camera towards the target display on the overall gaze estimation error and what is the role of marker properties and conditions? Based on the findings of Abawi et al. [1], i.e. the distance and angle of the scene camera to the target display are vital factors to marker detection I state the following hypothesis:

H3: *The farther the camera is away from the display and*
the higher the angle is relative to the perpendicular axis,
the higher will be the error for display detection
and thus for the final stage of gaze estimation.

4 Eye Tracking Hardware and Software

Figure 12: Pupil Pro head-mounted monocular eye tracker [18].

Building a gaze estimation error model requires to investigate the behavior of an eye tracker and of the corresponding software. As an example I use the Pupil Pro eye tracker, which is a head-mounted, monocular VOG system. The device itself is not ready to use for gaze-based interaction, but is accompanied by an extensible open source software platform called Pupil Capture. Within the following sub sections I introduce the Pupil Pro tracker, the Pupil Capture software and my observations and extensions on them.

4.1 Pupil Pro Eye Tracker

The Pupil Pro eye tracker [18] features a scene camera with a resolution of 1280x720 pixels and an eye camera with a resolution of 640x480 pixels, both capturing videos at 30 fps (see Figure 12). The eye camera uses an IR filter and active IR illumination for dark pupil detection. The manufacturer evaluated the accuracy of their device under ideal conditions with 0.6° in spatial accuracy and 0.08° in spatial precision as result. The reported FOV of the scene camera is 90°, but for further investigations it is of great importance to know the accurate horizontal and vertical FOV of the camera and to gather its intrinsic parameters.

4.1.1 Determining Parameters for Scene Camera

The availability of accurate FOV values for X- and Y-direction is important for the conversion of the gaze estimation error between pixel and degrees of visual angle as unit. Therefore I positioned a drafting board orthogonally to a desk and put the scene camera 25cm away in line to the origin of that board (see Figure 13 left). Subsequently I was able to determine the vertical and horizontal range covered by the camera on the reference display as can be seen in Figure 13 (right). The horizontal range covers two times 20.3cm and

the vertical range covers two times 11.7cm. For the given distance of 25cm this results in a horizontal FOV of 78.16° and a vertical FOV of 50.16°. Thus the diagonal FOV is 92.87°, i.e. the result is nearly equal to the FOV of 90° reported by the manufacturer.

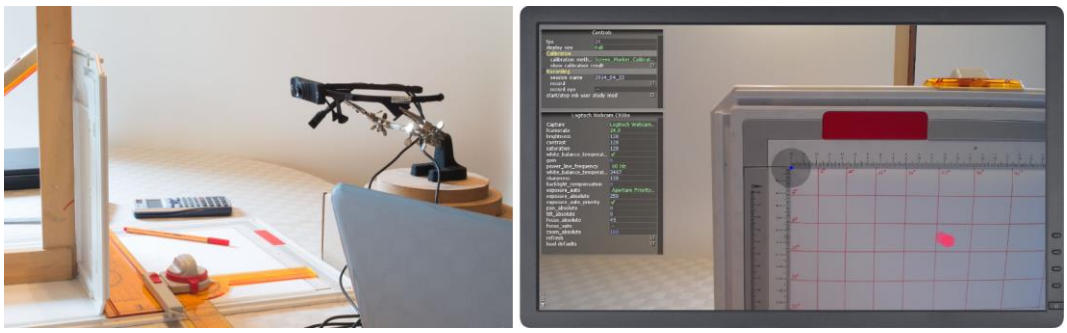

Figure 13: Apparatus to investigate FOV of the Pupil tracker's scene camera (left) and the corresponding camera still frame (right).

The intrinsic parameters of the scene camera are essential for accurate marker tracking and 3D pose estimation. To determine these parameters I followed the instructions of the OpenCV documentation [24] and recorded 40 images containing the checkerboard pattern from different perspectives. The results, a transformation matrix and distortion coefficients, facilitate undistorted camera images. Distortions can occur in the form of radial distortion, also known as "barrel" or "fish-eye" effect, or as tangential distortion, caused by lenses, which are not perfectly parallel to the camera sensor.

4.2 Pupil Capture and Extensions

Pupil devices come with an extensible open source software incorporating eye tracker calibration, gaze mapping and recording of gaze data and video streams [18]. Leveraging their plugin system I extended Pupil Capture according to the needs of my proposed error prediction model (see chapter 3) and of the planned data recordings (see chapter 5). To account for all conditions of the user study I adapted existing plugins, namely the built-in calibration and record functionality. With these changes I was able to choose different sizes for the on-screen calibration pattern during a study session (see Figure 14) and additional data could be logged.

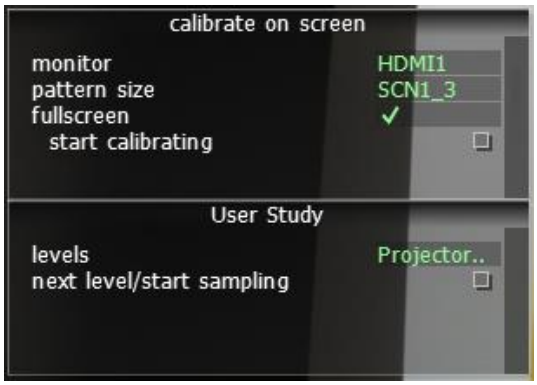

Figure 14: Extended control panel of standard calibration (top); Newly added control panel for level management during user study (bottom).

4.2.1 User Study Module

For the user study I added a plugin for selecting a set of fixation targets (levels) dependent on the condition (see section 5.1). Figure 14 (bottom) shows the corresponding plugin menu, further providing the functionality to trigger the data recording and to switch to the next level by a button or a shortcut. The actual fixation target was shown as a semi-transparent overlay on the scene camera view as a reference to adjust a cross-hair for the participant. The cross-hair was presented on a second display (see section 5.1.6).

4.2.2 Display Detection and Gaze Mapping

The Pupil device in combination with Pupil Capture facilitates gaze estimation, i.e. it covers the first part of the full processing pipeline for gaze-based interaction as introduced in chapter 3. To account for the second part, that is the display detection and the mapping of gaze to that display, I had to implement a further plugin. In a first step the plugin shows a full screen marker pattern on a display of choice. The marker pattern can be chosen according to the conditions of the data recording (see section 5.2.1). Subsequently the algorithm detects the display in the scene camera image. By means of the point correspondences a homography $H_{display}^{scene}$ is computed, for mapping points from scene camera space to display space. Applying the homography on the current gaze estimate – gathered by Pupil Capture – resulted in a gaze estimate relative to the display coordinate space (see Figure 15). In general this approach is similar to [19] as introduced in section 2.1, but I use a marker-based display detection similar to [5], in order to be more robust against changing lightning conditions. For marker detection I used the open source library ArUco [13].

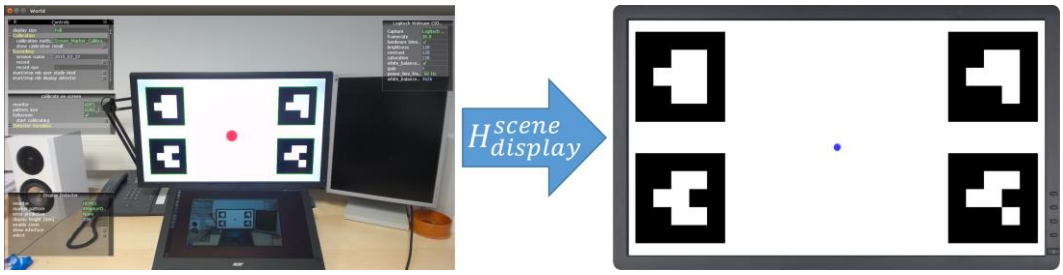

Figure 15: The gaze estimate (red dot on the left) is mapped by a homography from scene camera space to display space (blue dot on the right). The homography is computed with the aid of a marker-based display detection.

5 Data Recordings and Results

To get a data corpus for investigating the gaze estimation error and finally to build a model according to chapter 3, I conducted two separate recordings. One recording mainly aimed at the extrapolation and parallax error, i.e. at the gaze estimation component of the error. For this purpose I recruited 15 participants for a user study. Another recording aimed at error-prone conditions for display detection and mapping. In this case I conducted a data recording without participants, since no gaze data was required.

5.1 Extrapolation and Parallax Error

Within the scope of the user study I carried out two measurements to investigate the first error component when it comes to gaze-based interaction, the gaze estimation. The measurements concentrated on the parallax error and on the extrapolation error respectively, each executed in a controlled setting and to quantify their contribution to the overall gaze estimation error. Key parameters that I varied in both measurements were the distance between user and display during calibration d_{cal} and recording d_{rec} as well as the absolute size of the calibration pattern. I performed two further measurements to first, get an idea about gaze estimation in switching contexts, i.e. when exchanging a wall-mounted display with a tabletop screen, and second to figure out the participants' comfort-FOV, i.e. the region of the human FOV where focusing is still convenient. In total 15 volunteers were recruited for the study (eight female), aged between 19 and 50 years ($M = 24.067, SD = 7.459$). Each participant received 15€ as compensation.

5.1.1 Measurement 1: Extrapolation Error

To determine the extrapolation error, d_{cal} and d_{rec} were fixed to 250cm while the absolute edge length of the calibration pattern shown on the display (projector canvas) was varied between 100%, 75% and 50%. I asked the users to calibrate the eye tracker three times, each followed by one recording in which they looked at 13 target locations equally distributed across the FOV of the scene camera (see Figure 16).

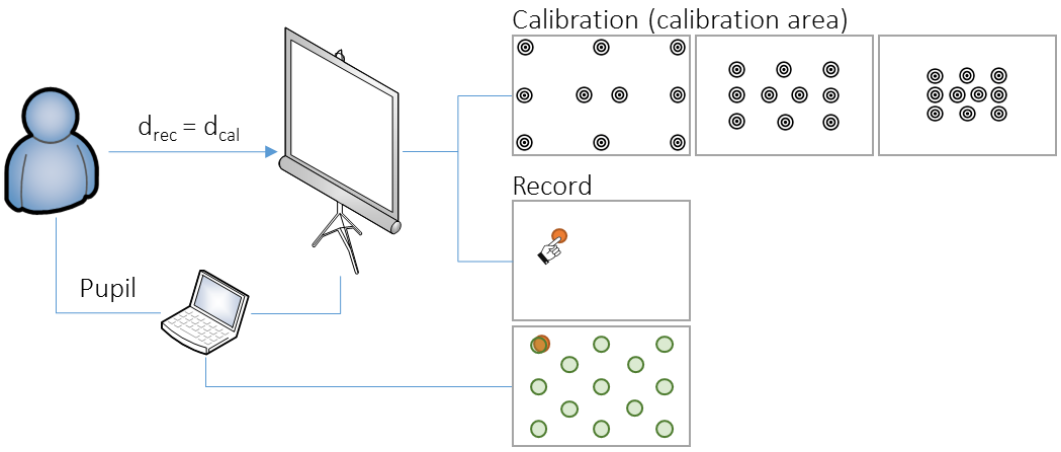

Figure 16: Experimental setup to measure extrapolation error. Participants looked at 13 locations defined with respect to the scene camera's FOV (shown in green) after calibrating the eye tracker on calibration patterns with three different sizes.

