

Submitted by
Michaela Murauer, BSc.

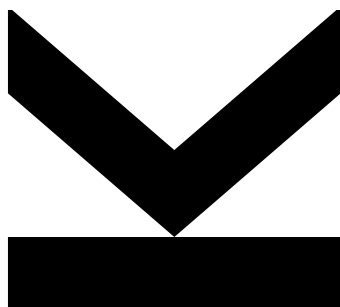
Submitted at
Institute for
Pervasive Computing

Supervisor
Univ.-Prof. Mag. Dr.
Alois Ferscha

Co-Supervisor
DI Michael Haslgrübler

November 2017

Natural Pursuits for Eye Tracker Calibration



Master Thesis
to obtain the academic degree of
Diplom-Ingenieurin
in the Master's Program
Computer Science

Abstract

Although, gaze-based interaction has been investigated since the 1980s and remains a promising concept to support universal interaction within distributed IoT environments, the main challenges, such as the Midas touch problem [Jac91] or calibration are still frequent topics of research. In this thesis, Natural Pursuit Calibration is presented, which is a comfortable, unobtrusive technique enabling ongoing attention detection and eye tracker calibration within an off-screen context. The user is able to perform calibration, without a digital user interface, artificial annotation of the environment and further assistance, by simply following any arbitrary moving target. Due to the characteristics of the calibration process, it can be executed simultaneously to any primary task, without active user participation. This then results in a frequently updated calibration model. A two-stage evaluation process is conducted to (i) optimize parameter settings in a first setup and (ii) compare the accuracy as well as the user acceptance of the proposed procedure to prevailing calibration techniques in an extensive user study.

Kurzfassung

Augen-basierte Interaktion wird seit den 1980er Jahren erforscht und stellt im Zeitalter von verteilten IoT Umgebungen ein vielversprechendes Konzept dar, um universielle Interaktion zu realisieren. Dennoch gibt es noch keine standardisierten Lösungsansätze für grundsätzliche Herausforderungen wie das 'Midas touch'-Problem [Jac91] oder Kalibrierung. In dieser Arbeit wird Natural Pursuit Kalibrierung vorgestellt, eine unaufdringliche Kalibrierungstechnik, die fortlaufende Aufmerksamkeitserkennung und Eye-Tracker Kalibrierung ermöglicht. Der Benutzer kann eine Kalibrierung ohne jegliche Benutzeroberfläche, ohne künstliche Annotationen der Umgebung und ohne fremde Unterstützung durchführen, in dem er einfach einem beliebigen bewegten Ziel mit dem Blick folgt. Aufgrund der Charakteristik des vorgestellten Kalibrierungsprozess ist es möglich, ein ständiges Update der Kalibrierung im Hintergrund einer beliebigen primären Aufgabe durchzuführen, ohne dass sich der Benutzer aktiv daran beteiligen muss. Ein zweistufiger Evaluierungsprozess wurde durchgeführt um (i) ein optimales Parameterset zu finden und (ii) die Genauigkeit und Benutzerakzeptanz der vorgestellten Technik mit vorherrschenden Kalibrierungsarten zu vergleichen.

Acknowledgement

First of all, I would like to thank my thesis supervisor Alois Ferscha, for his steady support, his trust in my thoughts and ideas and the great opportunity of working at his institute.

A very special gratitude goes to the co-supervisor Michael Haslgrübler, who encouraged me throughout the entire way, starting with some very vague ideas, leading to a robust prototype, including his continuous enthusiasm that kept me *pursuing* the right direction.

Thanks to the volunteer participants, sacrificing about two hours of their time to the evaluation of Natural Pursuit Calibration, with coffee and ice cream as the only remuneration and to Nina for proofreading and any linguistic revisions.

Last but by no means least, also to everyone that had to bear with a restless version of mine, to my mum and dad who always have a loving shoulder and to Matthias and Magdalena who always have a laugh to share. To Catharina, Doris and Susanne, who brighten up the darkest mood with a little chitchat and finally to Florian, who went down the same road and still had optimism and confidence left to share whenever I was about to lose ground.

Contents

1	Introduction	11
2	The Eye as Input Modality	13
2.1	Physiological Characteristics	13
2.2	Interaction Concepts	14
2.3	Calibration	16
3	Concept	19
4	System Design	23
4.1	Infrastructure	23
4.2	Software Design	24
4.2.1	Motion Path Extraction Model	25
4.2.2	Motion Path Matcher Model	29
4.2.3	Pupil Calibration Plugin	31
5	Evaluation	33
5.1	Research Questions	33
5.2	Preliminary Parameter Evaluation	33
5.2.1	Setup	34
5.2.2	Participants	35
5.2.3	Procedure	35
5.2.4	Results	35
5.2.5	Discussion	43
5.3	System Evaluation	45
5.3.1	Setup	45
5.3.2	Participants	51
5.3.3	Procedure	51
5.3.4	Results	53
5.3.5	Discussion	60
6	Future Work and Conclusion	63
	Bibliography	73

1 Introduction

The great potential of the eye as input modality has been studied since more than 30 years, and with the emergence of distributed Internet of Things (IoT) environments, the interest in gaze-based interaction increases. Due to its characteristics the eye promises to act as a universal modality enabling fast, effortless and hands-free interaction, taking human-computer interaction (HCI) to a next level. Other than prevailing input modalities, the eye indicates one's intention implicitly, as humans instinctively look at things of interest [JS16], revealing information that cannot be accessed by dominant interaction techniques, like traditional remote controls.

Nevertheless, a great challenge is to differentiate between intended and unintended eye movement, as the eye is used for perception and control simultaneously. This is referred to as the Midas touch problem [Jac91], and different concepts exist to overcome this issue, such as the use of additional input modalities, e.g. speech or traditional buttons. An approach in contrast to this, is the introduction of unnatural eye movements, which the user has to train and execute explicitly, to produce a signal clearly distinguishable from natural acting.

Furthermore, to exploit the entire potential, offered by the concept of gaze-based interaction, an accurate gaze estimation is necessary. This makes successful calibration essential for eye tracking, which is frequently described as a time-consuming task of insufficient usability, that is inconvenient and tedious [CJ11; KH16; MB14; OM04].

Commonly available concepts are N -point calibrations where the user has to dwell at certain predefined points distributed within the subject's field of view (FOV). Another approach, introduced by Pfeuffer et al. [Pfe+13] is based on the observation, that humans naturally dwell on moving objects [VBG13] and that the eye is only able to perform smooth movements while fixating such objects [JS16] (smooth pursuit movement): calibration is performed by following a moving target on a screen. However, this leads to promising results, where so far artificially generated trajectories are necessary.

Therefore, within the scope of this thesis Natural Pursuits are introduced, a concept to apply smooth pursuit calibration without generating specially designed moving targets, enabling eye tracker calibration without any digital user interface or artificial annotation of the environment. This is done by extracting any occurring motion trajectory out of the egocentric video stream and correlate the user's eye movement.

2 The Eye as Input Modality

As previously stated, gaze-based interaction offers a number of advantages, such as being fast, effortless and hands-free, especially for environments where a diversity of input devices require a universal modality. Nevertheless, different challenges arise, due to characteristic eye behavior, that will be illustrated in the following, along with various approaches to deal with them.

2.1 Physiological Characteristics

To begin with, it is important to understand how the eye actually moves. Basically, there are four different forms of movement the eye is capable of, that are considered in detail:

Saccades are defined as fast jumps, that can be voluntary if one aims to translate gaze on purpose, or driven by reflex, caused for example by sudden visual disturbances. The velocity of saccades typically ranges from 30 to 180 degrees per second and one jump takes between 20 to 140 ms [DD90]. Saccades are performed permanently when scanning the environment.

Fixation is the state, where the gaze stays stable for usually 200 to 600 ms in order to enable the brain's visual system to process the focused area [MB14]. During a fixation, the eye performs jittery motions, that can not be controlled and occur naturally.

Smooth Pursuit Movement is the eyes' ability, to perform continuous motion when guided by a moving visual stimulus. Thereby, the eye imitates the velocity and the direction of the traced moving object [Mul11].

Vestibular-ocular Reflex describes the eye's ability to perform compensation movement, in order to keep fixating a point while the head is moving [Mul11].

Others exist in addition, like vergence movements or the optokinetic reflex, which are outside the scope of this work but are explained in great detail by Mulvey [Mul11] or Dell'Osso and Daroff [DD90].

2.2 Interaction Concepts

The need for specific interaction design emerges, when talking about gaze-based user interfaces, as eye movement is vastly unconscious and instinctive. Further, targeting cannot be performed highly accurate, due to fixation jitter and technological constraints, and prevention of unintended interaction is crucial. Therefore, different approaches exist, which aim to support gaze-based interaction in both, digital and distributed ambient scenarios.

Eye as a Pointer

In a first concept, the eye is used partially for explicit interaction, in order to eliminate the Midas touch problem, but still benefit from the eye movements' speed. Thus, the gaze is employed as a pointer, whereas selection is performed by an additional input modality. Zhai et al. introduced the Manual And Gaze Input Cascaded (MAGIC) pointing approach where mouse cursor movement is enhanced using gaze pointing [ZMI99]. MAGIC has been further developed by replacing the mouse with a touchscreen, a touch mouse and a multi-touch surface [DS09; Pfe+14; SD12]. Additionally, head and hand gestures have been successfully examined to perform action selection [HRM13; ŠM12].

However, gaze pointing takes advantage of the eye's speed, it still requires an additional input modality, raising the need for modality synchronization [JS16]: Assuming that a user is well trained on a system, the gaze tends to move faster than the user is able to confirm the selection, as the procedure is already internalized.

Dwell-based Selection

Dwell-based selection, as a second approach, introduces deliberately extended fixations for gaze-based interaction in order to overcome unintended selections. This means that events can be explicitly triggered by focusing on an interaction object for a predefined duration (dwell time), longer than a natural occurring fixation. The appropriate dwell time depends on the application context and the user's experience.

Dwell-based selection has been applied successfully for interaction with digital user interfaces, like Majaranta et al. investigated dwell-based on-screen typewriting [MAŠ09; MR02; Maj+06], Dybdal et al. performed dwell time selection on smart screens [DAH12] and De Luca et al. developed a security system for entering a pin using dwell time [DLWD07]. In addition, Corno et al. [Cor+10] and Jungwirth et al. [Jun+17] utilized dwell-based selection for environmental control in an off-screen context.

Though the risk of unintended selections is reduced, dwell-time selection is not applicable in certain situations, e.g. if the interaction object is frequently looked at, like a TV. Moreover, it is only capable of binary interaction support.

Gaze Gestures

Contrarily to the two approaches introduced so far, gaze gestures rely on relative eye movements, that have to be trained and executed explicitly by the user. There are different concepts to design gaze gestures, but the basic idea is the same: the user intentionally performs some task with the eye, that is very unlikely to happen by chance from natural gazing behavior. In the following, differentiation will be done based on the type of eye movement the gesture relies on.

Gestures based on Saccades

Gestures based on saccades are frequently described as a sequence of eye movement patterns [DS07; Mul11], typically strokes, that can be either performed by transitions between predefined areas or by drawing shapes with the gaze without reference focus points [MB14].

The least complex, so-called one-stroke gestures have been implemented to interact with digital user interfaces by Juang et al. using off-screen areas [Jua+05], in order to enable web browsing, and Møllenbach et al. developed an on-screen interaction concept where a gesture is defined as a transition from one area to an opposite one [Møl+10], which has also been applied to smartphones [DAH12]. However, the proposed one-stroke gestures are robust to unintended executions only up to a certain degree, therefore various approaches exist, that use an extension of two-legged back and forth gestures [Han+16; Kan+14a; Kan+14b]. Moreover, the set of available gestures, and therefore triggered events, can be extended by striking a sequence of areas [Iso00; Ist+10].

Furthermore, multiple gestures, can also be supported by another approach, where shapes are relatively drawn with the eye, independent of any predefined interaction area. This is demonstrated by Drewes and Schmidt [DS07] and by Bulling et al. [BRT09] who applied a set of eight directional line components, cf. Figure 2.1a, that can be combined to gestures, illustrated in Figure 2.1b. A similar gesture design is defined by Wobbrock et al. who enabled typewriting by letter-like gaze gestures [Wob+07].

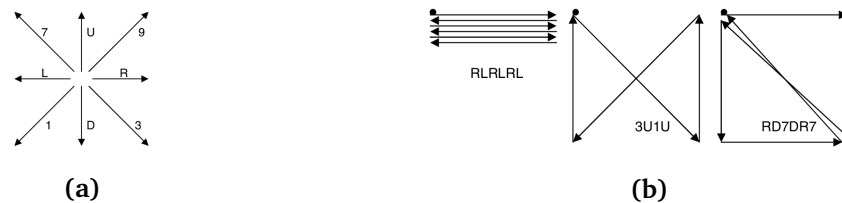


Figure 2.1: Eight possible stroke components in the right, and three sample gestures in the left image [DS07].

Complex gaze gestures are robust to unintended interaction and support a large set of possibilities. Nevertheless, it is demanding to the users to (i) train the available gestures and (ii) remember them and their corresponding triggered event.

Gestures based on Smooth Pursuit Movement

As a second category, gaze gestures based on smooth pursuit movement, attempt to overcome the shortcomings of saccade based gestures, by making use of natural eye movement characteristics. Moving targets are shown to the user, and following them enables interaction.

This was first explored by Vidal et al. who introduced Pursuits [VBG13], a system that detects whether a user is attending moving objects on a screen or not, demonstrated in Figure 2.2. Pursuits were picked up by Esteves et al. who proposed Orbits, which are interaction targets, moving in a circular manner on a smartwatch, and investigated design considerations for multi target behavior [Est+15]. Furthermore, Cymek et al. [Cym+14] and Liu et al. [Liu+15] implemented authentication interfaces upon smooth pursuit eye movements. The approaches mentioned so far, require specially designed dynamic interfaces, hence, Schenk et al. came up with a concept for digital interaction, based on multiple eye characteristics, that requires only minor extension of existing desktop environments: After dwell-based attention detection on an object, moving targets appear, which can then be followed for action selection [Sch+16].

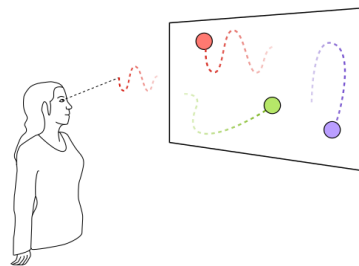


Figure 2.2: The concept of Pursuits: a person tracing motion trajectories with the eye in order to interact [VBG13].

Velloso et al. and Delamare et al. went a step further and employed smooth pursuit based interaction with ambient devices. Therefore, Velloso et al. augmented virtual as well as mechanical devices either internally (digital content, direct object manipulation) or externally (projection controls, laser pointer) in order to interact [Vel+16]. Whereas a further approach relies on augmented reality, using a head-mounted display, in order to build a gesture guiding system, supporting smooth pursuit interaction [DHI17].