5.1.2 Measurement 2: Parallax Error

To determine the parallax error, the difference between d_{cal} and d_{rec} was varied starting from -150cm to +150cm. The absolute size of the calibration pattern on the display (projector canvas) was changed accordingly, i.e. such that its relative size with respect to the scene camera's FOV remained constant. Similar to the first measurement users were asked to calibrate the system three times, but from three different positions $d_{cal} \in \{100\ cm, 200\ cm, 50\ cm\}$. After each calibration users then performed three recordings, one at the current calibration distance and two at the other distances. The target locations were the same as for the first measurement.

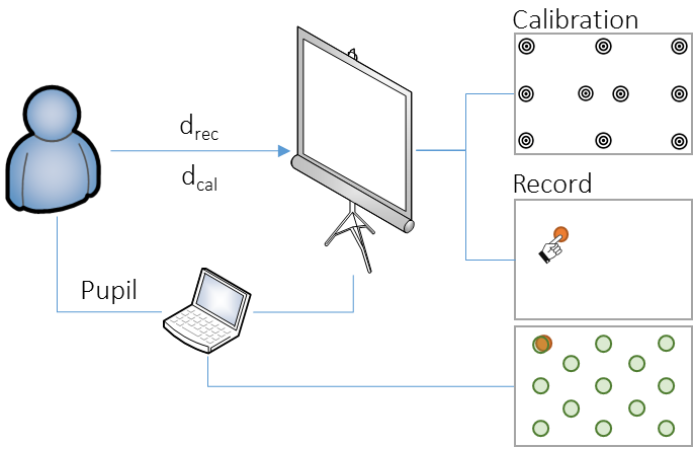

Figure 17: Experimental setup to measure parallax error. Participants looked at 13 locations defined with respect to the scene camera's FOV (shown in green) while their distance to the display was varied between -150 cm and +150 cm.

5.1.3 Measurement 3: Switching Context

To determine the error stemming from a context switch, d_{cal} and d_{rec} were fixed to 100cm while the context was varied, i.e. calibration and recording were performed across different devices. In total I asked the user to calibrate the eye tracker two times, once for a wall-mounted display and once for a tabletop device. Each calibration was followed by a record sequence on each of both devices. In contrast to previous recordings only seven scene levels were shown and arranged trapezoidal to account for the horizontally aligned tabletop device (see Figure 18).

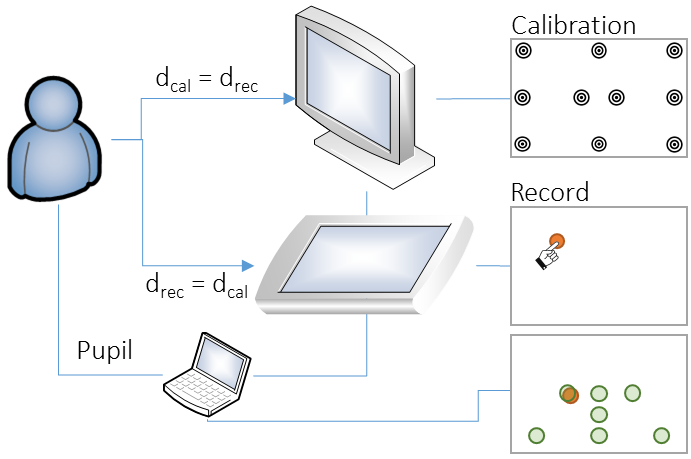

Figure 18: Experimental setup to measure error for switching contexts. Participants looked at 7 locations defined with respect to the scene camera's FOV (shown in green) after calibrating the eye tracker either on a wall-mounted display or on a tabletop screen.

5.1.4 Measurement 4: Comfort-FOV

To determine the comfort-FOV the user was seated 100cm in front of the display (projector canvas) without an eye tracker. The display showed a dynamic view containing a circle – bound to the mouse position – and a stroke indicating one of 16 circularly arranged orientations (see Figure 19). I instructed the user to give notice as soon as focusing the intersection point was no more convenient and started to slowly increase the radius of the circle. On signal by the user I stored the radius for the prevailing orientation and rotated the stroke to the next position.

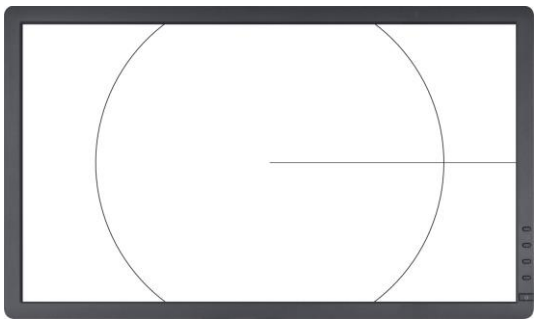

Figure 19: Dynamic HTML-based view showing a circle, which is bound to the mouse cursor, and a stroke, which can be rotated 16 times around the circle center point. The rotation as well as storing the radius are triggered by a mouse click.

5.1.5 Apparatus

To record gaze I used a PUPIL head-mounted eye tracker in combination with the provided open source software, which I extended according to the procedure of the study (see section 4). Stimuli for measurement 1, 2 and 4 were shown using a projector mounted at the ceiling with a resolution of 1400 x 1050 pixels and a corresponding display size on the canvas of 267 x 200 cm (5.25 pixel/cm ~= 13.34 dpi). For measurement 3 I applied two displays, a 50" wall mounted screen and a 40" tabletop device configured as display, both with a resolution of 1920 x 1080 pixels (17.42 pixel/cm ≈ 44.25dpi and 21.68 pixel/cm ~= 55.07 dpi respectively).

5.1.6 Procedure

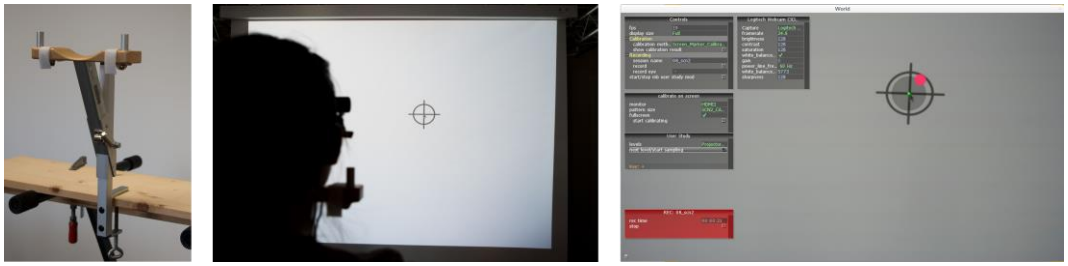

Figure 20: During calibration and recording the participant's head was fixed with a chin rest (left). It was positioned perpendicular to the vertical center axis of the screen and below the center point, i.e. the eyes of the participant were perpendicular to the center (middle). When recording the scene view on the laptop showed a semi-transparent overlay for each fixation target (grey circle with green dot) and the current gaze estimate (red circle). Thus the conductor could manually position the cross-hair on the projector canvas.

The study comprised two appointments per participant, one for measurement 1 & 2 and another for measurement 3 & 4. At the beginning of the first cycle participants were introduced to the experiment and asked to complete a preliminary questionnaire on demographics and prior eye tracking experience. For each measurement the eye tracker was calibrated several times using the standard 10-point calibration followed by one or more record sequences as defined above (see sections 5.1.1 & 5.1.2). Depending on the consecutive number of the participant the measurement order was altered to cope with potential learning effects. The stimuli during a record sequence were visualized on-screen as cross-hair and positioned manually by the conductor with the help of an overlay in the scene view of the laptop. At any time, i.e. when calibrating and recording, the participant's head was fixed by means of a chin rest to avoid unnecessary noise (see Figure 20). The first cycle took on average 50 minutes per participant. The second cycle embraced measurement 3, which was performed first, and measurement 4 (see sections 5.1.3 & 5.1.4). On average the second appointment required 20 minutes and in the end each participant received the proposed incentive. Table 3 summarizes the independent and dependent variables for measurement 1 to 3. The independent variable for measurement 4 was the direction for eye ball rotation, the dependent variable was the eye ball rotation in degrees of visual angle.

Independent Variables

Calibration Pattern Size $S_p^{display}$	$edge\ length \in \{100\%, 75\%, 50\%\}$
Distance d_{rec}	$d_{rec} \in \{100cm, 200cm, 250cm\}$
Distance d_{cal}	$d_{cal} \in \{100cm, 200cm, 250cm\}$
Fixation Target χ	As illustrated in Figure 16 & Figure 18, $\chi \in \mathbb{R}^2$
Display Size & Orientation	$context \in \{(wall, 50"), (\text{tabletop}, 40")\}$

Dependent Variables

Spatial Accuracy A	As defined in section 3.1.1
Spatial Precision P	As defined in section 3.1.2
Robustness R	As defined in section 3.1.3

Table 3: Summary of independent and dependent variables related to measurement 1, 2 and 3.

5.1.7 Problems

During the user study I experienced three severe issues concerning the applied eye detection algorithm in the given setting: mascara, a large pupil diameter and IR distortions for measurement 3 (see Figure 21). Mascara and IR distortion are common problems to video-based eye detection in contrast to the novel phenomenon of a large pupil diameter. As can be seen in the figure, the dark pupil approach breaks, because IR light is reflected from within the eye as known from bright pupil detection, causing a white blob at the right edge. However, IR distortions in an indoor environment are unusual as well. An explanation can be found looking at the tabletop device used for measurement 3, a Samsung SUR40 multi-touch table. It emits IR light even if configured as a display causing a quadrilateral artifact across the pupil. Especially when changing the context, which means that there's no possibility to adapt the eye detection parameters, the robustness suffered.