2.3 Calibration

Calibration is a critical and challenging task, that has to be considered in a majority of gaze-based interaction techniques, like the eye as a pointer, dwell-based approaches, area-guided gaze gestures or for any reliable gaze analysis, influencing user acceptance and interaction performance. During the calibration process, a model is created, that maps eye features (a model of the eye, extracted out of eye video stream, including e.g. pupil center

position or corneal reflection) to known reference points within the user's field of view, in order to gain a correct gaze estimation. Therefore, a reliable pupil detection is required, which is especially demanding when talking about eye tracking in the wild, as numerous challenges, like changing lighting conditions, have to be accounted [Eva+12; Fuh+16].

Sampling Points

The prevailing way to identify reference points is simply to ask the user to dwell on predefined ones, which requires active user cooperation. Such predefined calibration targets are shown to the user successively, distributed within the FOV, typically arranged in a grid-like manner. Fixation on points is assumed, and eye and reference data get sampled, either automatically after a fading in threshold or after an explicit command.

However, quite a large number (5 to 20 [OMY02]) of reference points is needed for common N -point approaches and calibration based on dwelling is tedious to the user and exhausting to the eyes [Pfe+13]. Therefore, various attempts have been made to reduce the number of calibration points, such as introducing complex mathematical models in order to reduce the number of required points to two [OM04; OMY02; VCP04]. In addition, there are also proposals that include extended hardware equipment, like additional cameras or light sources, in order to calibrate successfully using only one reference point [GE08; VC08]. Nevertheless, Guestrin and Eizenman [GE08] as well as Villanueva and Cabeza [VC08] stated, that a higher number of calibration points would increase the robustness of the process.

To enhance dwell-based approaches Renner et al. proposed a motion guided point-to-point calibration technique in a virtual environment, where a dragonfly flew from point to point, attracting the user's attention [Ren+11], cf. Figure 2.3.



Figure 2.3: A grid-like calibration pattern on the left, versus the motion trajectory of the dragonfly illustrated on the right [Ren+11].

Contrarily to prior approaches, Kasprowski and Harezlak implemented a more implicit form of point-guided calibration, by predicting the most probable fixated object out of a set of shown ones, in order to gain reference points. A gaming scenario was used for evaluation, where two different types of objects were shown and users were asked to click on one of them [KH16].

Sampling Motion

In order to make calibration less tedious and obtrusive, techniques supporting natural eye movements were aimed for. Therefore, besides dwell-based N -point methods, calibration based on moving targets has been investigated.

Early attempts to perform calibration using smooth pursuit movement were done by Kang and Malpeli, who presented moving targets in the horizontal and vertical axis in order to gain accurate gaze estimation for cats [KM03]. Similarly, Kondou and Ebisawa implemented a system where humans were presented moving targets on a stationary eye tracker and were asked to follow them, in order to calibrate. Although they achieved promising results, the system was not capable of detecting whether the user is really attending the motion trajectory or not [KE08].

Hence, the introduction of Pursuits [VBG13], which not only utilizes smooth pursuit movement but is also capable of attention detection, offered a great opportunity for eye tracker calibration. This is demonstrated by Pfeuffer et al., who introduced Pursuit Calibration, where a user is located in front of a public display, following different moving targets, such as a moving dot, shooting stars or floating words [Pfe+13], as illustrated in Figure 2.4. Further, Khamis et al. applied a similar concept but used revealing text in order to calibrate the user while reading [Kha+16].

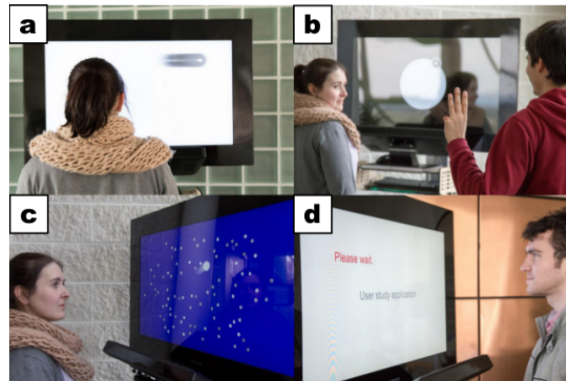


Figure 2.4: Different application scenarios for Pursuit Calibration: a moving dot (a,b), shooting stars (c) and floating words (d) [Pfe+13].

A different approach, named CalibMe and suggested by Santini et al., uses the Vestibular-ocular Reflex to provide another form of eye tracker calibration: A person is fixating on a marker, while moving the head in order to sample a wide range of gaze positions [SFK17]. Contrarily to state of the art concepts based on smooth pursuits, CalibMe does not enable attention detection and is therefore not considered any further.

3 Concept

However, usage of smooth pursuits leads to promising concepts for both, gaze-based interaction as well as calibration, so far special designs of dynamic user interfaces are necessary. As described in section 2.2, this can be done, either in form of digitally designed moving objects [Est+15; Pfe+13; VBG13] or of artificial annotation of the environment [Vel+16].

Leveraging the results of former research, we propose a concept, named Natural Pursuit Calibration (NPC). It applies smooth pursuit movements for mobile eye tracker calibration, without the need of designing artificial moving targets and any annotation of surroundings. Whenever a person wearing an eye tracker is attending an arbitrary moving object with the gaze, the systems detects it and uses the corresponding eye and motion data for calibration. A possible scenario is demonstrated in Figure 3.1, where a person is watching a car driving by while standing in front of a window, viewing the scenery.

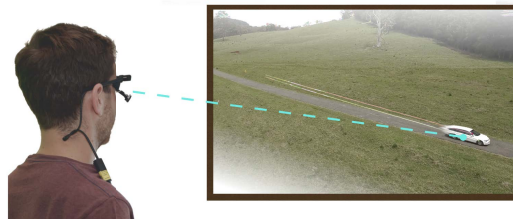


Figure 3.1: Person wearing an eye tracker is watching a moving car, which triggers an update of the calibration model.

The concept of NPC, as illustrated in Figure 3.2, is to extract motion trajectories out of the egocentric video stream, provided by an eye tracking platform, that represents the user's FOV, using optical flow [HS81] on the one hand, and visual marker tracking on the other hand. The extracted motion path is then correlated to the user's eye movement. If the correlation threshold for both eyes is high enough, the single points a motion trajectory consists of, are classified as a match, which is then used for updating the calibration model.

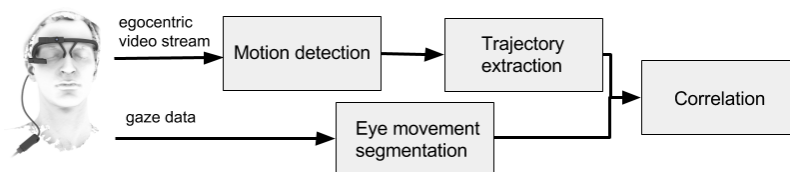


Figure 3.2: Schematic overview roughly representing the system design.

Beneficial Effects on Calibration

A continuous calibration process is proposed: Once a calibration model is created, every supplementary match is used to update the model only in the surroundings of it. Therefore the correspondent eye and path data around the affected area are exchanged and an updated calibration model is built and applied. An example is shown in Figure 3.3, where a match is recognized in the left upper corner of the FOV (blue dots), only the data in the upper left part is exchanged (gray dots). The size of the affected area depends on the range of the match which is influenced by various factors, like the speed of the motion path or the length of one extraction period.

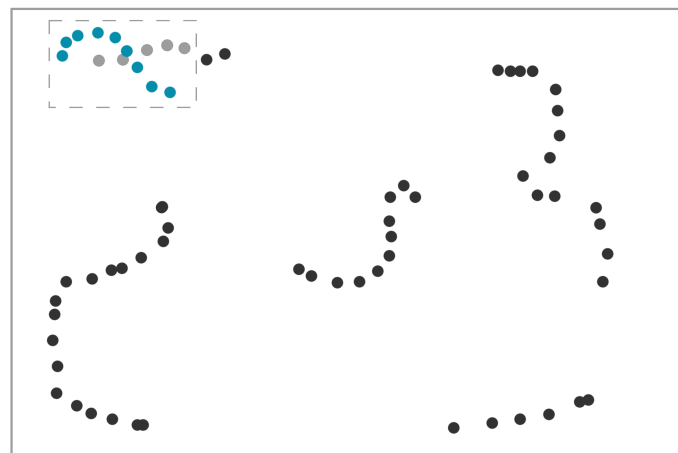


Figure 3.3: Conceptual illustration of the continuous calibration model, where the dark gray dots represent the original model data. The light gray dots demonstrate outdated data, that is replaced by a new match (blue dots).

Furthermore, NPC can be seen as an implicit, as well as an explicit technique, depending on the setting. An implicit example, as stated before, could be following a car that drives by, whereas for explicit calibration the user is able to generate motion on his own. This could be to move one's finger in one's field of view and to follow the movement with the eye until a sufficient calibration is reached. Implicit and explicit calibration can be combined to reach and maintain an optimal calibration result. Firstly, explicit calibration is executed to reach reasonable accuracy, and afterward, implicit calibration can be continued, where the user is not required to pay attention to it anymore. If any moving object is traced, calibration gets updated continuously.

This leads to another great advantage, as calibration is only executed if the user is attending to a movement in their field of view, the calibration process can be executed in the background, simultaneously to any primary task.

Beyond Calibration: Natural Pursuit Interaction

Moreover, the underlying concept of Natural Pursuits can be applied for gaze-based interaction, that is insensitive to calibration. As this is outside the scope of this thesis, only some application scenarios will be sketched, to demonstrate the capability.

As well as for calibration the concept of Natural Pursuit Interaction (NPI) can be implicit or explicit. A possible application scenario for implicit NPI could be an assistance system in visual inspection task at an assembly line in an industrial context: If the gaze remains on an examined object for longer than average, the velocity of the assembly line gets adapted, as the unusual duration of the fixation is an indicator for any irregularity.

Contrarily, NPI could also act as an explicit and universal interaction technique, where the user is able to draw any gesture with their eyes, and their own finger serves as guidance, cf. Figure 3.4. This would result in a gesture guiding system, that supports natural eye movement, similar to the system proposed by Delamare et al. [DHI17]. The support of gestures by smooth pursuits reduces the demand to the user, as a major drawback of saccade-based gaze gestures is eliminated: learning and training of unnatural eye movement is no longer required.

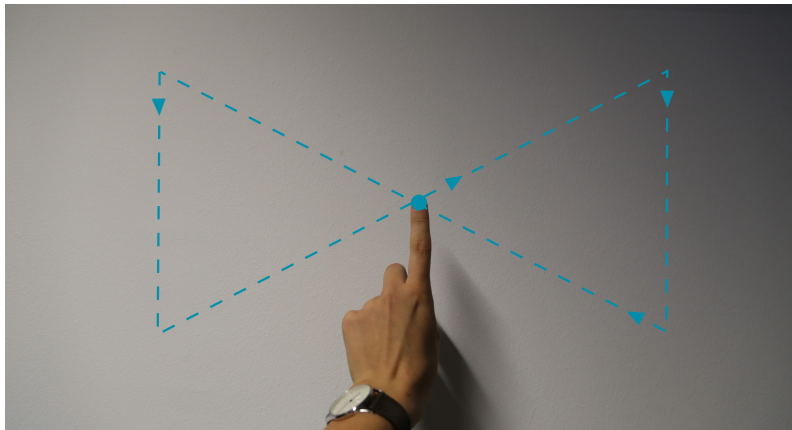


Figure 3.4: Explicit interaction scenario, where the movement of the finger acts as gesture guidance.

4 System Design

To empower and prove the concept of the proposed system, a prototype has been developed and evaluated. In the following chapter a detailed description of the hardware setup and of the software architecture is provided.

4.1 Infrastructure

A binocular Pupil Labs eye tracker, shown in Figure 4.1, which is equipped with two infrared cameras capturing each eye and a wide-range world camera, recording the egocentric view, was used. Due to performance issues, the sampling rate of the egocentric camera image was only 10 Hz and those of the eyes was 60 Hz. The eye tracking device has a diagonal resolution of 100° or 1480 px (for $1280 \text{ px} \times 720 \text{ px}$ world camera resolution).



Figure 4.1: A binocular Pupil Labs eye tracker, with two infrared cameras, which are capturing each eye, and a wide-range world camera, that is recording the egocentric view.

The implemented prototype is a standalone Java application, which communicates with the Pupil Labs framework, that is open source and implemented in Python, via messages. According to Haslgrübler et al. [Has+17], the framework uses Protocol Buffers [Var08], a language and platform neutral serialization library for defining and serializing messages. Definitions of data structures are written in a single format, which is then compiled to libraries for Java and Python. Furthermore, ZeroMQ, an asynchronous high-performance messaging protocol [Cor14] is utilized for transferring them. In Figure 4.2 it is shown, that the Pupil Labs framework was extended by a publisher plugin that acts as a ZeroMQ client and forwards normalized pupil coordinates of each eye together with the corresponding timestamp and the image stream of the egocentric camera to the proposed prototype, which again implements a messaging client, via a ZeroMQ router.

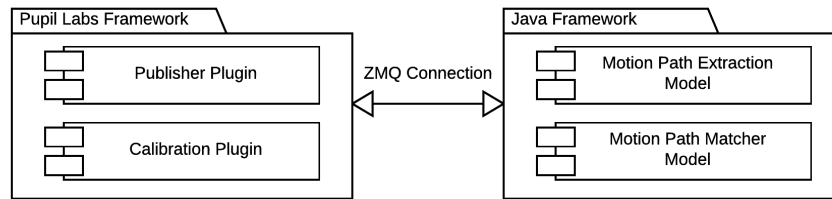


Figure 4.2: Communication between Pupil Labs framework and the implemented system via ZeroMQ.

4.2 Software Design

As illustrated in Figure 4.3, the system consists of three major parts, described in detail in the following sections:

The Motion Path Extraction Model , which is subscribed to the egocentric video stream and aims to extract relevant motion paths out of the provided images, while eliminating noise.

The Motion Path Matcher Model , that decides if a calibration process can be triggered, by comparing relevant motion paths against the eye movement.

A Calibration Plugin that takes reference points out of the egocentric images and corresponding pupil coordinates in order to update the calibration model.

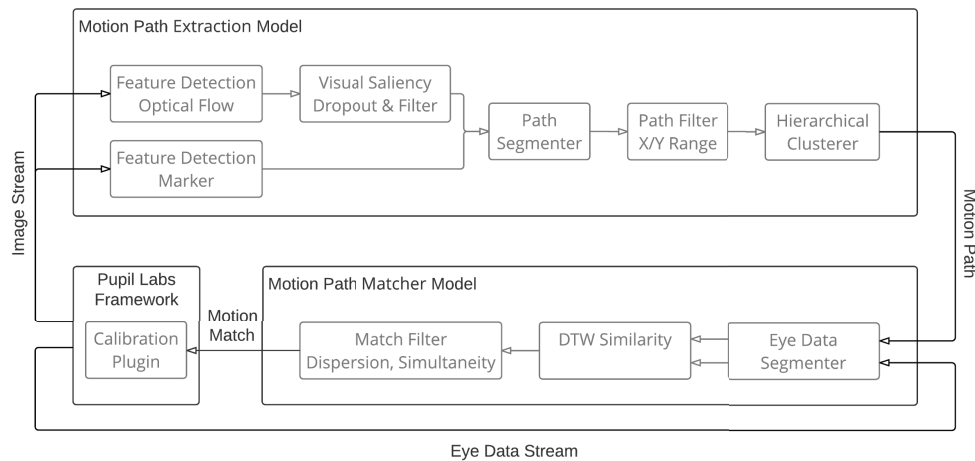


Figure 4.3: Detailed overview of the system architecture.

4.2.1 Motion Path Extraction Model

The highlighted box in Figure 4.4 represents the Motion Path Extraction Model and shows the sequential steps performed within this model.

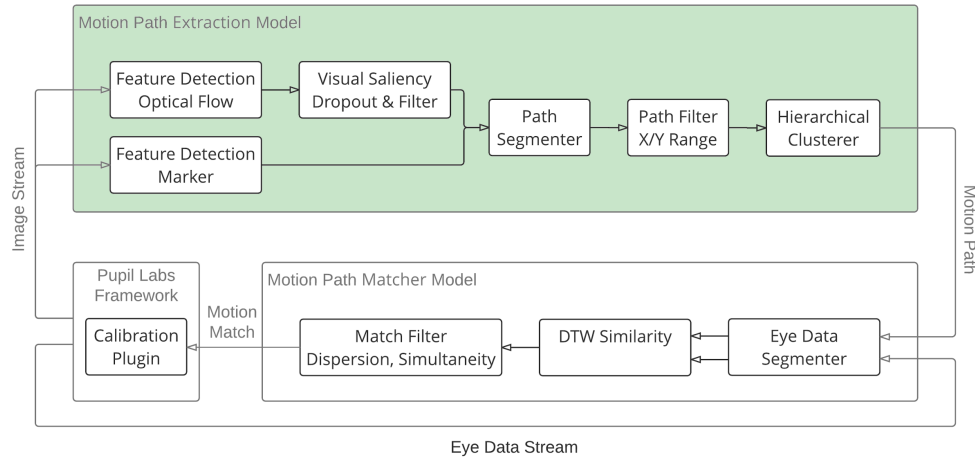


Figure 4.4: Features are identified, traced over a number of frames and motion noise is filtered, in order to extract valid motion trajectories.

In the first step feature tracking is used to extract motion paths out of the egocentric video stream. In order to identify and track points of interest, either an optical flow [HS81] based approach, which will be described in detail in a following subsection, or visual marker tracking is applied. Initially, the received image gets down-scaled by a factor of two and converted to a grayscale image.

For visual marker tracking, squared binary fiducials, cf. Figure 4.5, are utilized. Detection is done by a Fiducial Detector, provided by BoofCV computer vision framework [Abe16], which is applied to each frame. Therefore, in the first step, the image is converted into a binary image using the average value of the surrounding pixels, times the scale (0.95 is the default value, given by BoofCV), as a dynamic threshold for each pixel. Subsequently, internal and external contours are traced and extracted out of the binary blobs using the algorithm introduced by Chang et al. [CCL04]. These contours are then converted into four-sided polygons using a split-and-merge algorithm. The resulting quadrilaterals are then refined to subpixel accuracy, undistorted and matched against predefined fiducials. In a last step, the position of the detected visual markers is transformed into world camera's coordinates and the center of the detected fiducial is a potential motion path entry.

As a result of both techniques, fiducial marker and optical flow feature tracking, a number of different marker or features are detected for each frame. Each marker or feature is identified by a unique ID, x/y coordinates and a corresponding timestamp. If IDs appear on a number of consecutive frames, or a window of frames, each ID defines one motion path - a time series of x and y coordinates. The appropriate length of the window is determined in section 5.2.4.



Figure 4.5: Examples of squared binary fiducials.

As one would suggest, motion path extraction is sensitive to motion noise, since the user can move freely and the camera is placed on the head-worn eye tracking device. Therefore, numerous motion paths are irrelevant for further analysis and two different methods are used to eliminate them:

Filters are applied to eliminate irrelevant paths. The first requirement is a minimum number of entries, only paths containing 5 or more are further considered. Additionally, a certain range (10 %) in either x- or y-direction must be covered, otherwise, the path is rejected. Supplementary, thresholding the gradient is implemented as another type of filter.

Hierarchical clustering is used to separate noise from relevant paths. A feature vector is calculated for each path, which is then used for clustering into noise and non-noise. This is described in more detail and validated in one of the following subsections.

The outcome of the Motion Path Extraction Model is a number of paths, considered as relevant in the first step. All non-noise motion paths are forwarded to the Motion Path Matcher Model.

Optical Flow and Visual Saliency Dropout

In addition to square fiducial tracking, the system is capable of extracting any occurring motion trajectory out of the egocentric video stream, based on the optical flow. Horn and Schunk [HS81] defined the optical flow as follows:

"Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image."

In other words, optical flow describes the displacement of objects within a scene relative to the viewers perspective. If all objects would be static within this scene, the resulting displacement caused by the movement of the viewer is similar for every set of pixels. Therefore, discontinuities within the optical flow are an indicator for moving objects [HS81].

In order to detect the displacement of features within a sequence of frames in the proposed setup, a Pyramid Kanade-Lucas-Tomasi (KLT) feature tracker [LK+81; Shi+94; TK91], provided by BoofCV [Abe16], is utilized. The Pyramid KLT tracker implementation is capable of tracking a feature over a larger distance, than the basic KLT tracker.

KLT features are defined as a square window of pixels, where the size equals two times the defined radius (3 in this implementation) plus one, resulting in 7x7 pixels. Such a window is only an appropriate candidate for tracking if it encodes sufficient information, for KLT features this is defined as true if both eigenvalues are above a certain threshold. The lower boundary of the threshold is defined by the eigenvalues of a region of almost uniform brightness, whereas the eigenvalues of a region containing high information, like a corner, serve as an upper boundary [Shi+94].

In Figure 4.6 an example is given, where two consecutive frames are fed into the Pyramid Kanade-Lucas-Tomasi feature tracker. The features are marked with a circle and the resulting displacement of KLT features is shown in form of a 2D vector, with a different color assigned for every unique feature.

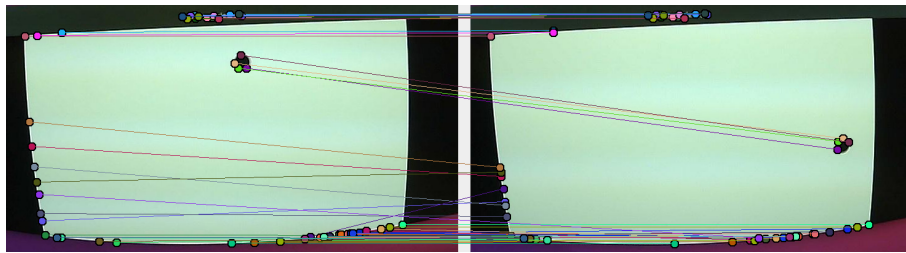


Figure 4.6: Two different frames where corresponding features are graphically associated, using the Pyramid Kanade-Lucas-Tomasi feature tracker.

For tracker initialization, a predefined number N of KLT features, the N with the highest eigenvalues, are marked as active and registered for tracking. The track of certain features can be lost over a number of frames, e.g. if they get occluded or if they move outside the world camera's FOV.

Additionally, a visual saliency dropout and filter have been implemented. The visual saliency specifies how a region stands out compared to its neighborhood, stated as a number from 0 to 256. Therefore, it is an indicator how likely it is that the region is of interest to humans. The proposed system uses an implementation introduced by Achanta et al. [Ach+08], who use the local contrast of a region compared to the surrounding at different scales as low-level saliency. For feature dropout, the maximal visual saliency for the KLT feature region is calculated, and if this is beyond the threshold of 50, the feature is marked as inactive and tracking is stopped. In Figure 4.7 an example is illustrated, where the region of interest is on a display. However, a lot of features are found in the surrounding environment, cf. Figure 4.7a, where the visual saliency is lower. In Figure 4.7b the same scene is represented but after the application of visual saliency dropout.

To ensure the number of features remains constant, although visual saliency dropout is applied and features can get inactive due to various reasons, new ones are added in every frame, if necessary.

Furthermore, a saliency filter is implemented, where a motion path entry is only considered if the saliency value is above 150. This means that only motion paths in salient regions are processed further.

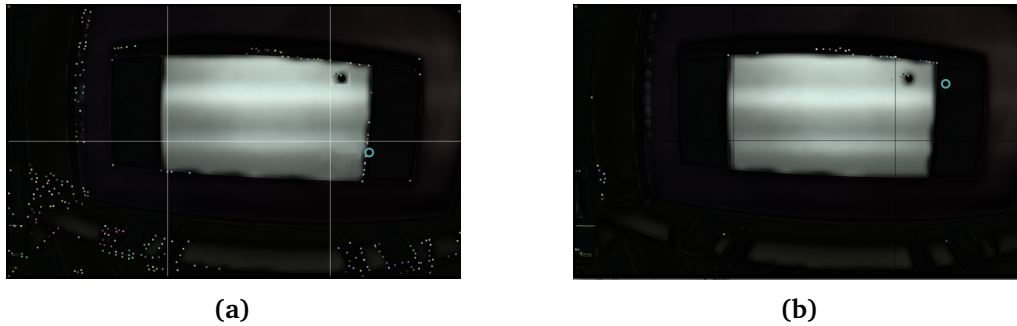


Figure 4.7: A lot of features in non-salient regions on the left image, versus a very reduced number of features on the right, to demonstrate visual saliency dropout.

Hierarchical Clustering

As already stated before, motion path extraction is sensitive to motion noise, due to the mobile eye tracker, worn on the head. However, it has been already discussed, that the output of the optical flow indicates moving objects, by paths that differ from all the others. Therefore, the non-relevant paths caused by motion noise are all very similar to each other, and clustering is a well-suited method for noise elimination.

To perform clustering, a feature vector is calculated for each path, including range in x/y direction, speed, average gradient, maximal gradient, dispersion, variance of x/y coordinates and the variance of magnitude. The vector is then fed into WEKA's [Wit+16] hierarchical clustering algorithm, where the number of clusters is two (noise/non-noise).

In order to validate hierarchical clustering for this specific problem, an experiment has been conducted: A sample application was written where four static visual markers were placed in the edges and one was sliding from left to right on an assistance line, like illustrated in Figure 4.8a. A session was recorded where a user, wearing the Pupil Labs eye tracker was asked to follow the marker with the eye. While doing so, moving the head was not prohibited. In Figure 4.8b the corresponding extracted motion trajectories and the movement pattern of the eyes are shown in different colors. The red and blue dots depict the recorded movement of the right and left eye, whereas the gray and pink dots represent extracted motion paths of the five markers. Gray motion paths are assigned to the noise cluster, while pink ones belong to the non-noise cluster.

Hierarchical clustering output was evaluated for each extracted motion path of the recorded session. Metrics of the confusion matrix could be extracted, as the IDs of the square fiducial were known, as well as the ground truth: Only the moving marker in the middle should be classified as non-noise, but this should be always the case, as the user was watching it during the whole session. Figure 4.9 shows the output of the experiment, in terms of False Positives (FP) and False Negatives (FN), True Positives (TP) and True Negatives (TN). False Negatives (FN) is zero, which means that no relevant path was rejected in the experiment, while TN is quite high, demonstrating that a lot of noise is filtered. Still, there are numerous

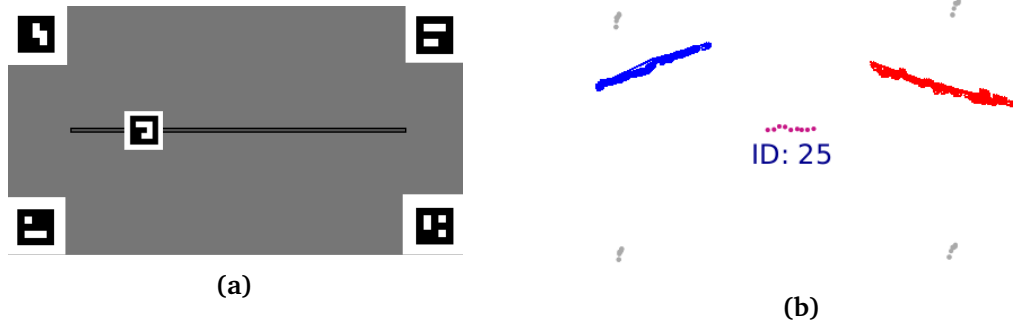


Figure 4.8: The interface used for validation of clustering algorithm on the left and the corresponding extracted motion trajectories on the right.

irrelevant paths caused by motion noise clustered as non-noise, but those are eliminated during the matching process.

Predicted	Non-Noise	FP: 423	TP: 344
	Noise	TN: 1625	FN: 0
		False	True

Figure 4.9: A lot of noise is filtered out, while no non-noise trajectory is eliminated.

4.2.2 Motion Path Matcher Model

Once the motion trajectories are extracted out of the egocentric video stream, and a number of irrelevant paths are rejected, the aim of the Motion Path Matcher Model is to detect whether the user was following one of the motion paths with the eyes. This process has to be very robust, as the resulting matches are used for updating the calibration model.

In the first step, after relevant motion trajectories are received from the Motion Path Extraction Model, the cached eye data stream, containing normalized pupil coordinates for left and right eye and corresponding timing information, gets segmented appropriately. The region of interest within the eye data stream starts with the timestamp of the first motion path entry and ends with the last one. As a result, there are two time series, one for each eye, that can be correlated with the corresponding motion path, which can also be represented as a time series of x/y coordinates.

As the sampling rate of the eye data and the image stream differs, dynamic time warping is used to correlate eye movement and motion trajectories, which will be explained in detail in the following section. Both, the similarity of left and right eye against the motion trajectory, have to be above a predefined threshold to be classified as a match.

4 System Design

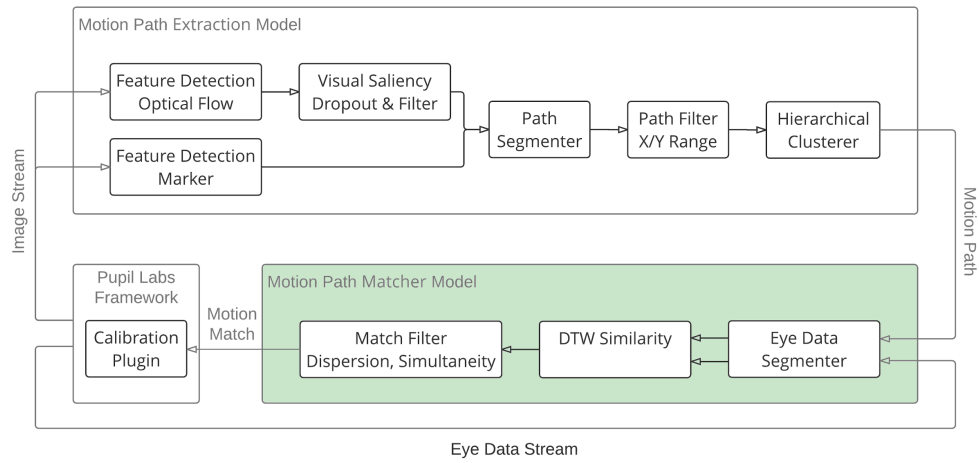


Figure 4.10: Motion paths and corresponding eye movements are compared, in order to detect motion matches.