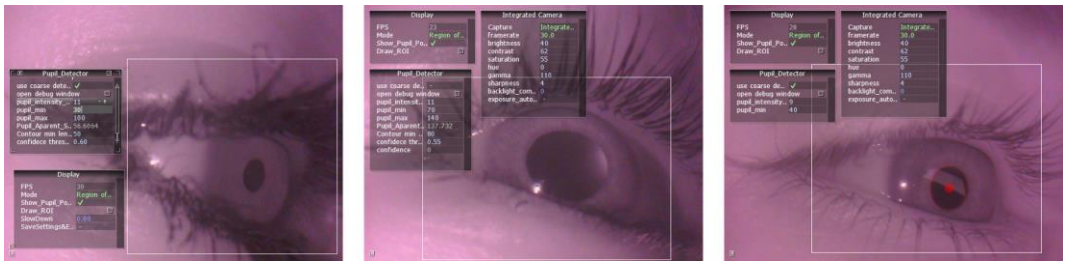

Figure 21: Obstacles experienced during the user study: Mascara avoiding rough pupil detection (left), a large pupil diameter causing IR reflections (middle) and IR artifacts caused by the applied tabletop device.

5.1.8 Results

All recordings were stored as raw samples, i.e. I first had to extract fixations to calculate the values of the dependent variables. Section 5.1.8.1 takes a closer look at my approach to extract fixations. Subsequently I delve into detail about the results of the four measurements. However, not all datasets were used for analysis due to the problems mentioned in section 5.1.7. I excluded participants with an overall robustness lower than 65% per measurement what accompanied with notes on problematic participants taken during the study. In total 10 participants remained for measurement 1 and 11 participants remained for measurement 2 & 3. Measurement 4 was not affected, since no eye tracker was used. Have in mind that results are reported with respect to the scene camera, either normalized or absolute with pixels as unit.

Measurement	*Condition*	*Fixations*	*Mean [px]*	*SD [px]*
Measurement 1	100%	492	28.313	13.891
	75%	476	28.233	19.823
	50%	425	28.953	22.815
Measurement 2	-150cm	484	34.251	20.114
	-100cm	532	30.534	15.292
	-50cm	540	27.845	14.996
	0cm	1636	25.033	14.68
	+50cm	551	27.924	15.65
	+100cm	559	32.408	16.345
	+150cm	547	36.79	15.78
Measurement 3	Tabletop	274	19.016	13.06
	Tabletop → Wall	167	44.847	19.718
	Wall → Tabletop	114	41.266	18.917
	Wall	297	18.613	12.429

Table 4: Mean and SD of spatial accuracy for the sub-conditions of measurement 1, 2 and 3, averaged over the corresponding scene targets in scene camera space.

Spatial Precision	*Measurement 1*	*Measurement 2*	*Measurement 3*
Fixations	1393	4847	852
Mean	1.424px	1.456px	1.609px
Standard Deviation	1.26px	1.253px	1.552px
Median	1.00px	1.081px	1.01px
90% Percentile	3.018px	2.835px	3.712px
95% Percentile	4.134px	3.77px	4.828px

Table 5: Descriptive statistics for spatial precision of measurement 1, 2 and 3, averaged over all sub-conditions.

5.1.8.1 Fixation Extraction

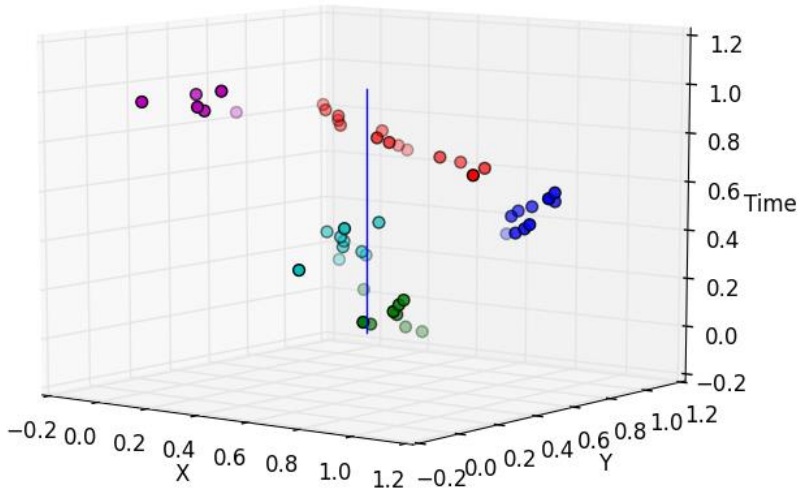

Figure 22: Exemplary visualization of the fixation extraction result. In total five fixations (dots with same color) were found surrounding a fixation target with a constant position (blue line).

Spatial accuracy and spatial precision are defined in terms of fixations (see section 3.1), that's why an offline processing was required to extract fixations for each condition and its scene targets. Basically a spatio-temporal clustering was applied, i.e. space and time were considered. Figure 22 shows the clustering result for one scene target of a participant. As you can see, five fixations were identified, highlighted by different colors. The X, Y and Time values were previously rescaled by the minimum and maximum values for each scene target.

5.1.8.2 Measurement 1: Extrapolation

I first analyzed the spatial accuracy for each sub-condition of the first measurement averaged over all scene targets. The means were 28.31px ($SD = 13.89$), 28.23px ($SD = 19.82$) and 28.95px ($SD = 22.82$) as summarized in Table 4. As can be seen from Figure 23, the variance increased from 192.97 over 392.95 to 520.52. A Levene's test showed that these differences in variance were significant ($p < 0.001, df = 1390$), but a Welch-ANOVA test showed no significant difference for the corresponding means ($p = 0.86, df = 851.888$).

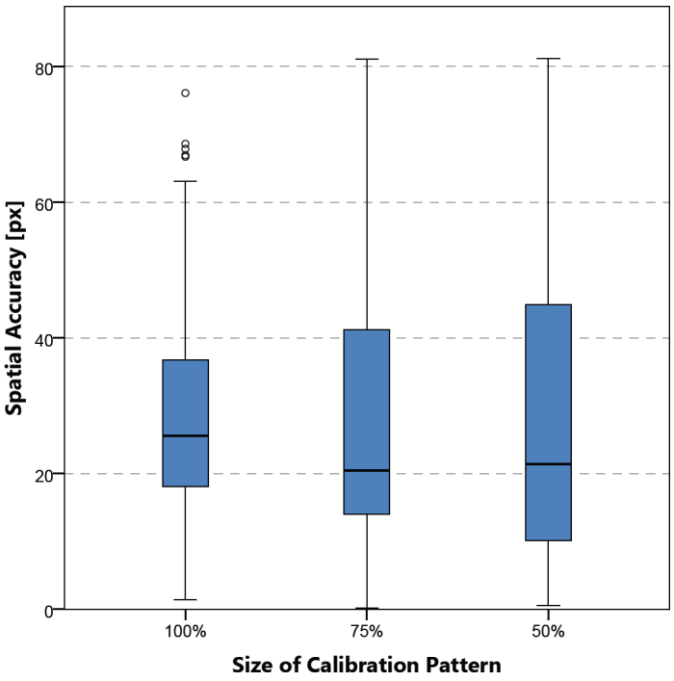

Figure 23: Boxplot showing the spatial accuracy averaged over all scene target locations for different sizes of the calibration pattern.

I therefore analyzed the gaze estimation error separately for each scene target location (see Figure 24). The spatial accuracy for the full-size pattern was evenly distributed across the camera's FOV, only the upper left corner showed a slight increase. However, for patterns 75% and 50% in size, the error at the scene camera borders increased by 33.15% and 56.18% respectively, while it decreased in the center region by 37.58% and 51.87% compared to the full pattern. A Welch-ANOVA test revealed that the differences were significant for the border regions ($p < 0.001, df = 441.747$) as well as for the center regions ($p < 0.001, df = 408.881$). The spatial precision for all fixations was 1.424px ($SD = 1.26px$) (see Table 5). In total 21450 samples have been considered of which 15467 were valid, resulting in a robustness of 72.11%.

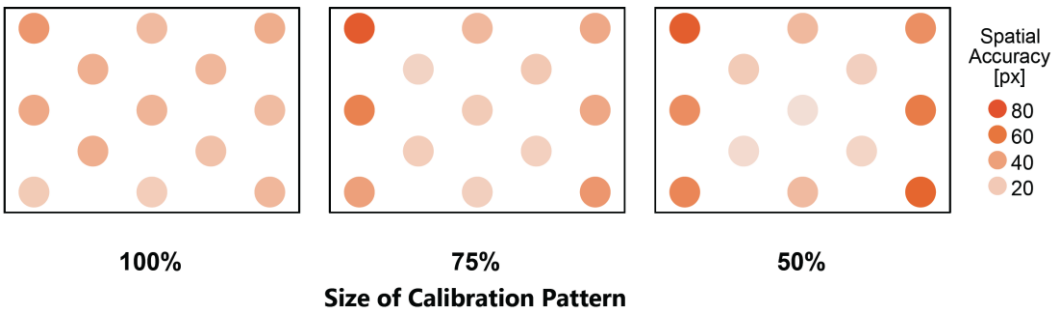

Figure 24: Heatmap showing the spatial accuracy in scene camera coordinate space for the 13 scene targets for different sizes of the calibration pattern.

5.1.8.3 Measurement 2: Parallax

As a first step I grouped the data with respect to the difference in calibration and recording distance $|d_{rec} - d_{cal}|$, ranging from -150 cm to +150 cm with 50 cm increments. Table 4 summarizes the corresponding mean and standard deviation values for spatial accuracy. As one can see in Figure 25 these means describe a valley with the values of -150 cm ($M = 34.25px, SD = 20.11px$) and +150 cm distances ($M = 36.79px, SD = 15.77px$) at its edges and the value of 0 cm ($M = 25.03px, SD = 14.68px$) at its center. A Welch-ANOVA test showed that the differences in spatial accuracy were significant ($p < 0.001, df = 1720.107$). Simplifying the distances to their absolute values yields that a movement of 50 cm after calibration results in a loss of spatial accuracy of 11.39% (25.81% for 100 cm, 42.21% for 150 cm). The average spatial precision for all fixations was 1.456px ($SD = 1.253px$) (see Table 5). In total 64350 samples have been considered of which 54978 were valid, resulting in a robustness of 85.436%.