To increase the robustness of the calibration process, all detected matches are cached. The aim of this cache is to eliminate falsely classified matches, that are unlikely to occur, due to eye movement characteristics. Therefore, concurrent matches are eliminated partly, as the major assumption is that the eye can follow only one path at a time. Nevertheless, information can be lost due to this rule, as two very closely located features can both be followed simultaneously, but their first detection time may differ. Therefore, both matches contain information not covered by the other one. To overcome this data loss partly, two matches are defined as concurrent if there is less than 40%, of the defined path window size, time gap between the two starting points. For an additional sanity check and to validate matches that are partly overlapping, the dispersion of them is calculated. All of the concurrent matches are rejected if the dispersion is above 10% of the image resolution size. Further, two more requirements are implemented, namely, restriction for a maximal number of matches and neglecting of single detected matches. The length of the match cache is chosen to be 5 seconds. This seems to be enormously large, but the longer the cache is, the more robust the system performs. Moreover, the calibration process is not required to be capable of strict real-time feedback.

As the last step the motion match is converted into a calibration message, which basically consists of motion path entries with corresponding timestamps. These entries, referred to as reference points within the calibration process, are forwarded to the Pupil Labs framework.

Time Sequence Matching

Contrarily to related work [Est+15; Pfe+13; VBG13], dynamic time warping (DTW) is used for calculating the similarity of time series, rather than the Pearson correlation coefficient. The captured motion is independent of the system and, therefore, it is not possible to

represent the movement as an explicit function, that can be sampled at any arbitrary sampling rate. Due to the risk of rejecting valuable information when sub-sampling the eye data, and due to the broad range of applications, where DTW was applied successfully for time sequence matching, the system rather goes for this solution.

The principal of DTW is to detect the similarity of two time series, even if their phase is shifted slightly. This is done by aligning them and not only by re-sampling one of the signal. Already in the 1970s Sakoe and Chiba used a time sequence matching approach by dynamic time warping for spoken word recognition [SC78]. Ever since DTW has been applied in various domains, such as speech [SC78; SM90] or voice recognition [MBE10], correlating hand-written sequences [RM03], or gesture matching [Cor01].

FastDTW, introduced by Salvador and Chan [SC07] and implemented in the Java ML library [APS09], is improving performance from quadratic to linear time and space complexity. Therefore, it is utilized for the proposed prototype to perform time alignment of the eye movement and the motion path. Furthermore, the time series are normalized before correlation, according to equation 4.1.

$$\begin{aligned} x' &= \frac{x - x_{min}}{x_{max} - x_{min}} \\ y' &= \frac{y - y_{min}}{y_{max} - y_{min}} \end{aligned} \quad (4.1)$$

If the similarity between either eyes and the motion path is above a predefined matching distance, whose influence is evaluated in section 5.2, the path window is classified as a match and further proceeded to the match cache, which has been described previously.

4.2.3 Pupil Calibration Plugin

To complete the work chain of Natural Pursuit Calibration, a plugin, highlighted in Figure 4.11, within the Pupil Labs framework is required, in order to update the calibration model. Therefore, detected reference points, which are received as a calibration message from the Motion Path Matcher Model, are associated with the corresponding eye data. Due to the complexity of the 3D eye model data, which Pupil Labs utilizes, these are cached within the Calibration Plugin for a sufficiently long time span. The cached eye data are allied to the reference points independently to the matching process, which is performed only with normalized pupil coordinates.

Primary, the normalized coordinates of the reference points are upscaled to the actual screen size, and the former calibration model is loaded, if already existent. The model is defined by a set of 3D eye model data for both eyes, which are associated to reference points on the screen and enriched with timing information.

In case that there is no calibration model available yet, the currently received path reference points are associated to the corresponding eye model data. This is done by using the timing

4 System Design

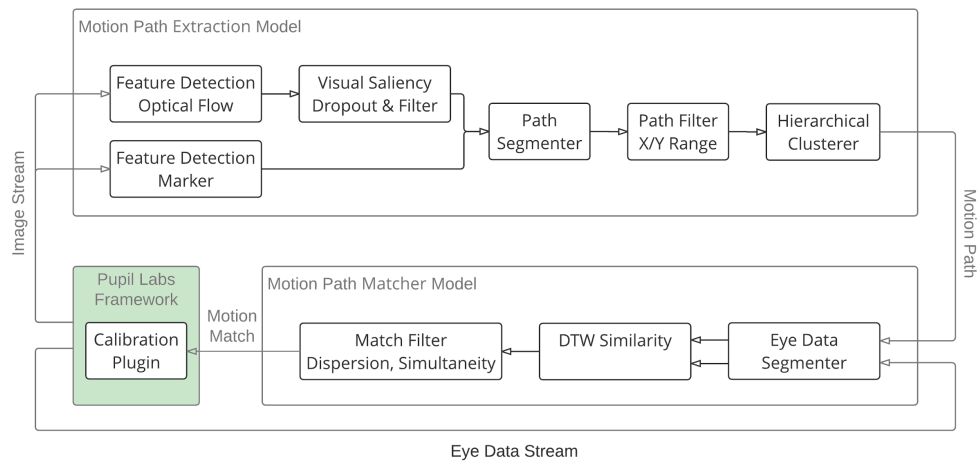


Figure 4.11: A Python plugin, implemented within the Pupil Labs framework, takes reference points via an API and triggers the calibration process.

information of the reference points' root image frame, recorded by the world camera, which has been passed through the whole process. In the end, the internal finalization of the Pupil Labs calibration gets triggered.

Otherwise, if calibration has already been performed at least once, the existing calibration model is partly updated but not completely discarded. Therefore, reference points within the neighborhood, defined with a 20% range in x- and y-direction, of newly received ones are identified as obsolete. The combination of non-obsolete old reference points and new ones, together with the corresponding eye information, is then used to complete the process, enabling an on-going refinement and update of the calibration model.

5 Evaluation

In order to prove the concept, to relate NPC to common calibration techniques and to demonstrate the value of the proposed system a two-stage evaluation process has been conducted:

Preliminary Parameter Evaluation has been executed at first, to investigate the impact of different parameters and restrictions and to optimize a final parameter setting for NPC using optical flow feature tracking.

System Evaluation has been carried out to compare the proposed calibration technique (using fiducial as well as optical flow feature tracking) to prevailing ones offered by Pupil Labs, namely Natural Feature Calibration and Screen Marker Calibration, regarding user acceptance and accuracy.

The aim of the evaluation methods is to answer the research questions, given in the following section.

5.1 Research Questions

Question 1 (Q1): Do gained results of Pursuit Calibration [Pfe+13] apply to Natural Pursuit Calibration, in terms of attention detection, target velocity, path and sample size?

Question 2 (Q2): Is Natural Pursuits Calibration improving subjective usability, in terms of acceptance, user demand and time required, without a significant loss of accuracy, compared to Natural Feature Calibration?

5.2 Preliminary Parameter Evaluation

The first setup is employed to investigate attention detection, as well as target velocity of NPC using optical flow and compare the outcome to Pursuit Calibration [Pfe+13], in order to provide a partial answer to Q1.

Additionally, the aim of preliminary parameter evaluation is to investigate a set of different parameters and regularizations, having an impact on the system performance and if, how their values can be attuned to gain reliability. The strict requirement to the prototype is to

eliminate FP, in order to reach an accurate calibration, therefore optimization of parameter settings is crucial.

After identifying six parameter settings, from loose to very strict ones, the influence of the DTW matching distance, the speed of motion trajectories and the motion path window size to the system detection rate is examined in detail.

5.2.1 Setup

In order to reach comparability, the setup of Pursuit Calibration (PC) [Pfe+13] has been imitated: participants were located in front of a 23' screen with a resolution of 1920×1080 px at a distance of 65 cm, which is illustrated in Figure 5.1.

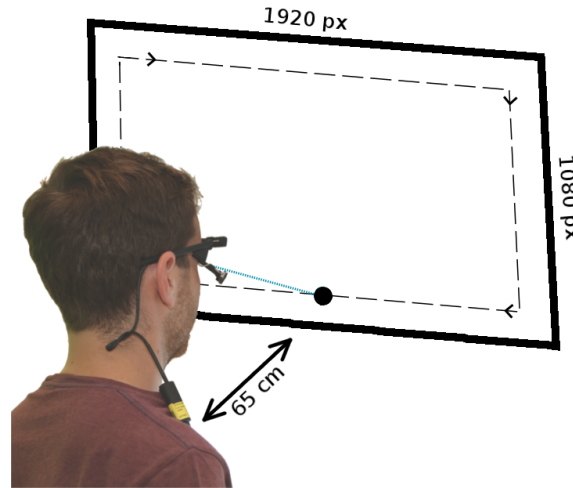


Figure 5.1: In order to discover appropriate parameter settings subjects are located in front of a display, following a black moving dot.

A black moving circle, with 50 px diameter, was animated choosing a subset of different velocities used for PC evaluation, reported in Table 5.1. The circle was clockwise traveling along the border of the screen, starting in the upper left corner, which results in a 10 second lap for the fast case (522 px/s), 20 second lap for the medium (261 px/s) and 30 second lap for the slowest movement (175 px/s). For the sake of completeness, velocities are listed in pixel as well as in degrees per second. In the following sections velocities will only be stated in px/s.

Table 5.1: Three different velocities chosen for parameter evaluation.

	<i>Pixel per second (px/s)</i>	<i>Degrees per second (°/s)</i>	<i>Seconds</i>
Fast	522	11.6	10
Medium	261	5.8	20
Slow	175	3.9	30

5.2.2 Participants

The procedure for preliminary parameter estimation was accomplished with three male participants, with different skill levels (expert, beginner, novice). None of them were wearing glasses permanently or suffering from any visual impairment.

5.2.3 Procedure

First, participants received an explanation of the eye tracker, if necessary. Then they were asked to put the device on and the infrared cameras of the eye tracker were positioned appropriately to cover the whole pupil in every possible eye position. Also including extreme cases like looking far right/left/up/down.

Users were asked to place themselves in front of the display, and to follow the black dot with the eyes, whenever it was visible. Once, the user stated to be ready, the dot began to move with the first velocity and after tracing a full round the next speed setting was displayed. After completion of at least one round for all three velocities, participants were asked to move and look around unguided, for about 30 seconds. This was done to gain a non-task sequence, in order to check for FP.

The whole study was recorded, to investigate parameter influence and optimization of fine.

5.2.4 Results

In the first experiments the following ten parameters and regularizations have been studied, by incomplete grid search:

Saliency Dropout deactivates feature points, on regions with low saliency.

Saliency Filter rejects motion paths entries, whose saliency is below a threshold.

Gradient Threshold is a lower boundary for the gradient of a motion path.

x/y Range Filter discards motion paths with smaller range in either x- or y-direction.

Dispersion Threshold neglects motion matches, that are distributed over the screen.

Single Match Suppression rejects single motion matches within one match cache, suggesting that one is likely to happen by chance.

Maximum Match Restriction is used as an upper boundary for the number of matches.

Image Scaling Factor is used to downscale the image.

Feature Points determines how many features are created initially, and retained over the tracking process.

Clustering can be applied to assign motion paths either to noise or non-noise cluster.

Application of the Gradient Threshold appeared out to have no effect at all, while the Single Match Suppression and the Maximum Match Restriction worsen the sensitivity of the system, without improving precision. Therefore, only the remaining six parameters and regularizations have been considered any further.

In Table 5.2 the six resulting parameter settings, that are used for evaluation of the matching distance, speed, and the window size, are listed. In contrary to the rest, Setting 1 and 2 do not apply visual saliency dropout. Therefore a larger number of feature points are needed, as non-relevant, low salient features are not neglected. Setting 1 and 3 represent very loose conditions, where a lot of FP are expected. Adversely, Setting 2, 5 and 6 are very severe, where the only difference between 5 and 6 is the utilization of hierarchical clustering. Setting 4 is supposed to be somewhere between very relaxed and strict.

Table 5.2: Six different parameter settings used for matching distance, speed and window size evaluation.

Parameter	1	2	3	4	5	6
Saliency Dropout	FALSE	FALSE	50	50	50	50
Saliency Filter	FALSE	150	FALSE	150	150	150
x/y Range Filter	FALSE	10%	FALSE	FALSE	10%	10%
Dispersion Threshold	FALSE	10%	FALSE	FALSE	10%	10%
Image Scaling Factor	2	1	1	1	1	1
Feature Points	200	300	150	150	150	150
Clustering	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE

Image scaling does not impact computational performance but suppresses visual details, which are relevant for optical flow feature tracking. Therefore, it is not applied in strict conditions.

Evaluation was applied to the top level system outcome, which are the final matches encoded as calibration message, and not on single path level, due to the lack of reliable ground truth, as neither the KLT feature detection, the ID allocation, nor the tracking is deterministic. Therefore, slots of match cache length (5 seconds) are evaluated as being True Positives (TP), TN, FP or FN.

TP: Only motion paths (at least one), caused by the black moving dot are classified as a match, while the participant is tracing the target.

FP: Whenever a single motion path is encoded as calibration message, not belonging to the moving target, either while tracing or during the non-task phase.

TN: Not one matched path in a whole slot of the non-task phase.

FN: Participant is following the black dot, but the system does not detect any matches.

Positives (P): Total number of positive evaluation slots ($TP + FN$).

Negatives (N): Total number of negative evaluation slots ($TN + FP$).

With the resulting values, the following figures are calculated for a final evaluation and conclusions: True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR), Positive Predictive Value (PPV) and Accuracy (ACC).

Matching Distance Evaluation

After determining the six different parameter settings, the first chosen value to be evaluated is the DTW matching distance. For the following experiments a relatively large window size of 2000 ms is applied, which is suggested by Vidal et al. for reliable correlation of more than eight concurrent motion trajectories in the Music shop display application scenario [VBG13]. Similarly, only one velocity (175 px/s) value is chosen for demonstrating the influence of matching distance evaluation, as the effect remains the same for every speed.

The returned warping distance of the FastDTW implementation [APS09; SC07] is a positive similarity value in a range from about 1 to 60 for the supplied normalized time series of the NPC prototypical system. Therefore, a broad range of values has been tested in the first experiments, which led to the following three values for comprehensive evaluation: 15, 20 and 25.

In Figure 5.2 the accuracy (ACC) values for each of the three velocities as well as for all six parameter settings are reported. Concerning the accuracy, the best results are achieved with a matching distance of 20, which can also be seen in Table 5.3 where the average accuracy is given over all experiments, whereas distances of 15 and 25 perform quite equally.