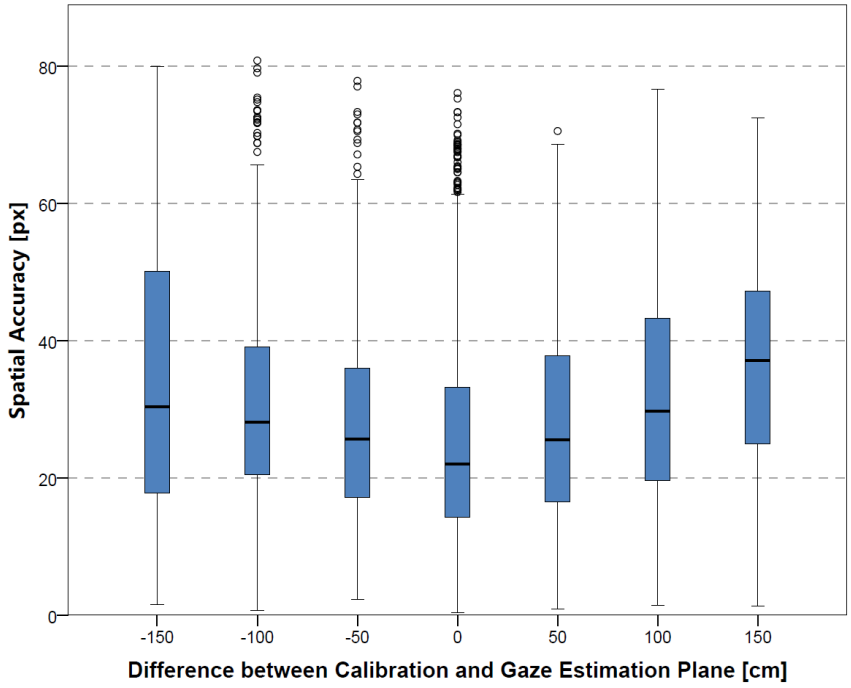

Figure 25: Boxplot showing the spatial accuracy averaged over all scene target locations for varying differences in calibration and recording distances.

5.1.8.4 Measurement 3: Switching Context

For this measurement I grouped the data considering the performed switch in context. Both recordings without a switch in context had similar values for spatial accuracy, 19.016px ($SD = 13.06px$) for the Tabletop and 18.613px ($SD = 12.429px$) for the wall-mounted display ($M = 18.806px$). A t-test showed that this difference is not significant ($p = 0.706, df = 569$). In contrast, I observed an error increase of 130.746% considering the mean spatial accuracy of recordings with a prior switch in context ($M = 43.394px$). A Welch-ANOVA test confirmed that the differences in means were significant ($p < 0.001, df = 335.409$). Table 4 provides a summary of all means and Figure 26 illustrates these values with the aid of a boxplot. The average spatial precision for all fixations was

1.609px ($SD = 1.552px$) (see Table 5). In total 15400 samples have been considered of which 9057 were valid. This results in an overall robustness of 58.812% for measurement 3. However, when regarding the sub-conditions without a switch in context, the robustness averages at 83.39% (6421 valid samples of 7700). In comparison the sub-conditions with a switch in context have a robustness of 34.234% (2636 valid samples of 7700).

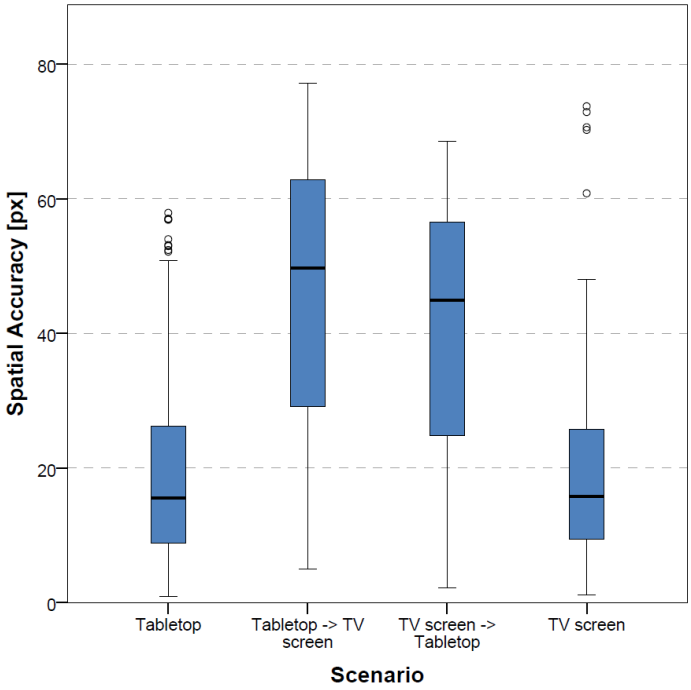

Figure 26: Boxplot showing the spatial accuracy averaged over all scene target locations for switching contexts.

5.1.8.5 Comfort-FOV

To analyze the comfort FOV I grouped the data by the rotation direction, i.e. the preset direction to which the participant had to rotate his eyes. In total 16 such orientations were defined starting at 0°, which describes the up-direction, and a maximum rotation of 337.5° (clockwise rotation). Figure 27 shows a boxplot averaging over all participants and a line indicating the corresponding mean of the comfort FOV values, 47.24°.

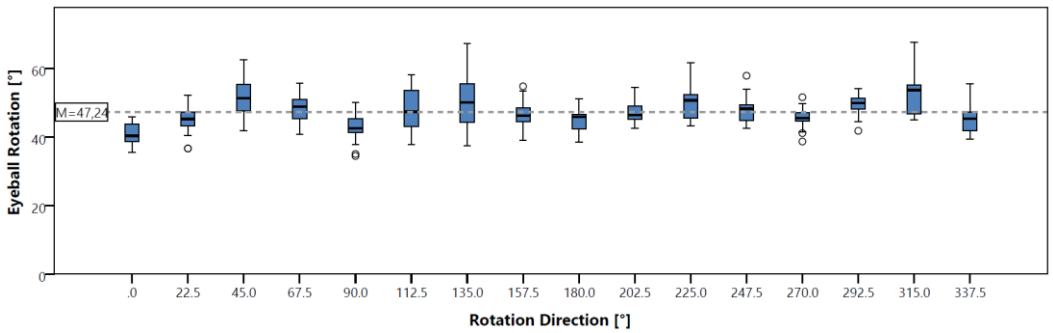

Figure 27: Boxplot illustrating the comfort FOV in degrees averaged over all participants on the x-axis. The y-axis indicates the orientation of the eyeball rotation where 0° is the up-direction and increasing values describe a clockwise rotation.

5.2 Display Detection and Mapping Error

Besides gaze estimation, an essential component for gaze-based interaction is detecting the interactive display and mapping gaze to that display. At the same time this phase contributes to the overall gaze estimation error. For that reason I conducted an experiment aiming at erroneous conditions of display detection. As a sample system I use a marker-based approach which is increasingly used for gaze-based interaction (see [18,37]). Since display detection is independent from the gaze estimation in general I investigate this component of the gaze estimation error in isolation and without participants. Following I provide a detailed explanation of how and what data were recorded and which findings could be derived.

5.2.1 Conditions

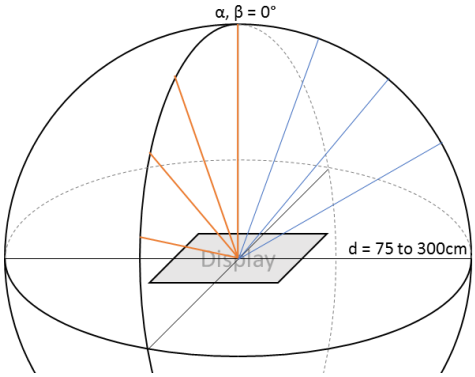

Figure 28: Positions for data recording including angle α around Y-axis, angle β around X-axis and the distance from camera to display center.

To determine the error originating from the display detection component I sampled data from different perspectives to the interactive display and with different marker patterns. Figure 28 illustrates the different recording positions determined by the distance of the scene camera to the fixation target d, the angles for the X-axis α (aka pitch) and the y-Axis β (aka yaw) with $d \in \{75cm, 100cm, 200cm, 300cm\}$ and $\alpha, \beta \in \{0°, 20°, 40°, 60°\}$. The angle for Z-axis (aka roll) was assumed to be zero based on the assumption that the user won't tilt his head to the left or to the right. To vary the size of the markers I applied three different marker patterns for each perspective and recording (see Figure 29). In total this results in 84 conditions $= 4\ distances * 3\ patterns * (3\ angles * 2\ axes + 1\ angle)$, where 0° is the same for α and β.

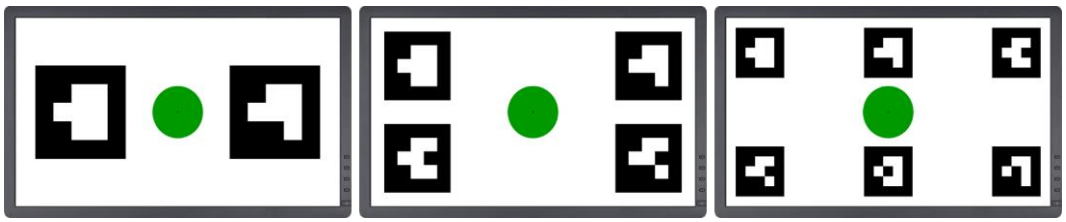

Figure 29: Three marker patterns with differences in size and position used for the display detection and mapping error measurement: two markers with an edge size of 750px (left), four markers with an edge size of 550px (middle) and six markers with an edge size of 400px (right).

5.2.2 Apparatus

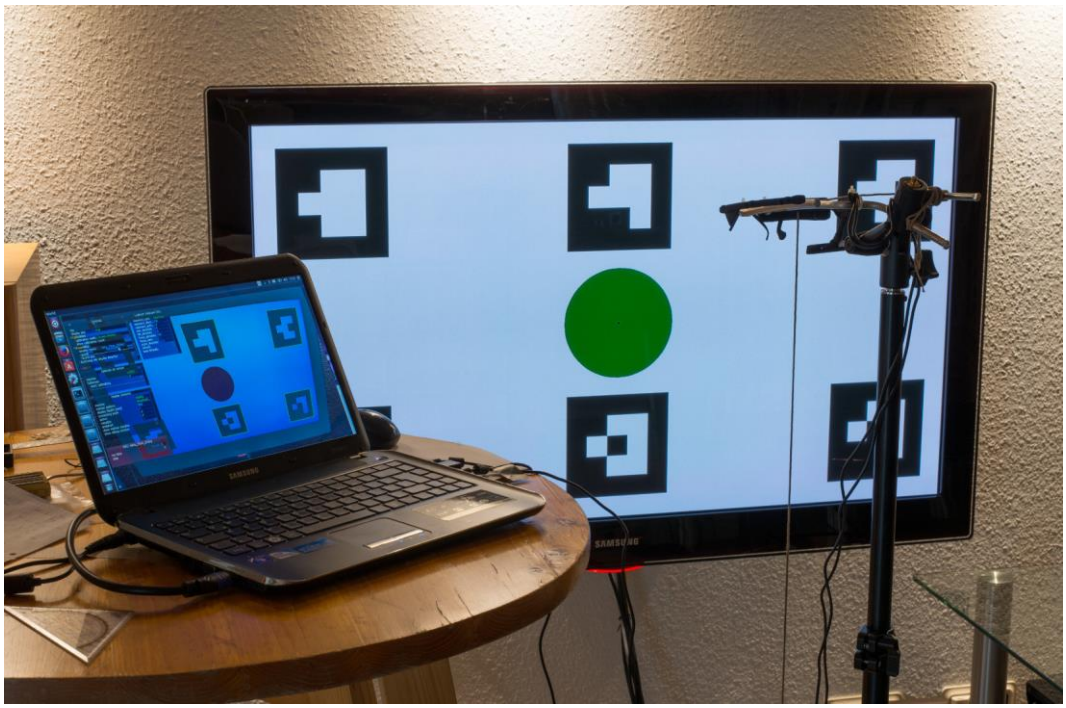

Figure 30: Setup for data recording with a 50-inch display, the tripod-mounted Pupil device and an evaluation laptop

The data recording was conducted using the Pupil eye tracker as well as a 50-inch wall-mounted flat screen with a resolution of 1920x1080 pixels with 17.42 pixel/cm ≈ 44.25 dpi (see Figure 30). For some trials the display had to be dismounted and was laid on the bottom, because of the restricted height of the ceiling. The eye tracker was mounted on a tripod that was adjustable in height. I used a plumb bob attached to that tripod to precisely position the tracker at predefined locations marked on the floor (see Figure 31). Due to the reflecting glass surface of the display the room was partially shaded during the record session. Further all recordings were performed using the Pupil Capture plugins as introduced in chapter 4.

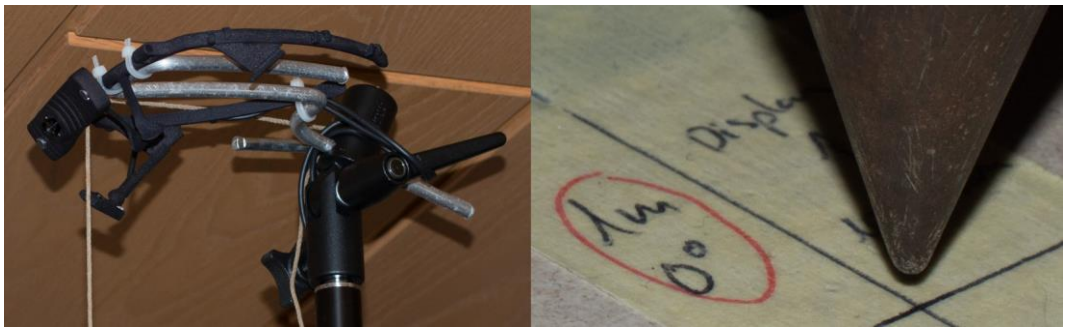

Figure 31: [Left] Pupil Pro eye tracker mounted on a tripod. [Right] Plumb bob attached to the tripod for precise positioning at predefined locations in front of the display.

5.2.3 Procedure

For each of the 84 conditions I recorded a dataset of 800 frames resulting in a total amount of 67200 samples. In a post-hoc analysis I manually annotated each sample with the display center in scene camera coordinates, which was highlighted by a green circle with a black dot at its center (see Figure 30). One frame per condition was used as reference. The display centers were mapped to display coordinate space with the homography matrix, which was automatically obtained during the recording session. Under ideal conditions for detection and mapping, these points should be mapped to the center of the display, which served as ground-truth for computing the mapping accuracy. Mapping accuracy is calculated as the difference between the ground-truth center point and the mapped center point for x and y respectively: $mapping\ accuracy_x = \left|center_{gt}^{x} - center_{mapped}^{x}\right|$. An overview of all independent and dependent variables can be found in Table 6. Since I experienced a rapidly shrinking detection rate for angles >60°, probably caused by the reflective surface of the display, higher angles were excluded.

Independent Variables

Distance d	$d \in \{75cm, 100cm, 200cm, 300cm\}$
Angle X-axis (pitch) α	$\alpha \in \{0°, 20°, 40°, 60°\}$
Angle Y-axis (yaw) β	$\alpha \in \{0°, 20°, 40°, 60°\}$
Size & Amount of Markers	As illustrated in Figure 29

Dependent Variables

Spatial Accuracy A	$mapping\ accuracy$

Table 6: Summary of independent and dependent variables of the measurement for the display detection and mapping error component.

5.2.4 Results

5.2.4.1 Mapping Accuracy

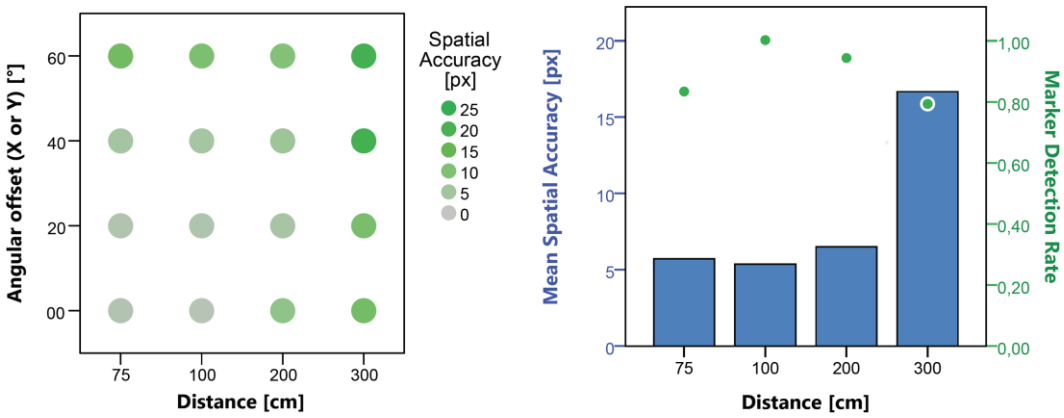

Figure 32: Gaze estimation error for the display mapping component for different angles and distances to the display in display coordinate space (left). Relations between the distance, marker detection rate and the distance (right).

Figure 32 (left) shows the error for mapping 2D gaze positions in scene camera coordinates to display coordinates for different angles (α, β combined) and distances to the display (see also Table 7 for detailed results). As you can see the error increases with both, increasing angle and distance. Figure 32 (right) puts emphasize on the relation of the distance, the marker-detection rate and the mapping error. The smallest error of 5.369px (SD=7.277px) was achieved for a distance of 100cm with a detection rate of nearly 100%. With increasing distance beyond 100cm the detection rate and the error increases. A visual analysis of the scene videos, stored with each dataset, revealed that the divergent behavior at 75cm was caused by the fact that not all markers were visible in the scene camera's field of view.

Measurement	*Condition*	*Samples*	*Mean [px]*	*SD [px]*
Angular Offset	0°	10705	5.766	5.448
	20°	24016	5.022	4.002
	40°	21109	8.818	21.765
	60°	21534	13.36	12.209
Distance	75cm	21544	5.712	8.789
	100cm	18613	5.369	7.278
	200cm	18417	6.509	3.418
	300cm	18790	16.673	23.359

Table 7: Mean and SD of gaze mapping accuracy different angles and distances towards the display in display coordinates (1920x1080px, 44.25dpi).

5.2.4.2 Pose Estimation

It emerged that the marker-based pose estimation worked well. In Table 8 you can see the differences between the approximated pose and the actual one. 95% of the differences in distance are below 1.9cm, for rotation around X-axis and Y-axis values are below 2.8° and 1.7° respectively.

Difference to real pose	d	α	β
Samples	77364		
Mean	0.992cm	1.430°	0.802°
Standard Deviation	0.995cm	0.793°	0.558°
Median	0.787cm	1.233°	0.787°
95% Percentile	1.888cm	2.745°	1.622°

Table 8: Separate evaluation of the performance of the marker-based pose estimation in terms of differences in distance, rotation around X-axis and rotation around Y-axis.

6 Evaluation of the Combined Error Model

Previously I investigated the parallax error, the extrapolation error as well as the error generated by display detection and mapping and how they contribute to the overall gaze estimation error. Based on this knowledge and using the data corpus as a foundation I generated separate error prediction models, one for the gaze estimation part and one for the display detection and mapping part. As proposed by [15] I further split both models in their horizontal (X) and vertical (Y) error component. To get a combined error model for each direction I merged the particular horizontal and vertical components. This chapter puts emphasize on the building process of the models and on its performance.

6.1 Method

To build the combined error prediction model, I trained two support vector regression (SVR) models with a radial basis function (RBF) as kernel: one for the gaze estimation part, where the corpus of measurement 1 and measurement 2 of the user study served as input, and one for the display detection and mapping part, where all samples of the second data recording were taken into account. There are three parameters, which have to be determined beforehand, the cost parameter C, the width of the radial basis function γ and the value of the insensitive zone ϵ, i.e. deviations smaller than ϵ do not contribute to the costs. Following common practice in machine learning, I optimized these parameters using a grid search on a random 10% subset of all data (see Table 9 for results). I started with a coarse grid and iteratively refined the limits until no further change was recognizable. The remaining 90% of the data were partitioned in a training and a test set (70%/30%) used to train and evaluate the optimized model.

The individual models were unified by means of a software module, consecutively predicting the gaze estimation component, then the display detection and mapping component of the error. The first component reports the error in scene camera coordinates that I further transferred to display coordinate space with a distance dependent mapping. Eventually I summed up both components to get the overall gaze estimation error in display coordinate space.