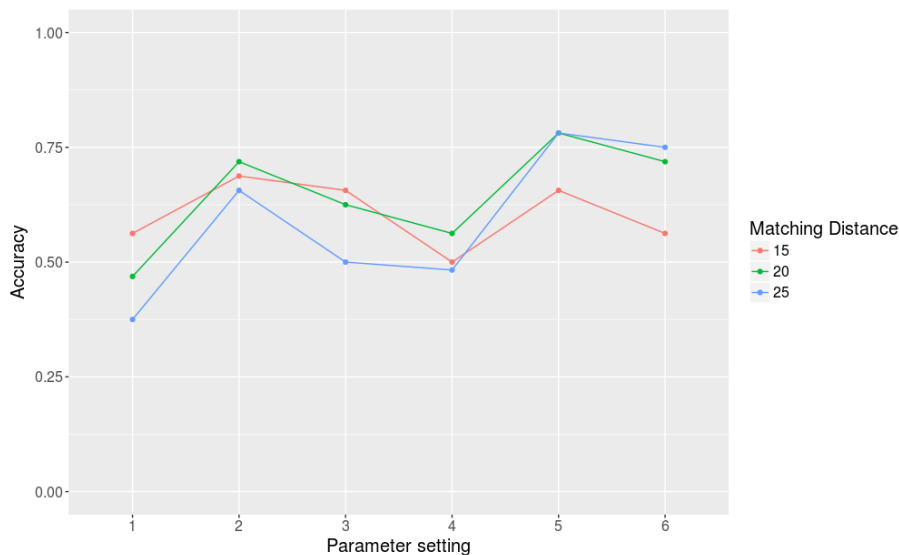


Figure 5.2: Accuracy for three matching distances and six parameter settings.

More important for the NPC system than the absolute accuracy, are the specificity or TNR and the precision or PPV. This is due to the sensitivity of the system against FP, as one wrong match can distort the whole calibration. The average values, reported in Table 5.3

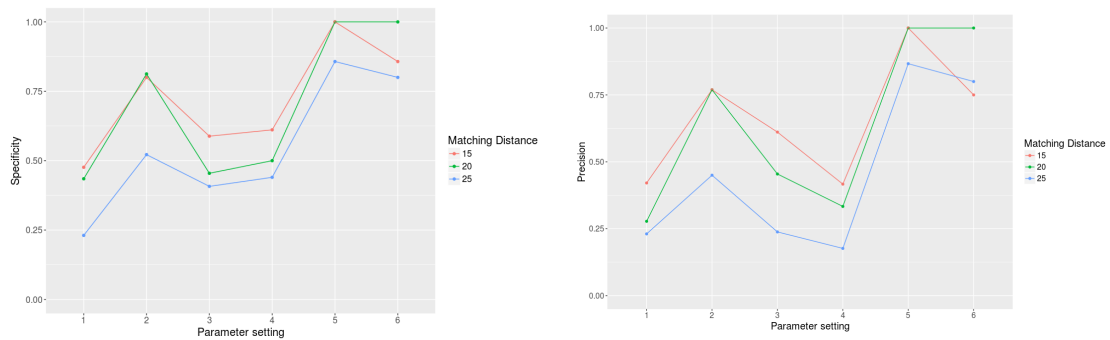
5 Evaluation

Table 5.3: Results for distance evaluation. * values are reported as average values over all six parameter settings.

Distance	Accuracy*	Specificity*	Detection Rate*	Precision*	Max. Precision
15	0,60	0,72	0,52	0,66	1
20	0,65	0,70	0,67	0,64	1
25	0,59	0,54	0,86	0,46	0,87

would, therefore, indicate that a matching distance of 15 is the best choice, as both, highest specificity and precision are achieved.

However, as seen in Figure 5.3, a more detailed look on the results shows that a matching distance of 20 yields equal or even better values for a subset of parameter settings (2, 5, 6). Therefore, the matching distances 15 and 20 are both considerable, depending on the parameter setting. Contrarily, a matching distance of 25 is too high to fulfill the requirements of the proposed prototype, concerning avoidance of false detections.



(a) Specificity for three matching distances and six parameter settings. (b) Precision for three matching distances and six parameter settings.

Figure 5.3: Avoidance of false positives is crucial for NPC, therefore maximal precision and specificity are crucial.

Conversely, a matching distance of 25 yields the best results concerning the detection rate, also referred to as sensitivity or TPR. Figure 5.4 shows, that none of the other two choices are able to outperform the distance of 25 for any parameter setting, but for setting 3 and 4 a matching distance of 20 is competitive.

This conclusion, in combination with the former evaluated figures of specificity and precision, suggests that a matching distance of 20 offers a good trade-off: While providing satisfying robustness concerning FP, the sensitivity of the system is acceptable.

Window Size Evaluation

In the next evaluation step, the effect of the window size is investigated. Therefore, analogous to the matching distance evaluation, three appropriate window sizes, from

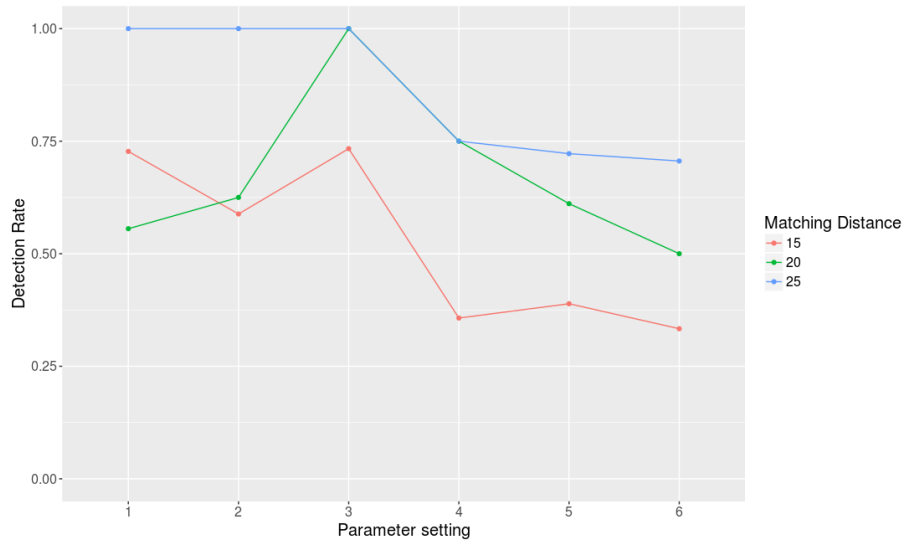


Figure 5.4: Sensitivity for three matching distances and six parameter settings.

relatively small to quite large are chosen (1000, 2000 and 2500 ms), and tested with each parameter setting. For all experiments the matching distance is 20, which is determined to fit best, and again, the slowest velocity (175 px/s) is used.

To clarify, which window size is evaluated in this section, it is stated that the window size corresponds to the duration of the motion path time series, which gets correlated to the time series representing the eye movements of equal duration.

Table 5.4: Results for window size evaluation.* values are reported as average values over all six parameter settings.

Window Size	Accuracy*	Specificity*	Detection Rate*	Precision*	Max. Precision
1000 ms	0,49	0,44	0,86	0,38	0,81
2000 ms	0,65	0,70	0,67	0,64	1
2500 ms	0,71	0,79	0,66	0,76	1

Resulting figures, averaged over all six parameter settings for the different window lengths are addressed in Table 5.4. The largest window size of 2500 ms overtakes the others in nearly all evaluation disciplines, except the TPR. This supports the outcome of the matching distance evaluation: in order to gain a robust setting, avoiding any FP a lower detection rate has to be accepted.

Considering Figure 5.5, where the accuracy (ACC) values are shown in a more comprehensive manner, there are two cases, namely, parameter setting 2 and 5, where the accuracy achieved with a motion path duration of 2000 ms exceeds the one with 2500 ms. In all other cases ACC is the highest in case of the largest window size.

5 Evaluation

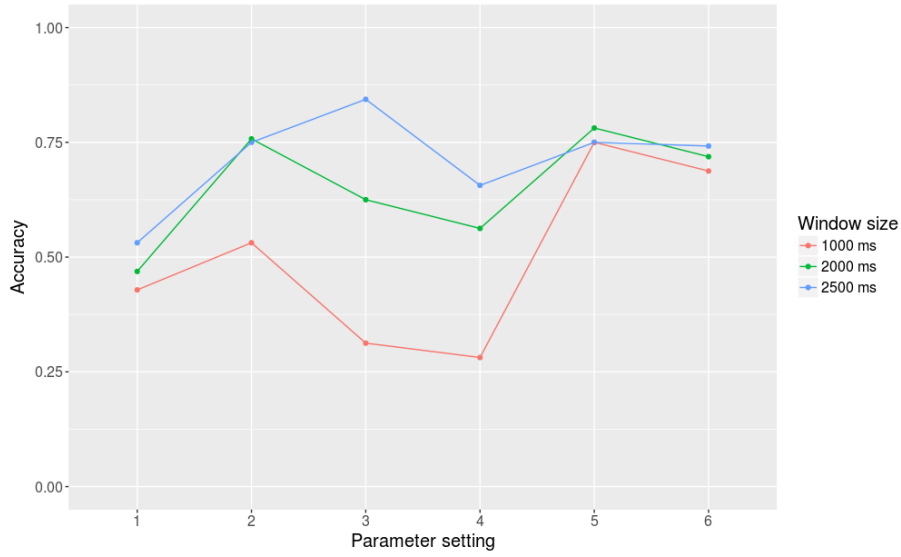
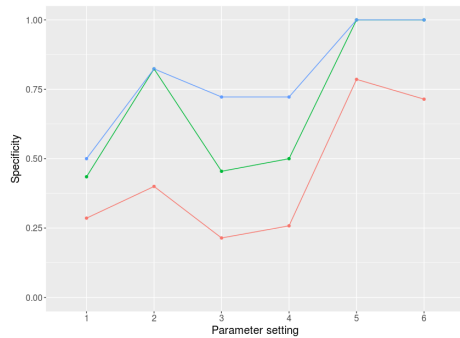
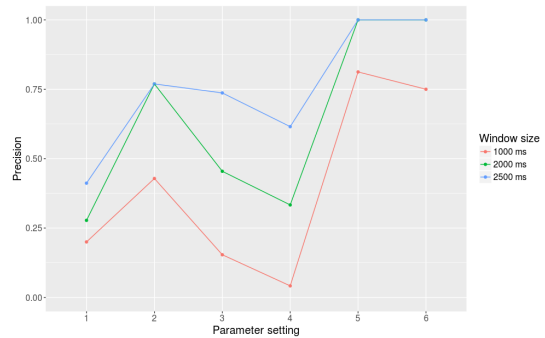


Figure 5.5: Accuracy for three window sizes and six parameter settings.

The highest accuracy value is achieved with parameter setting 3 and a window size of 2500 ms. Nevertheless, the successful application of NPC is again more dependent on the specificity and the precision, which are reported for all 18 experiments in Figure 5.6.



(a) Specificity for three window sizes and six parameter settings.



(b) Precision for three window sizes and six parameter settings.

Figure 5.6: Avoidance of false positives is crucial for NPC, therefore maximal precision and specificity are crucial.

Utilizing 2000 and 2500 ms window size, a 100 % TNR and PPV can be achieved, for parameter setting 5 and 6. Both figures are quite low for all experiments performed with the smallest window size. This leads to the conclusion that 1000 ms is too short for gaining a robust calibration system, even though the other window sizes are beaten by far considering the sensitivity, shown in Figure 5.7.

The detection rate of the remaining window sizes, is almost the same in the first three settings, whereas for parameter setting 4 and 5 results of 2000 ms is better, and vice versa for setting 6.

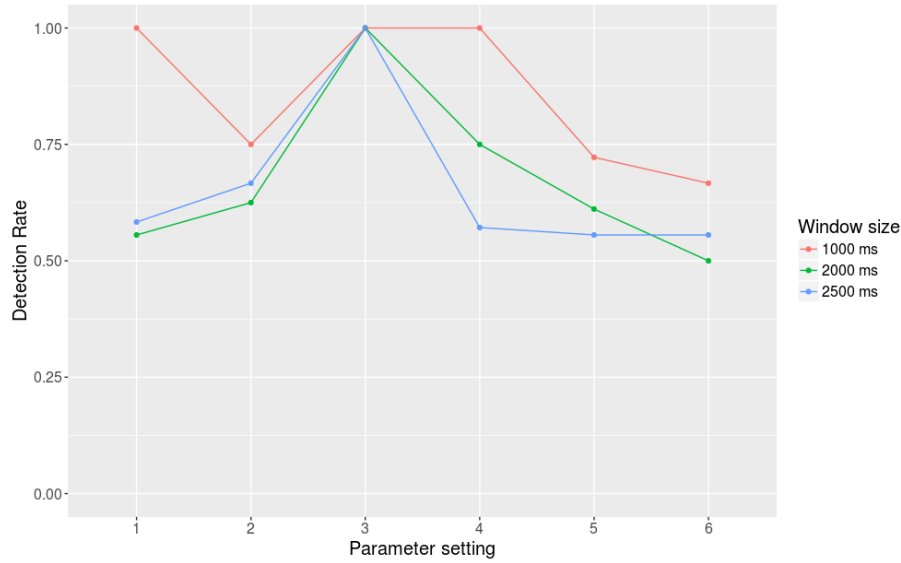


Figure 5.7: Sensitivity for three window sizes and six parameter settings.

In the case of NPC, where theoretically a number of up to 300 different motion trajectories, depending on the parameter setting and filter configuration, have to be correlated, larger window sizes are clearly preferred. According to the final parameter setting, and the application scenario both, 2000 ms as well as 2500 ms, can be an appropriate choice, according to this evaluation.

Speed Evaluation

The analysis of the matching distance and window size already yielded to a system that is capable of eliminating 100 % of FP in this scenario, although sensitivity suffers. The aim of the speed evaluation is to find the most suitable velocity, that maximizes the detection rate of the system. For experiments so far, only the slowest speed, namely, 175 px/s has been considered. Furthermore, the question, whether Natural Pursuit Calibration is competitive to Pursuit Calibration concerning the system's detection rate, is addressed in the following section.

For the evaluation of the influence of the velocity to the motion path detection rate, only relevant paths, corresponding to the black moving dot are taken into account. The no-task part of the recording is not considered, as it is assumed, in a final system setup, there are no FP matches whatsoever. Therefore, the occurrence of FP and TN is impossible by definition.

In Table 5.5 the absolute values of TP and FN are listed for every velocity, parameter setting and participant. Once again, it has to be clarified, how the absolute number of positive samples emerges. Due to top-level system evaluation, time slots of 5 seconds are considered as either being TP or FN. Therefore, the sum of TP and FN for one participant and one parameter setting, divided by three (different evaluated window sizes), is equal to the

Table 5.5: TP and FN values for parameter setting 1 to 6, for the three different velocities and participants, over all window sizes.

		522 px/s						261 px/s						175 px/s					
		1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
TP	P1	5	2	6	5	2	2	10	8	11	8	8	6	15	9	17	12	11	12
	P2	0	1	2	3	2	3	9	9	10	9	9	10	14	11	16	12	12	12
	P3	5	4	6	4	4	4	9	7	11	10	11	11	13	8	15	10	11	11
	Total	10	7	14	12	8	9	28	24	32	27	28	27	42	28	48	34	34	35
FN	P1	4	7	3	4	7	7	2	4	1	4	4	6	3	9	1	6	7	6
	P2	6	5	4	3	4	3	3	3	2	3	3	2	4	7	2	6	6	6
	P3	4	5	3	5	5	5	3	5	1	2	1	1	5	10	3	8	7	7
	Total	14	17	10	12	16	15	8	12	4	9	8	9	12	26	6	20	20	19

number of 5-second time slots. It can be seen in Table 5.6 that the duration of the tracing sequence slightly differs for the participants, which leads to a different number of total evaluation samples for the fastest velocity.

Table 5.6: Duration and number of evaluation slots of single recordings for each participant and velocity.

Participant	P1			P2			P3		
Velocity [px/s]	522	261	175	522	261	175	522	261	175
Duration [s]	12	18	28	9	20	30	12	19	30
Evaluation slots	3	4	6	2	4	6	3	4	6

The averaged results over all participants and parameter settings for the detection rate of the 54 experiments are reported in Table 5.7, grouped by velocity and window size. The highest sensitivity is achieved with a speed of 261 px/s and a window size of 1000 ms. Further, the medium velocity yields the best detection rate for all window sizes.

The results so far, suggest that the system is not capable of tracing fast movements, but for small window sizes the sensitivity of the system is competitive, for a velocity of 522 px/s, as shown in Figure 5.8a. Nevertheless, the window size evaluation resulted in an indisputable exclusion of a path length of only 1000 ms.

For a window size of 2000 ms, where TPR is presented in detail in Figure 5.8b, the speed of 261 px/s overtakes 175 px/s for every settings except parameter setting 1. Moreover,

Table 5.7: Detection rates for different window sizes and velocities.

Velocity	522 px/s			261 px/s			175 px/s		
Window Size	1 s	2 s	2.5 s	1 s	2 s	2.5 s	1 s	2 s	2.5 s
Sensitivity*	0.65	0.44	0.17	0.88	0.75	0.66	0.75	0.67	0.63

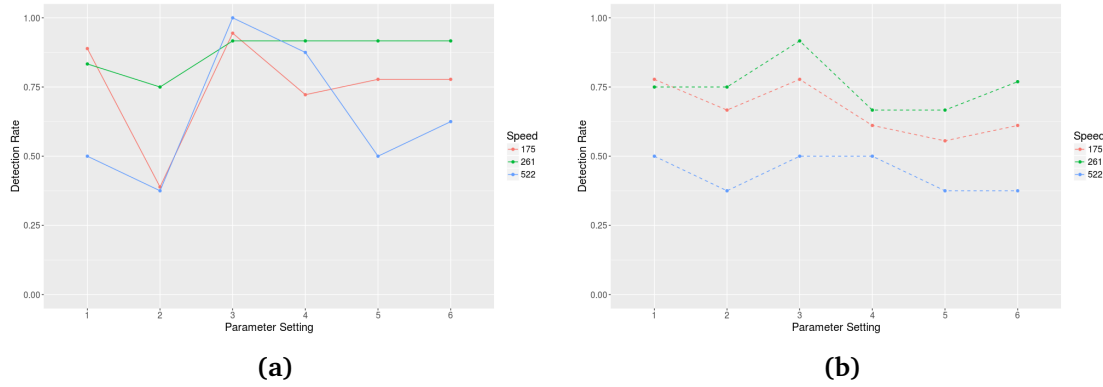


Figure 5.8: Overall detection rates for different velocities and window sizes (1000 ms and 2000 ms).

Figure 5.9 reveals a drop of detection rate for all velocities if the visual saliency filter is added (parameter setting 2, 4, 5 and 6), the sharpest effect can be seen for the slowest velocity.

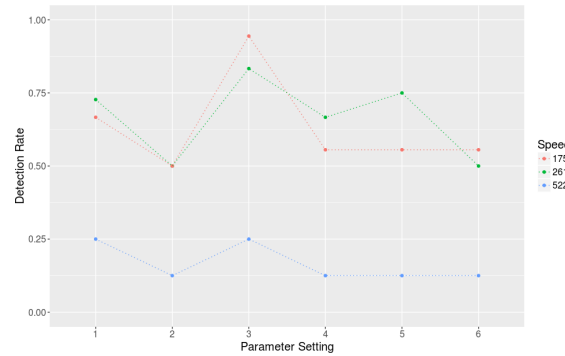


Figure 5.9: Overall detection rates for different velocities for window size 2500 ms.

Additionally, Figure 5.10 gives a good impression, of how the velocities have an impact on the systems detection rate and shows clearly that the medium velocity outperforms the others.

At last, the average detection rates over all six parameter settings and the three different window sizes are reported for every speed value in Figure 5.11, including the standard error of the mean.

5.2.5 Discussion

The conducted experiments revealed the best matching distance of 20 and preferable long window sizes of 2000 or 2500 ms, depending on the application. Additionally, an analysis of the regularization techniques and the parameter settings can be done with the available data. The appliance of Saliency Dropout, x/y Range Filter, and the Dispersion Threshold

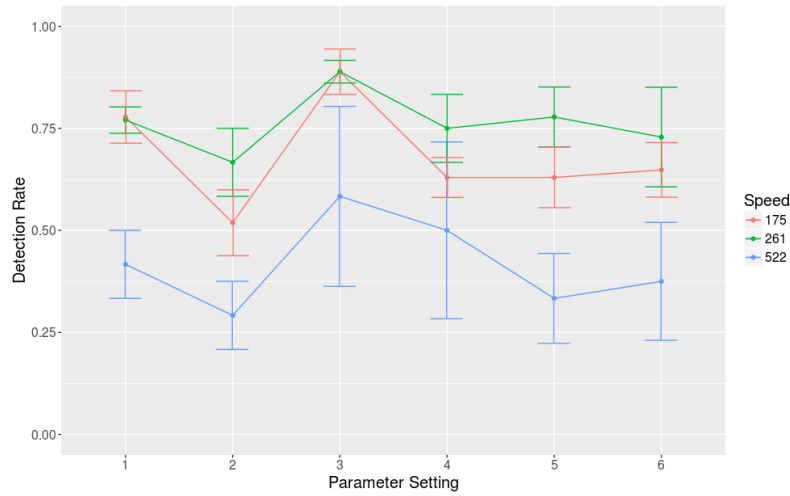


Figure 5.10: Average detection rates over all window sizes for each velocity.

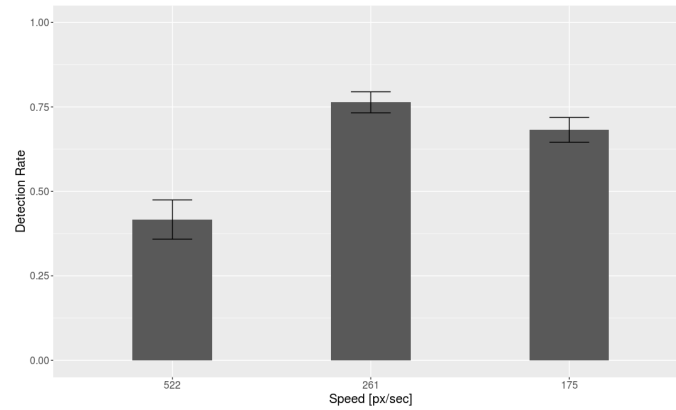


Figure 5.11: Average detection rates for different velocities, over all window sizes and parameter settings, including the standard error of the mean.

lead to the most effective elimination of FP. In contrary, the Saliency Filter led to a drop of the detection rate without causing a reasonable increase of the Specificity and Precision. Furthermore, the use of hierarchical clustering can neither be assessed as necessary, nor irrelevant in general, it has to be decided individually for every application scenario. If the target velocity is very low, the figures suggest better detection rate applying clustering, whereas a slight decrease is noticed with medium velocity, cf. Figure 5.10.

As a conclusion of the Primary Parameter Evaluation, research question Q1 can be answered partly.

Do gained results of Pursuit Calibration [Pfe+13] apply to Natural Pursuit Calibration, in terms of **attention detection**, **target velocity**, path and sample size?

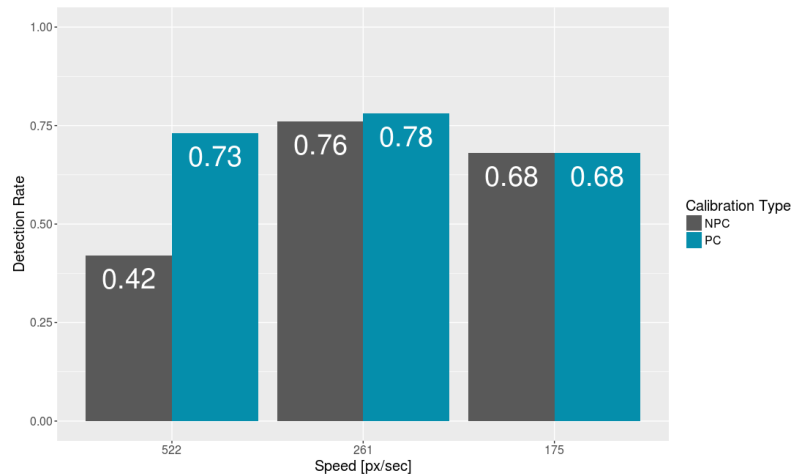


Figure 5.12: Overall detection rates for different velocities, compared the Pursuit Calibration results [Pfe+13].

Compared to Pursuit Calibration, introduced by Pfeuffer et al. [Pfe+13], NPC achieves similar results for attention detection for a target velocity of 261 and 175 px/s, as seen in Figure 5.12. However, the proposed system is not able to reach competitive sensitivity for high velocities, using optical flow feature tracking.

Nevertheless, referring to the explicit NPC scenario described in the concept in chapter 3, where the movement of the own finger is used for calibration, there is only limited control of velocity, but the results of speed evaluation act as a guideline.

5.3 System Evaluation

The second step of the evaluation process aims to provide a proof of concept, to compare NPC to prevailing n-point calibration techniques and to answer the research question Q2 as well as the remaining aspects of Q1. To do so, the path or trajectory design, the sample size and the usability of NPC have to be examined.

Therefore, explicit Natural Pursuit Calibration is executed by generating motion trajectories with one's own finger. This is done in two manners, using a visual marker as well as optical flow feature tracking. The performance in terms of (i) calibration accuracy, (ii) completion time and (iii) user demand is compared to Natural Feature Calibration and Screen Marker Calibration, both provided by Pupil Labs [KPB14].

5.3.1 Setup

For the system evaluation, an industrial scenario is simulated where a worker is placed in front of his workstation, wearing an eye tracker. The eye tracker can be used for any

reason, the only considered requirement for this study is a calibration accuracy as high as possible, within the whole working place.

It is compulsory, that conditions for each calibration technique are comparable, and none of them are favored due to the study design. Thus, the parallax error, which is explained in detail in the next section, has to be considered as a source of distortion.

Parallax Error

Due to the vertical displacement of the pupil center and the eye tracker's egocentric camera, the calibration distance plays an important role in maximizing calibration accuracy.

Figure 5.13 demonstrates the effect of a discrepancy between the calibration and interaction plane. The view vector of the egocentric camera and the pupil intersect at the calibration plane at a certain angle. Since the entry angle equals the exit angle, the deviation of the predicted and actual gaze point increases with the distance of the interaction to the calibration plane. This discrepancy is known as the parallax error and has to be accounted for comparing different calibration techniques.

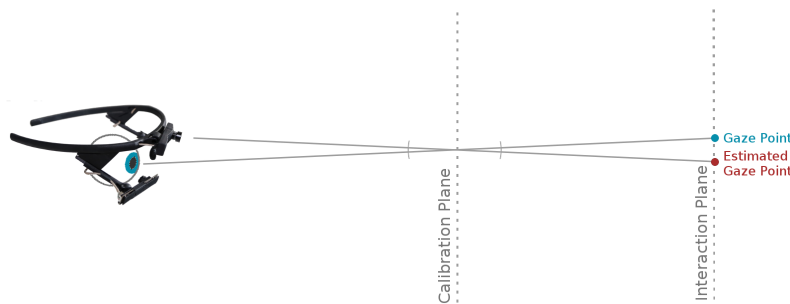


Figure 5.13: Illustrating the effect of displacement of the pupil and the egocentric camera mounted on the eye tracker.

To ensure equal preconditions for each calibration method, the parallax error should be approximately identical for each method. Additionally, to keep the discrepancy of estimation and real gaze position as small as possible, the distance of the calibration and intersection plane should be minimized [Eva+ 12]. Accordingly, different calibration areas and one common evaluation area are introduced for the study setup.

Calibration Areas

The chosen calibration techniques require distinct infrastructures, therefore executing all of them in a common setting is not possible without disadvantaging one of them. As a result, different calibration areas are constructed, offering best conditions for every method. However, all of them share one evaluation area, which represents a workstation.

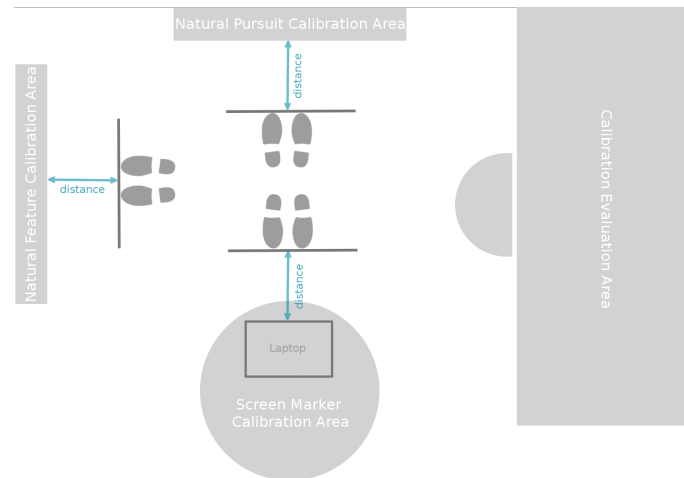


Figure 5.14: Study setup: Three different calibration areas and one evaluation area, where the participant is positioned at arm's length distance.

The final study setup is shown in Figure 5.14, where it can be seen that the distance of every calibration area to the user is equivalent. The distance is chosen to fit the selected application scenario, where the worker's hands are occupied, performing some task. The area of interest, where the parallax error should be minimized, is within arm's length of the worker.

Screen Marker Calibration

The first of two calibration techniques, NPC is compared to, is the Screen Marker Calibration, offered by the Pupil Labs framework. This method is chosen as it can be performed unaccompanied by any supervisor or assistant. A calibration process is executed in front of a display, dwelling at the center of a concentric marker, which is shown at 5 different positions successively: at the center of the screen and in all four corner areas.



Figure 5.15: Screen Marker Calibration Area, position illustrated in the left, and an image of the real setup in the right. Calibration is performed on a screen.

In the proposed setup, a laptop is positioned on a table, cf. Figure 5.15. To perform Screen Marker Calibration (SMC) a person is located at approximate arm's length distance to the table, facing the screen.