		Gaze Estimation		*Display Detection and Mapping*	
		Horizontal (X)	*Vertical (Y)*	*Horizontal (X)*	*Vertical (Y)*
Parameters	C	32	32	181.02	32
	γ	4	64	4	2.83
	ϵ	8	0.00098	1	0.5

Table 9: Optimized parameters of horizontal and vertical SVR models resulting from a grid search on randomly chosen 10% subsets of the corresponding data corpus.

6.2 Results

Individual evaluation results are reported in terms of the root mean square error (RMSE) of the residuals, i.e. the RMSE of the differences between the predicted and the actual values, and R^2, i.e. the portion of the data variance that is explained by the model. This test was repeated 50 times to balance noise due to the random sample selection. I further added the different parameters incrementally to observe their influence on the error prediction performance.

The performance curve of the first error component, namely gaze estimation, divided in horizontal and vertical fraction is shown in Figure 33. The performance denoted in terms of RMSE of residuals and R^2 continuously increases as more parameters are added. One exception is the last parameter of the y-model that caused a small performance decrease of 4%. However, the overall improvement in model performance was 26.83% for the x-model and 50.34% for the y-model. Hereby the mean RMSE decreases to 11.89px for x and to 6.94px for y, R^2 grows up to 46.19% for the x-model and to 74.76% for the y-model.

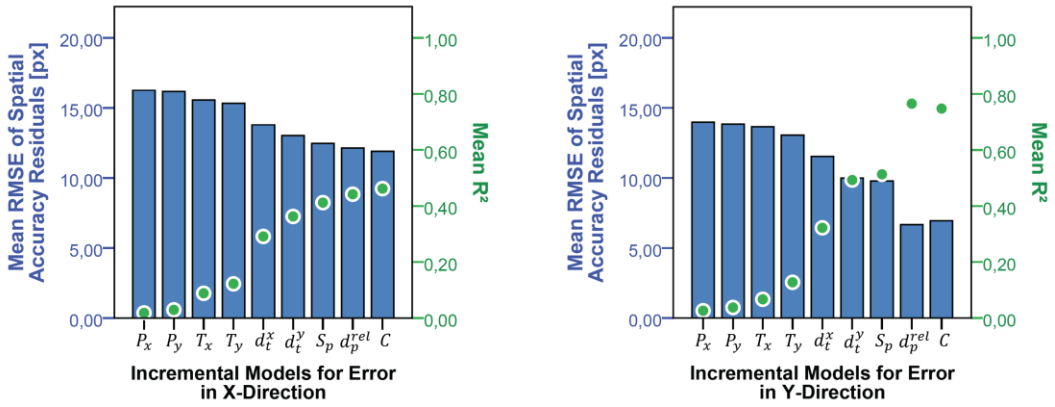

Figure 33: Error prediction performance of the Gaze Estimation component for x and y in scene camera space (1280x720 pixels).

The evaluation of the second error component, which covered the display detection and mapping, was performed in the same way. Figure 34 shows a continuously increasing performance for an increasing amount of parameters as we have seen for the first component. Here, the overall improvement in model performance was 64.65% for the x-model and 56.52% for the y-model, where the RMSE decreases to 4.72px for x and to 2.26px for y, R^2 reaches 87.35% for the x-model and 84.56% for the y-model.

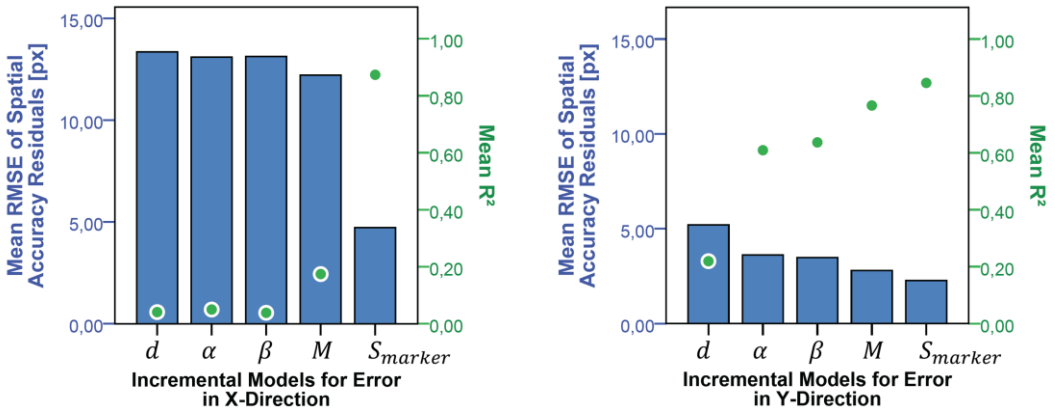

Figure 34: Error prediction performance of the Display Detection and Mapping component for x and y in display coordinate space (1400x1050 pixels; 267x200 cm).

Based on the individual performance of the respective horizontal and vertical models I evaluated the combined error model in terms of their summarized RMSE values in display coordinate space. Figure 35 shows the constantly growing error for increasing distance to the display. On average the model accuracy ranges from 3.96px (50cm) to 23.74px (300cm) for the x component and from 2.36px (50cm) to 14.19px (300cm) for the y component. The reference display has a resolution of 1400x1050 pixels and dimensions of 267x200cm, i.e. the error in degrees of visual angle yields 0.86° for x and 0.52° for y. The Euclidean distance serves as overall performance indicator, resulting in a joined accuracy of 1.01°. In addition, I compared my model against two baseline approaches for error prediction. The naïve approach Best assumes a constant error of 0.6°, which is reported as best-case spatial accuracy of the Pupil eye tracker. The naïve approach Measured incorporates the mean error in visual degrees extracted from my measurements that was 1.26° for x and y. Both naïve approaches take the distance towards the interactive display into account. As can be seen in Figure 35 my model performs better than both naïve approaches.

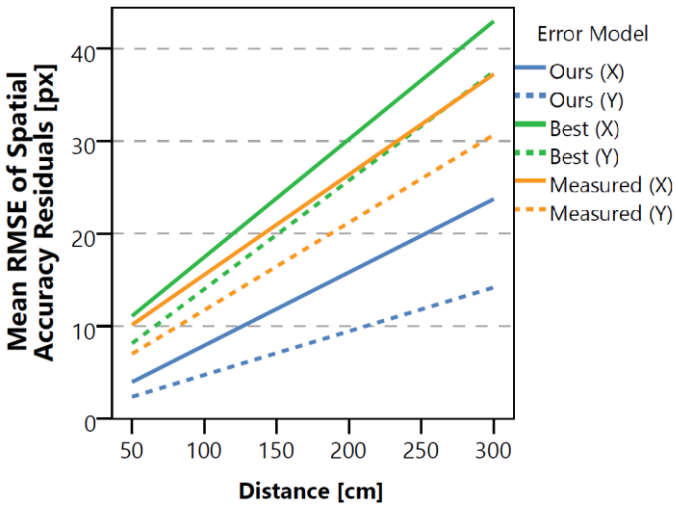

Figure 35: Error prediction performance of the combined error model (Ours) compared to the naïve model Best and the naïve model Measured. The performance is reported in display coordinate space as residuals, i.e. the differences between the estimated and the observed gaze estimation error. The pixel density was 5.24px/cm.

7 Discussion

Following I discuss the outcomes of my work, that is the conducted measurements and their results as well as the combined error model and its evaluation. Further I derive guidelines to inform the interface design for gaze-enabled interactive systems and hint at use cases applying those guidelines. Lastly I point out limitations of my work and come up with solutions for future work.

7.1 Measurements

Measurement 1 focusing on the extrapolation error showed that the overall spatial accuracy does not change significantly as a function of the relative calibration pattern size S_p (see Figure 23), i.e. the first part of the hypothesis H1 stated in section 3.3.2 does not hold. However, with decreasing pattern size, the error significantly increased at the borders by 56.18% and decreased at the center region by 51.87% (see Figure 24). While the error increase is caused by the gaze estimation system having to extrapolate considerably outside the area covered by the calibration pattern, the error decrease is caused by a more dense calibration pattern, activating a more accurate interpolation within that area. This observation is evidence for the second part of the hypothesis H1 and further explains the differences in variance as well as the steady overall gaze estimation error.

The results of measurement 2 showed that the parallax issue is indeed a significant source of error for gaze estimation with monocular head-mounted eye trackers. Besides a confirmation of previous findings on that topic, I quantified the impact of the parallax issue as can be seen in Figure 25. I further showed that moving merely 50cm from the calibration position yields an 11.39% error increase, growing up to 42.21% for 150cm. In summary the hypothesis H2 on the parallax error turns out to be true.

Prior to each calibration the eye tracker settings were adjusted to the current context, i.e. to the applied display and its properties and surroundings. The results of measurement 3 showed that changing the context during gaze estimation yields a significant and severe loss of spatial accuracy, i.e. the gaze estimation error increased by 130.746%.I observed a similar behavior for the robustness, which drops 58.947%, if there was a context switch. Especially the robustness suffers from the IR distortion of the tabletop device.

Measurement 4 showed that on average the participants were able to rotate their eyeballs 47.24° in each direction, without losing comfort. As we know from section 4.1 the scene camera covers ±39.08° on the horizontal, ±25.08° on the vertical axis and ±46.435° on the diagonals. Apparently the range of the user's eye ball exceeds the scene camera's FOV. However it is still unclear if the participants would have rotated their eyeballs that much to fixate a target. This could be investigated in a further experiment.

The measurement on display detection and gaze mapping showed that both, the distance to the display and the rotation angle are significant factors for the gaze estimation error (see Figure 32 left). In addition I observed the negative influence of low marker detection rates. Even at 75cm, where the detection quality is assumed to be best, the low detection rate causes a reduction of accuracy. Finally the corresponding hypothesis H3 could be proven.

7.2 Combined Error Model

Informed by these measurements I presented a computational model activating forecasts about the gaze estimation error of monocular head-mounted eye trackers in context of gaze based interaction. To the best of my knowledge, this was the first attempt to build such a model, characterize its input parameters, evaluate the prediction performance and outline potential use cases.

To evaluate the set of parameters listed in Table 2, I incrementally added them to a support vector regression model and observed the model performance in terms of the root mean squared error and R^2 (see section 6.2). For the gaze estimation component of the model, the proposed scene target to calibration pattern relation d_t^x and d_t^y as well as the relative offset to the calibration position d_p^{rel} definitely enhanced the error prediction performance (see Figure 33). For display detection and gaze mapping, the metric S_{marker} revealed to be important (see Figure 34).

I further reviewed the combined error model, merging both individual components. The results suggest that the set of parameters is comprehensive and allows the model to predict the gaze estimation error with a root mean squared error of 1.01 degrees of visual angle (X & Y model combined). In comparison to the two baseline approaches the model appears competitive. Figure 35 illustrates that the combined error model outperforms the baseline methods Best – assuming a constant best-case error of 0.6° – and Measured – taking the spatial accuracy into account that was achieved during the measurements.

7.3 Guidelines and Use Cases

In order to design interfaces it is helpful to follow guidelines that cope for the most common issues and assist in reaching usability and user experience goals. In section 7.3.1 I outline important aspects of gaze-based interaction with the target to inform interaction designers. In addition I provide exemplary use cases in section 7.3.2 illustrating how the proposed error model can be applied in real-time to enhance gaze-based interfaces.

7.3.1 Guidelines for Mobile Gaze-based Interaction

It emerged that the extrapolation error, the parallax error, the application context as well as the display detection play an important role for the gaze estimation error. Following I summarized my findings on these factors in Table 10 to guide interaction designers in the field of gaze-based interaction.

1st. Be Aware of the Gaze Estimation Error.

- Incorporate the inevitable gaze estimation error in your interface design. E.g. enlarge selection targets dependent on the error or move small selection targets to high accuracy regions on the screen.
- Make sure that the calibration routine of the eye tracker covers the most important regions of the gaze-based interface to reduce the extrapolation error. Remember that a dense calibration pattern enhances gaze estimation results.

2nd. Take Care of the User's Movement.

- After calibrating the eye tracker the distance of the user to any fixation target should at maximum differ ±50cm from the calibration distance. This keeps the increase of the parallax error below 12%. For each further distance increase of 50cm the error roughly doubles.

3rd. Consider the Alignment between User and Interface.

- Keep the maximal interaction distance below 200cm. A further 100cm results in a triplication of the error originating from display detection. A more sophisticated display detection algorithm as well as a better scene camera might increase the maximal distance.
- Allow gaze-based interaction in a range of ±40° towards a display (combined rotation). The combined rotation is calculated as the Euclidean distance of α and β angles.
- Distribute the visual markers such that at least one is visible to the scene camera for all possible user locations, i.e. a proper display detection and gaze mapping is enabled. For more robust results, three markers forming a right-angled triangle should be visible all the time.

4th. Keep the Context Constant.

- Once calibrated, the eye tracker should be used for the current context only. Otherwise the spatial accuracy drops severely. If it is necessary to change the context consider to re-calibrate the eye tracker.

5th. Manage the User's Expectations with Feedback

- Provide unobtrusive and informative feedback to the user that enables better individual performance with less frustration. Help the user to understand achievements and mistakes and to gradually generate a deeper understanding of the system. Examples are feedback about the marker detection rate or indicating the current gaze estimation error.

Table 10: Guidelines for mobile gaze-based interaction with monocular eye trackers.

7.3.2 Use Cases of the Gaze Estimation Error Model

Embracing the previously introduced guidelines as well as the real-time error prediction, facilitates error-aware gaze-based interaction and the possibility to simulate gaze-enabled interactive systems. Following I describe exemplary use cases for application that shall help interaction designers to meet these guidelines and that can be included as part of a user interface.

7.3.2.1 Uncertainty Indicator

The general idea of the uncertainty indicator is to augment a gaze pointer by a semi-transparent ellipse indicating the currently predicted gaze estimation error. The ellipse enables the user to instantly know about the error and to implicitly cope for it. In addition the user learns about the limitations of gaze estimation helping him to gradually increase his mental model for gaze-based interaction. Mardanbegi et al. [19] already introduced this concept, but without providing any method for predicting the gaze estimation error. From the technical point of view, the uncertainty indicator is generated by predicting the error for X- and Y-direction for each fixation and incorporating the Z-rotation as rotation of the ellipse. The real-time capability was successfully tested by means of a prototype (see Figure 36), i.e. Pupil Capture continued grabbing 30 frames per second with active feedback.

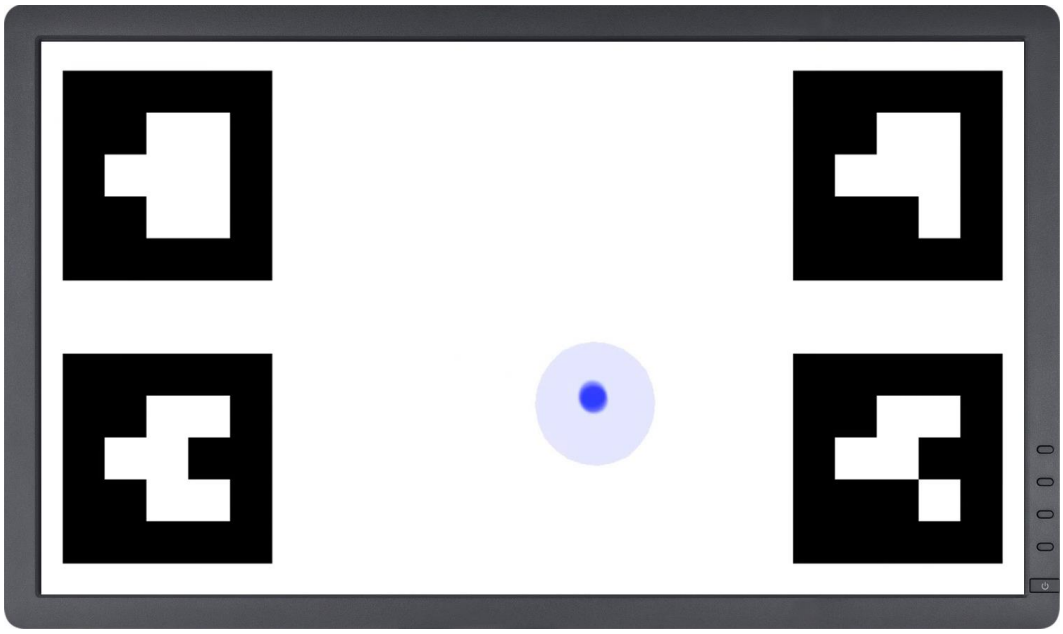

Figure 36: Screenshot of the ellipsoid uncertainty indicator, giving real-time feedback about the currently predicted gaze estimation error.

7.3.2.2 Display Detection Feedback

Display detection is an essential part of selective gaze-based interaction, because it links the gaze position in scene camera coordinates to an on-screen position. Providing feedback about the quality of display detection is therefore important for the user such that he can adapt to that error or change his position to enhance display detection.

7.3.2.3 Error-aware Target Resizing

This concept describes an adaptive interface, whose interactive targets dynamically increase or decrease in size dependent on the predicted gaze estimation error. First, this approach shall increase the selection accuracy and second, the target size indirectly conveys the gaze estimation error with a similar effect on the user's mental model as the uncertainty indicator. However, the impact on selection accuracy has to be shown.

7.3.2.4 Error-aware Target Alignment

If resizing of interactive targets is not an option, reordering them according to the error distribution on the screen can be a solution. However, one should not break the general logic behind a target alignment to prevent frustration. One possibility could be to reorder groups of icons or similar, where the group itself is easy to find and all contained targets have the same size.

7.3.2.5 Heatmap Overlay

The heatmap overlay is intended to be a tool for interaction designers, by providing fast access to accuracy information across the whole target display. It can be used to identify high- and low-accuracy regions of a display in order to arrange selection targets of different size or to evaluate existing interfaces with regard to gaze-based interaction. A similar technique is used by Kassner et al. [18] to illustrate gaze hotspots, i.e. regions where people look more frequently. To generate the heatmap overlay, the gaze estimation error is predicted for a regular $N \times M$ grid across the target display. Further the estimates are converted to a common heatmap color space reaching from blue (low error) to red (high error). The emerging texture with size $N \times M$ is then scaled with nearest neighbor interpolation and mapped to the target display within the scene camera view. In Figure 37 you can see a working prototype calculating and visualizing a heatmap of size 8x4 in real-time.

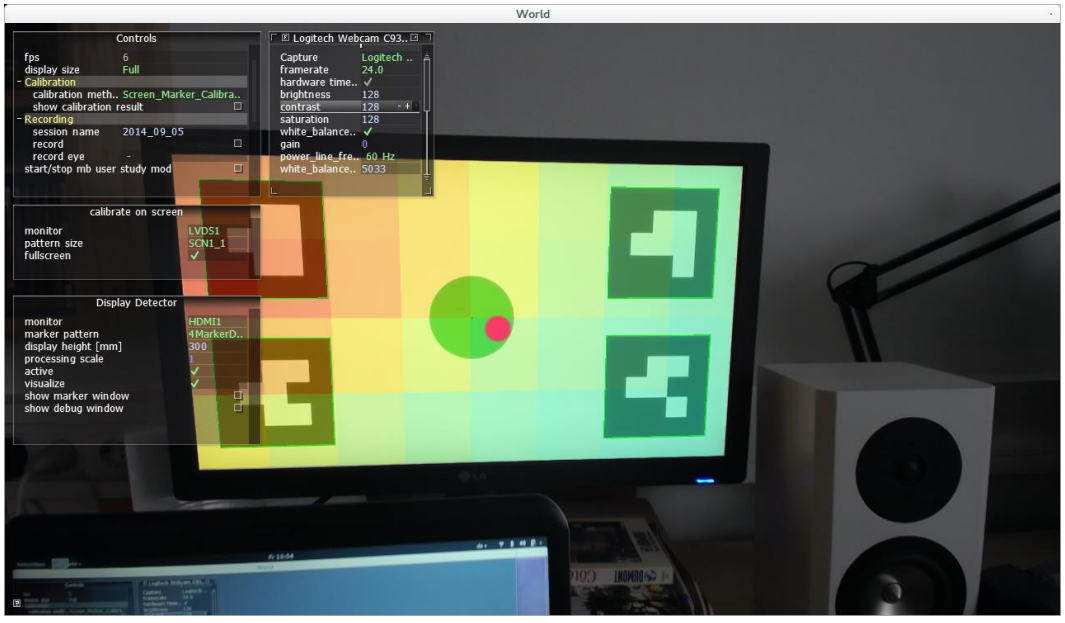

Figure 37: Screenshot of the scene camera view with active 8x4 heatmap, indicating the real-time error across the whole target display.