Natural Feature Calibration

The second calibration method of choice is Natural Pursuit Calibration, also included in the Pupil Labs framework, due to the calibration process being performed off-screen. Natural Feature Calibration (NFC) requires two persons and coordinated communication to accomplish calibration. Either, the user wearing an eye tracker, or the supervisor has to select a set of points, where the user is fixating consecutively, while the supervisor confirms the current point for calibration by clicking the corresponding position in the world camera's frame on the screen.

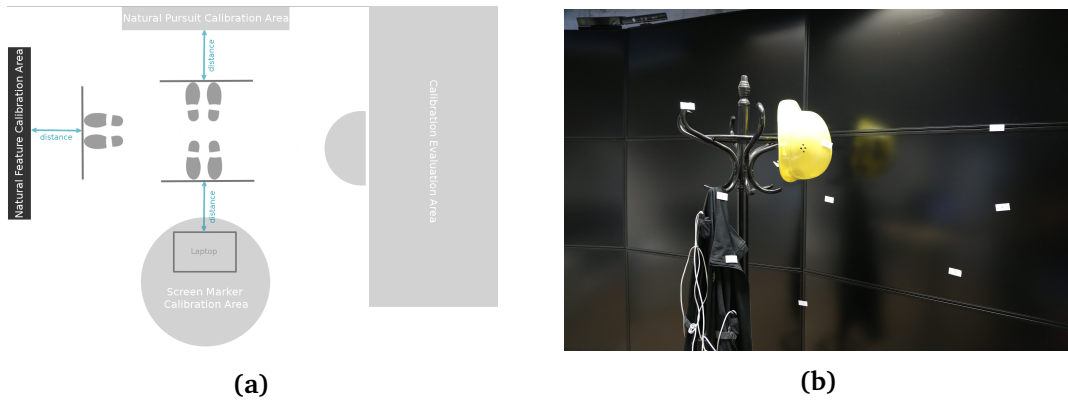


Figure 5.16: Natural Feature Calibration Area, position illustrated in the left and an image of the real setup in the right. Nine numbered labels are used as an assistance.

Calibration points are predefined in the proposed setup, in order to counterfeit communication failures, by labeled stickers from 1 to 9, cf. Figure 5.16b. Figure 5.16a shows, that the user is again placed in front of the calibration error at arm's length distance facing the predefined points. Whenever the user is fixating one point, he/she says the labeling out loud and keeps fixating, until the supervisor gives an auditive command to switch to the next point.

Natural Pursuit Calibration

As already stated before, explicit NPC is executed by tracing the movement of one's own finger with their eyes, in order to demonstrate the potential of the proposed system. Two different tracking methods are applied, to extract the motion trajectory out of the egocentric video stream, using (i) a square fiducial tracker and (ii) an optical flow feature tracker. In the following, appliance of (i) is called Natural Pursuit Marker Calibration (NPC Marker), and (ii) is referred to as Natural Pursuit Finger Calibration (NPC Finger).

For both applies, that the user's field of view and the egocentric camera's FOV differ. Therefore, an expert user with a lot of training would perform way better than a novice. To overcome this learning effect, the calibration area is defined by a rectangle on a wall representing the camera's FOV, when the user is positioned at arm's length distance.

The calibration process is terminated when match detections are distributed over the whole FOV. Therefore, a grid overlay is shown to the study supervisor. The grid cells are dyed by a different color, once motion path entries of a detected match are within, and the calibration is stopped if all cells are colored. This also matches the termination criteria for SMC and NFC, because both approaches are executed until reference points distributed over the whole screen are collected.

Natural Pursuit Marker Calibration

In order to validate the proposed system, Natural Pursuit Calibration is performed using a visual marker in the first step. A squared fiducial is placed on the user's hand, and calibration is performed by moving the hand, see Figure 5.17a, within the mentioned calibration area, highlighted in Figure 5.17b. In order to make conclusions concerning the trajectory design, three different designs are executed for NPC Marker: rectangular, circular and random.

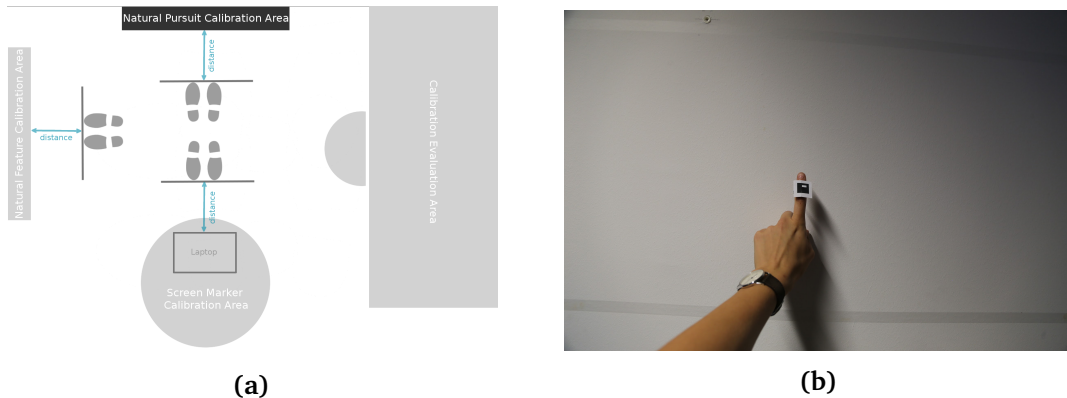


Figure 5.17: Natural Pursuit Calibration Area, position illustrated in the left and an image of the real setup in the right. A visual marker placed on the finger is used for performing NPC Marker calibration.

Natural Pursuit Finger Calibration

The last calibration method to be evaluated is NPC performed without any annotation of the environment, just by following one's own finger, like demonstrated in Figure 5.18. This is done to validate the optical flow approach. Nevertheless, some meta information is used to reach valuable results, as big calibration targets worsen the calibration accuracy [Pfe+13], the uppermost matching feature path is used for calibration.

5 Evaluation



Figure 5.18: Natural Pursuit Calibration Area, position illustrated in the left and an image of the real setup in the right. Only the person's finger is used for calibration.

Evaluation Area

A universal evaluation area is defined, emphasized in Figure 5.19a, to calculate accuracy for each calibration method. For none of the approaches calibration is performed inside the evaluation scenery, to avoid giving preference to any method.

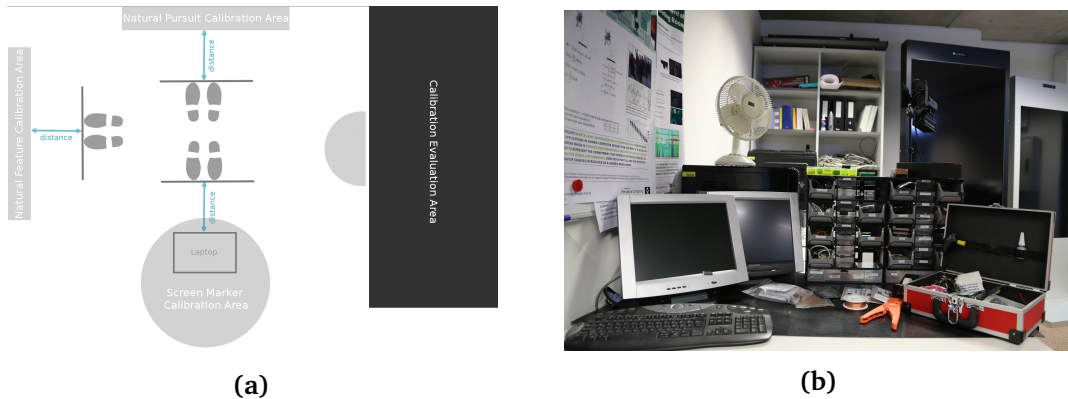


Figure 5.19: Calibration Evaluation Area, position illustrated in the left and an image of the real setup in the right. All objects are approximately at arm's length distance, if the user is placed appropriately on the desk.

Nevertheless, as already mentioned calibration and evaluation plane should be approximately within the same distance. For the chosen scenario, presented in Figure 5.19b, a workplace is simulated, where all work utensils are placed within reach, therefore, the interaction plane is at arm's length distance, if the user is positioned at the workplace appropriately.

Furthermore, it is important that the evaluation area is a 3D environment, and not a planar plane such as a display, as this matches the real environment in an industrial context.

5.3.2 Participants

Six subjects (3M/3F) participated in the system test, aged between 25 and 62 years ($M = 37.83$, $SD = 17.63$). Two of them had completed an apprenticeship, another two obtained the A-level certificate, one possessed a Bachelor's and the sixth one a Master's degree. Two out of six wear glasses regularly, one of them also uses contact lenses and wore them when he conducted in the system test. The use of contact lenses did not interfere with the eye trackers accuracy, whereas there were several problems caused by makeup, especially mascara. The participants rated their prior experience with eye tracking at 4.67 ($SD 2.65$) at a 0 to 10 Likert scale (from low to high experience), four of them had already performed Natural Feature Calibration once, and one had experiences with all calibration techniques.

5.3.3 Procedure

Upon arrival, people were introduced to the scenario and the eye tracking device. To evaluate the qualitative part of the system test, a questionnaire was provided, which was split into different sections. The first section, containing demographics as well as questions regarding prior experience, was answered immediately after the introduction phase. Then the eye tracker was adjusted to the face geometry of the participant, and a checklist containing the following five points was performed for each eye:

- Pupil is located in the center of the camera image
- Pupil is tracked when looking far right
- Pupil is tracked when looking far left
- Eyelashes do not cover the eye and therefore lead to distortions
- 3D-model is built

This was followed by a short training phase and then the actual system test was performed.

Introduction and Training Phase

The introduction was supported by a slide-show where, as a first step, the participant received an explanation how the Pupil Labs eye tracker works, why the device has to be calibrated, and what calibration actually means. All four calibration techniques (SMC, NFC, NPC Marker, NPC Finger) and the study procedure were described. It is important to mention, that none of the advantages and disadvantages of the different calibration techniques were mentioned to the users beforehand, as the subjective opinion about the single methods should not be biased by any prior knowledge. For example, users attention

was not expressly drawn to the necessity for a second person when performing NFC or to the need for a screen device when using SMC.

The introducing scenario was followed by a short training phase, aiming to show the user appropriate speed for performing NPC. Therefore, the subject moved their finger in an arbitrary manner within the NPC calibration area, and after six successful matches, training was completed. Natural Pursuit Calibration was the only technique that required any training phase, as it was the only one where the user had to actively perform something without guidance.

System Test

The design of the system test was an iteration of alternating calibration and evaluation phase. Each method was repeated three times, in order to eliminate coincidence up to a certain degree, the total number of executions for each method can be seen in Table 5.8. Each calibration procedure was followed by accuracy evaluation, which will be explained in more detail in the following section. After a completion of all attempts for each method, additional parts of the questionnaire, containing the NASA Task Load Index [Har06], was answered by the participant in order to gain insights on qualitative measures, in particular, the subjective usability and user demands.

Table 5.8: Total number of executions for each calibration technique, taking several variations into account. The order of techniques and variations is just for demonstration issues, and varied for each participant in the actual sessions.

Calibration Technique	Variation	Total Executions
SMC	-	3
NFC	-	3
NPC Marker	Rectangular	9
	Circular	
	Random	
NPC Finger	-	3

Similar, to the introduction scenario each calibration attempt was supported by a checklist, to ensure the same procedure for each process and participant:

- All prior data is reseted (eye and calibration) and the eyes are checked
- Pupil is started and right calibration technique is selected
- Person is calibrating in predefined position
- Accuracy evaluation plugin is selected
- Accuracy is tested within evaluation area

Furthermore, the sequential order of calibration techniques (including variations) were designed in a randomized way and varied for each participant.

In average, the extensive system evaluation took about 90 minutes per participant. Therefore the number of participants is quite limited.

Accuracy evaluation

As stated before, a fictional workplace scenario is used for evaluating calibration accuracy. This, for example, would be an electronic control box. It is important that the evaluation area is a 3D environment, and not a planar plane like a display, as this matches the real environment in an industrial context. Visual markers are placed within the workplace while evaluating the current calibration attempt, see Figure 5.20, and the Pupil Labs Accuracy Plugin is used for calculating the angular accuracy of the calibration model in a slightly adjusted way.



Figure 5.20: Accuracy evaluation using visual markers placed within a fictional working place.

5.3.4 Results

The system test resulted in 108 calibration attempts, where 104 were classified as successful and only four as failed (1× SMC, 1× NFC, 2× NPC Finger). In the proposed scenario, a calibration was considered successful, if the resulting angular accuracy, reported in degrees of visual angle, was above 10°.

In the following sections the calibration accuracy, the calibration completion time as well as qualitative feedback, gained from the questionnaire was analyzed in detail for each of the four calibration techniques. In addition, the effect of trajectory design was examined.

Accuracy Evaluation

The resulting accuracy is the most crucial measure to be evaluated, regarding calibration. Therefore, in this section, the calibration accuracy values of all four methods are presented. The results include three attempts, for six people for every calibration method, except NPC Marker, where every participant performed nine tries.

Figure 5.21 gives a first impression of the performance of each technique, in terms of the calibration accuracy. As already mentioned before, the accuracy is reported as the displacement of the estimated gaze in degrees of visual angle, where the lower is the better.

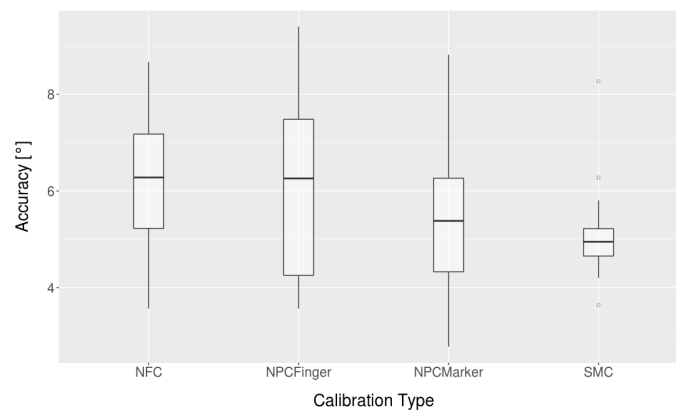


Figure 5.21: Calibration accuracy, reported as the displacement of the estimated gaze point in degrees of visual angle - the lower the better.

The best accuracy was reached with Screen Marker Calibration, which resulted in an average of 5.14° ($MD = 4.95^\circ$), followed by Natural Pursuit Marker Calibration, reaching a mean accuracy of 5.46° ($MD = 5.38^\circ$). Natural Pursuit Finger Calibration and Natural Feature Calibration performed very similarly in terms of accuracy, gaining an average of 6.08° ($MD = 6.26^\circ$) and 6.17° ($MD = 6.28^\circ$). It can be seen, that SMC delivered most constant results, despite a few outliers, whereas there was a great variance in the outcomes of NPC Finger.

This can be seen also in Figure 5.22, where the variability within users is low for SMC, except for participant 2, indicating a very high reproducibility. Similar results are achieved with NPC Marker, where all users were able to perform very constant over all attempts. For NPC Finger calibration participant 2 stands out again, as the variance of the three different calibration attempts is very high.

This can be analyzed in more detail in Figure 5.23, where it can be seen that participant 2 was only able to perform NPC Finger successfully two out of three times. Once it resulted in a very good and once in a very bad accuracy measure.