7.4 Limitations and Future Work

Even if the proposed error prediction model is flexibly applicable, several limitations remain and need to be addressed. Limitations concern the procedure of error prediction, its portability between different eye tracking devices and the integrity of the included error sources. Finally it has to be shown that error-awareness, enabled by the proposed model, actually improves gaze-based interaction.

7.4.1 Enhancing the Error Prediction Procedure

The current error prediction procedure embraces four individual error models that need to be trained and applied: the gaze estimation component and the display detection and gaze mapping component, both split in X and Y (see section 3.3). It would be preferable to have a single multivariate regression model summarizing those individual parts, especially in consideration of the generalizability of the model.

7.4.2 Generalizing the Model

The error prediction model was developed and evaluated for a specific setup, i.e. an eye tracker, a display as well as a marker detection and tracking library. However, the modeling approach is generic and applicable to other types of equipment and target systems. But, all measurements needed to adapt and train the model, currently have to be performed manually. A solution would be to have an automatic model fitting procedure, capable of generating a model for any given setup. Aiming at a single regression model as described in the previous section would further simplify this process. Besides including other brands an extension for future work could be to apply the modeling process on binocular head-mounted and stationary remote trackers.

7.4.3 Include further Error Sources

A further limitation to the model is that the concerned error sources are not extensive. E.g. a displacement of an eye tracker is a likely source of error in real-world settings, but currently not covered by the proposed model. Headset movements can occur when using an eye tracker for longer periods of time or during physical activities. Fast head movements further lead to motion blur, which might have a negative impact on marker detection, especially when using interlaced images.

7.4.4 Validate the Model's Usability

An essential part of future work should be to study the use of the proposed model for adapting interfaces and for improving gaze-based interaction dependent on the current gaze estimation error. This should comprise the evaluation of the use cases mentioned in section 7.3.2 and a deeper investigation of the model's real-time capabilities. Another interesting point for future work would be to investigate the feasibility of real-time error compensation with the aid of the proposed model (see section 2.2.3).

8 Conclusion

Within this work I presented a computational model for predicting the gaze estimation error of head-mounted monocular eye trackers in real-time. It takes relevant factors as input parameters, such as the user's 3D pose, and computes the suspected spatial accuracy that can be used to inform novel gaze-based interaction. To accomplish that, I performed carefully executed measurements delivering insights into the individual error caused by gaze estimation and display mapping. In parallel these data served as input for the modeling process. Subsequently I evaluated the prediction performance of the emerging error model in terms of the root mean square error of spatial accuracy residuals, i.e. the RMSE of the differences between the observed and the estimated spatial accuracy. The combined result of 1.01°, which outperforms considered naïve approaches, is promising for a new generation of gaze-based interfaces, which are aware of the inevitable gaze estimation error. A set of guidelines and use cases was included in order to help interaction designers in achieving their usability and user experience goals for these future interfaces.

BIBLIOGRAPHY

[1] Daniel F. Abawi, Joachim Bienwald, and Ralf Dörner, "Accuracy in Optical Tracking with Fiducial Markers: An Accuracy Function for ARToolKit," in *ISMAR '04 Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2004, pp. 260-261.

[2] Deepak Akkil, Poika Isokoski, Jari Kangas, Jussi Rantala, and Roope Raisamo, "TraQuMe: a tool for measuring the gaze tracking quality," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, March 2014, pp. 327-330.

[3] Stanislav Bardins, Tony Poitschke, and Stefan Kohlbecher, "Gaze-based interaction in various environments," in *Proceedings of the 1st ACM workshop on Vision networks for behavior analysis*, Vancouver, 2008, pp. 47-54.

[4] Pieter Blignaut and Tanya Beelders, "TrackStick: a data quality measuring tool for Tobii eye trackers," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, March 2012, pp. 293-296.

[5] Jurek Breuninger, Christian Lange, and Klaus Bengler, "Implementing gaze control for peripheral devices," in *Proceedings of the 1st international workshop on pervasive eye tracking & mobile eye-based interaction*, 2011, pp. 3-8.

[6] Andreas Bulling and Hans Gellersen, "Toward Mobile Eye-based Human Computer Interaction," *Pervasive Computing, IEEE*, vol. 9, no. 4, pp. 8-12, 2010.

[7] Andreas Bulling, Daniel Roggen, and Gerhard Tröster, "EyeMote - Towards Context-Aware Gaming Using Eye Movements Recorded from Wearable Electrooculography," in *Proceedings of the 2nd International Conference on Fun and Games*, September 2008, pp. 33-45.

[8] Andreas Bulling, Jamie A Ward, Hans Gellersen, and Gerhard Tröster, "Eye movement analysis for activity recognition," in *Proceedings of the 11th international conference on Ubiquitous computing*, 2009, pp. 41-50.

[9] Guy Thomas Buswell, "How people look at pictures: a study of the psychology and perception in art," 1935.

[10] Juan J. Cerrolaza, Arantxa Villanueva, Maria Villanueva, and Rafael Cabeza, "Error characterization and compensation in eye tracking systems," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, March 2012, pp. 205-208.

[11] Jan Drewes, Guillaume S Masson, and Anna Montagnini, "Shifts in reported gaze position due to changes in pupil size: Ground truth and compensation," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2012, pp. 209-212.

[12] Andrew T. Duchowski, "A Breadth-First Survey of Eye Tracking Applications," *Behavior Research Methods, Instruments, & Computers*, vol. 34, no. 4, pp. 455-470, 2002.

[13] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280-2292, 2014.

[14] Dan Witzner Hansen and Qiang Ji, "In the Eye of the Beholder: A Survey of Models for Eyes and Gaze," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 478-500, March 2010.

[15] Kenneth Holmqvist, Marcus Nyström, and Mulvey Fiona, "Eye tracker dataquality: what it is and how to measure," in *Proceedings of the symposium on eye tracking research and applications*, March 2012, pp. 45-52.

[16] Anthony J Hornof and Tim Halverson, "Cleaning up systematic error in eye-tracking data by using required fixation locations," *Behavior Research Methods, Instruments, & Computers*, vol. 34, no. 4, pp. 592-604, 2002.

[17] Samuel John, Erik Weitnauer, and Hendrik Koesling, "Entropy-based correction of eye tracking data for static scenes," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2012, pp. 297-300.

[18] 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*, Seattle, Washington, 2014, pp. 1151-1160.

[19] Diako Mardanbegi and Dan Witzner Hansen, "Mobile gaze-based screen interaction in 3D environments," in *Proceedings of the 1st conference on novel gaze-controlled applications*, 2011, p. 2.

[20] Diako Mardanbegi and Dan Witzner Hansen, "Parallax error in the monocular head-mounted eye trackers," in *Proceedings of the 2012 acm conference on ubiquitous computing*, September 2012, pp. 689-694.

[21] Darius Miniotas, Oleg Špakov, and Scott I MacKenzie, "Eye gaze interaction with expanding targets," in *CHI'04 extended abstracts on Human factors in computing systems*, 2004, pp. 1255-1258.

[22] Akito Monden, Ken-ichi Matsumoto, and Masatake Yamato, "Evaluation of gaze-added target selection methods suitable for general GUIs," *International Journal of Computer Applications in Technology*, vol. 24, no. 1, pp. 17-24, 2005.

[23] Marcus Nyström, Richard Andersson, Kenneth Holmqvist, and Joost van de Weijer, "The influence of calibration method and eye physiology on eyetracking data quality," *Behavior research methods*, vol. 45, no. 1, pp. 272-288, 2012.

[24] OpenCV. (2015, Feb.) Camera calibration With OpenCV - OpenCV 2.4.9.0 documentation. [Online]. http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

[25] Pupil Labs. (2014, Aug.) Home · pupil-labs/pupil Wiki · GitHub. [Online]. https://github.com/pupil-labs/pupil/wiki

[26] SensoMotoric Instruments GmbH. (2014, June) SensoMotoric Instruments GmbH > Gaze and Eye Tracking Systems > Products > Overview. [Online]. http://www.smivision.com/en/gaze-and-eye-tracking-systems/products/overview.html

[27] Jeffrey S Shell, Roel Vertegaal, and Alexander W Skaburskis, "EyePliances: attention-seeking devices that respond to visual attention," in *CHI'03 extended abstracts on Human factors in computing systems*, 2003, pp. 770-771.

[28] John D Smith, Roel Vertegaal, and Changuk Sohn, "ViewPointer: lightweight calibration-free eye tracking for ubiquitous handsfree deixis," in *Proceedings of the 18th annual ACM symposium on User interface software and technology*, 2005, pp. 53-61.

[29] Oleg Špakov, "Comparison of eye movement filters used in HCI," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2012, pp. 281-284.

[30] Oleg Špakov, "Comparison of gaze-to-objects mapping algorithms," in *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications*, 2011, p. 6.

[31] Oleg Špakov and Yulia Gizatdinova, "Real-time hidden gaze point correction," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2014, pp. 291-294.

[32] Edward Llewellyn Thomas, "Movements of the Eye," *Scientific American Magazine*, August 1968.

[33] Tobii Technology. (2014, June) Eye Tracker: products and systems. [Online]. http://www.tobii.com/en/eye-tracking-research/global/products/

[34] Vytautas Vaitukaitis and Andreas Bulling, "Eye gesture recognition on portable devices," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 711-714.

[35] Roel Vertegaal, "Attentive user interfaces," *Communications of the ACM*, vol. 46, no. 3, pp. 30-33, 2003.

[36] Mélodie Vidal, Andreas Bulling, and Hans Gellersen, "Pursuits: spontaneous interaction with displays based on smooth pursuit eye movement and moving targets," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, September 2013, pp. 439-448.

[37] Lawrence H. Yu and E. Eizenmann, "A new methodology for determining point-of-gaze in head-mounted eye tracking systems," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 10, pp. 1765-1773, October 2004.

[38] Yanxia Zhang, Andreas Bulling, and Hans Gellersen, "SideWays: a gaze interface for spontaneous interaction with situated displays," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 851-860.

[39] Yunfeng Zhang and Anthony J Hornof, "Easy post-hoc spatial recalibration of eye tracking data," in *ETRA*, 2014, pp. 95-98.

[40] Yunfeng Zhang and Anthony J Hornof, "Mode-of-disparities error correction of eye-tracking data," in *Behavior research methods*.: Springer, 2011, pp. 834-842.

[41] Xinyong Zhang, Xiangshi Ren, and Hongbin Zha, "Improving eye cursor's stability for eye pointing tasks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 525-534.