Further, the four failed calibration attempts can be identified (participant 1: NPC Finger, participant 2: NPC Finger, participant 3: NFC and SMC). Moreover, it can also be seen that

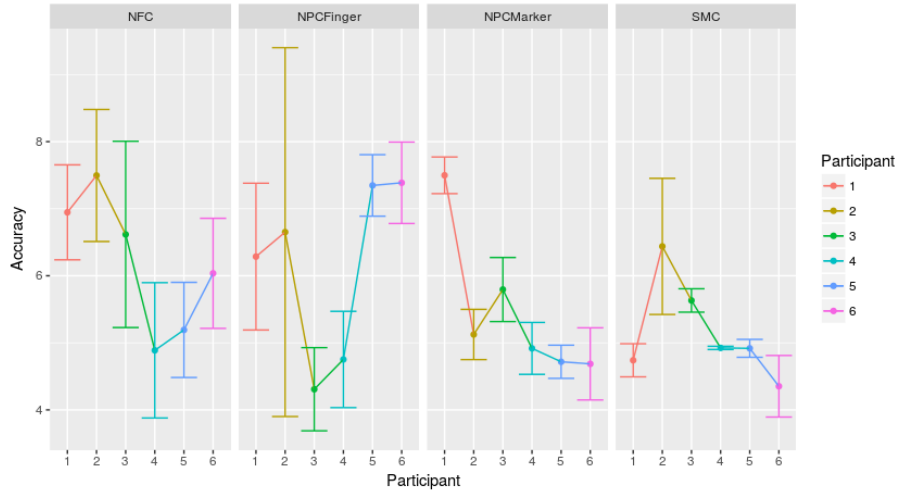


Figure 5.22: Detailed calibration accuracy, for each calibration technique and each participant, over all runs.

there is not only a high variance of calibration accuracy within techniques, but there are some users that perform more constantly than others. For example, participant 4 achieved very similar calibration accuracy for each of the four methods.

Additionally, it has to be taken into account that all values, shown in Figure 5.23, are absolute, except for NPC Marker. Here every bar is the mean of three attempts: each for each trajectory design. These will be evaluated in more detail in a following section.

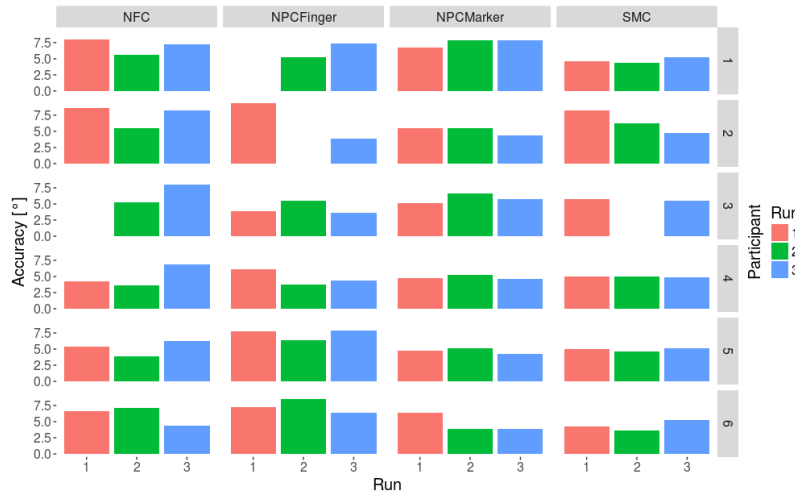


Figure 5.23: Calibration accuracy, for each participant, each run and each calibration type.

Another representation of the accuracy results, can be seen in Figure 5.24, where values over all participants are shown for each method and run.

This representation suggests a learning effect when considering NPC Finger and NPC Marker as the accuracy improved with every calibration attempt, although it has been tried

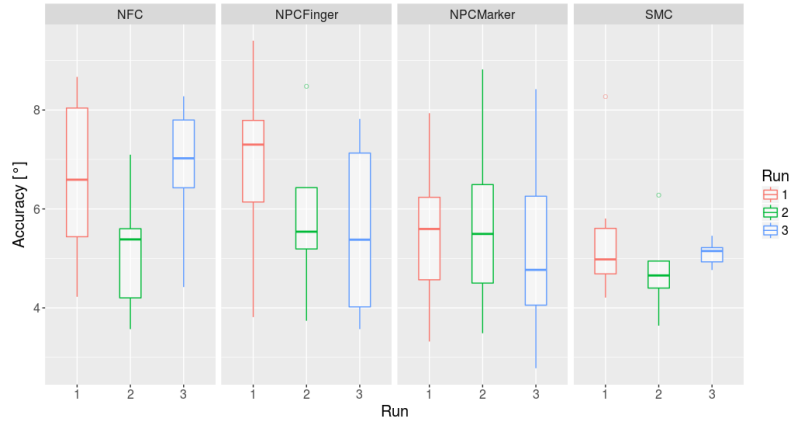


Figure 5.24: Calibration accuracy, over all participants, for each calibration technique, split into first, second and third run.

to overcome the effect by indicating the calibration area, as well as the training phase to get to know the best performance speed. Nevertheless, more research would be necessary in order to prove this statement. Contrary, there is a deterioration of calibration accuracy for the last run for NFC.

We further performed a within-subjects factorial analysis of variance (ANOVA), to investigate if the method has a significant effect on the accuracy. The results ($F_{5,90} = 2.10$, $p = 0.073$, $\eta^2 = 0.104$) suggest that there is no significant effect of the calibration type on the calibration accuracy.

Calibration Time Evaluation

In contrary to the results gained for calibration accuracy, the calibration completion time is identified as the major drawback. This is illustrated in Figure 5.25, where it can be seen clearly, that SMC and NFC overtake both, NPC using visual marker as well as optical feature tracker.

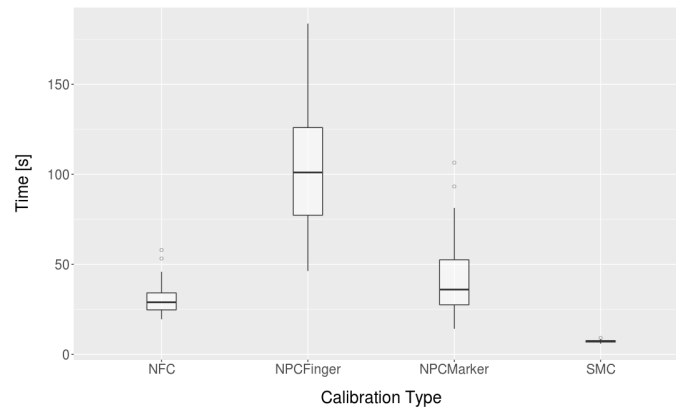


Figure 5.25: Completion time measurement, for each calibration technique

Considering the completion time, the best result is, again, achieved by using SMC, as this is a predefined process of similar length, taking about 6 to 10 seconds ($M = 7.3$, $MD = 7.3$).

Moreover, NPC Marker, with an average calibration time of 42.4 seconds ($MD = 36$ s), can nearly keep up with NFC ($M = 31.5$, $MD = 28.9$). However, NPC Finger is outperformed by far, having an average completion time of 103 seconds ($MD = 101$). Further details can be seen in Figure 5.26, where time is listed for each method and run over all participants.

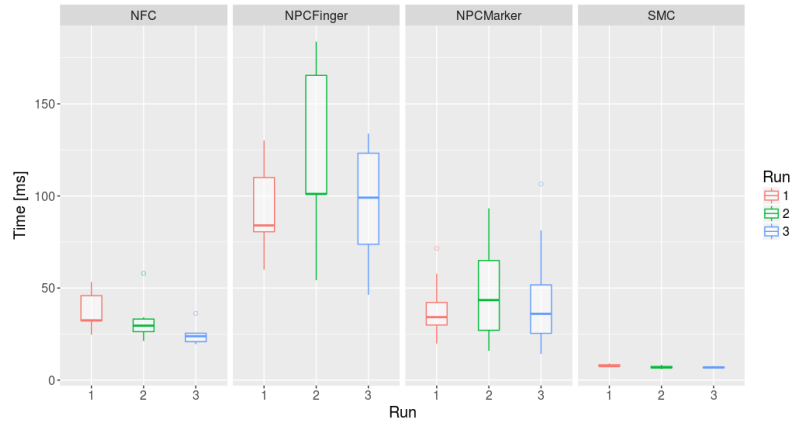


Figure 5.26: Completion time measurement, for each calibration technique and for each run, over all participants.

Additionally, a significant effect of the type on calibration time was found by again using ANOVA ($F_{5,90} = 38.54$ $p < 0.05$ $\eta^2 = 0.68$).

Pattern Design

So far, NPC Marker has been monitored, using average values over all three trajectory designs. In this section, the effect of the pattern or trajectory design on the calibration accuracy and time will be evaluated.

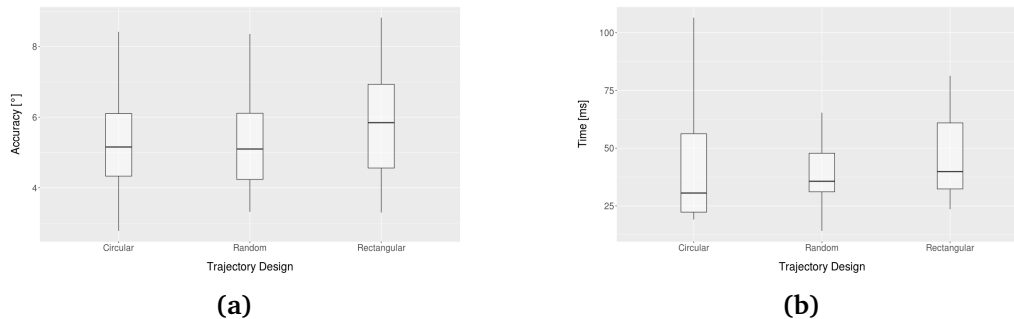


Figure 5.27: Evaluation of three different trajectory designs: calibration accuracy on the left and completion time on the right, both over all participants and runs.

Therefore, accuracy values over all participants and runs for each of the pattern shapes, are listed in Figure 5.27a. It can be seen, that all three designs perform quite similarly, with Circular movement reaching an accuracy of 5.32° in average (MD = 5.16°). Rectangular resulting in a mean of 5.89° (MD = 5.85°), and the best result for Random movement with 5.15° in average (MD = 5.1°). The corresponding calibration times can be seen in Figure 5.27b and are 43 seconds (MD = 30.6) for Circular, 46.4 seconds (MD = 39.9) for Rectangular and 37.7 seconds (MD = 35.7) for Random trajectory design in average.

Both, average accuracy and time, show a minor favoritism for Random pattern design, which actually matches also the users subjective rating, shown in Figure 5.28. Three out of six liked the Random movement most, while two preferred Rectangular and only one voted for Circular as being the best choice.

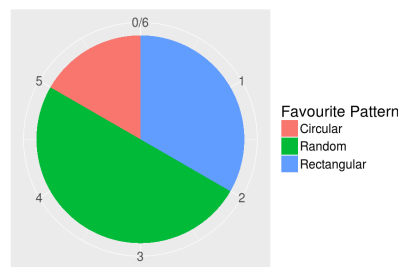


Figure 5.28: Users subjective ranking of the performed trajectory designs.

Additionally, ANOVA was applied to identify any significant effect of the trajectory design on the calibration accuracy. But again, the result suggests no significant effect of pattern design on calibration accuracy ($F_{2,15} = 56.9$ $p = 0.58$ $\eta^2 = 0.071$).

Usability Evaluation

In addition to quantitative evaluation, every participant was asked to complete a NASA TLX [Har06], after all attempts of each calibration method. They were also requested to perform a ranking in the end of the whole system evaluation. The NASA TLX asks for the following qualitative measures:

Mental Demand: *How mentally demanding were the tasks?*

(Very low: 0, Very high: 10)

Physical Demand: *How physically demanding were the tasks?*

(Very low: 0, Very high: 10)

Temporal Demand: *How hurried or rushed were the paces of the tasks?*

(Very low: 0, Very high: 10)

Performance: *How successful were you in accomplishing what you were asked to do?*

(Perfect: 0, Failure: 10)

Effort: *How hard did you have to work to accomplish your level of performance?*

(Very low: 0, Very high: 10)

Frustration: *How insecure, discouraged, irritated, stressed, and annoyed were you?*

(Very low: 0, Very high: 10)

The results can be looked up in Figure 5.29, where the rating values are averaged over all participants for each calibration technique.

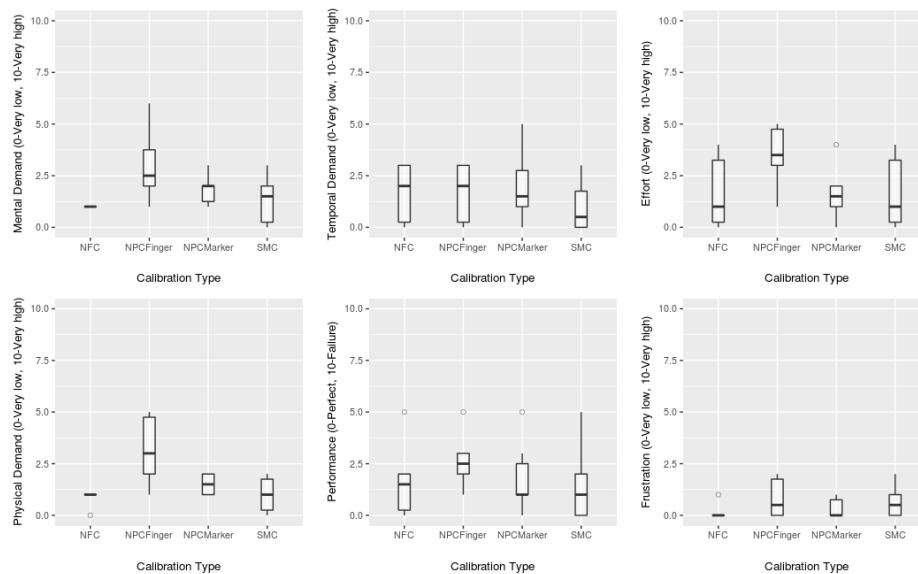


Figure 5.29: Results of the NASA TLX questionnaire, which was performed for each calibration technique separately.

Participants stated NPC Finger as most mentally demanding, but still staying in the lower half of the scale, followed by NPC Marker, SMC and NFC. Similar behavior, but with an even stronger effect, can be seen for the physical demand and effort. Furthermore, while performing NFC and NPC Finger people felt most hurried or rushed. Frustration was rated very low for all of the four calibration techniques. Nevertheless a minor favor to NFC and NPC Marker could be identified. Lastly, the performance (the lower is the better) demonstrates very similar results for NFC, NPC Marker and SMC, whereas participants felt a little less successful when carrying out NPC Finger. Despite slight differences for individual figures, all demand values, however, stayed quite low.

In addition, at the end of the system test, each user was asked to perform a subjective ranking for the four calibration methods, and which method they liked most and which least and give a short explanation. The resulting ranking for each participant can be seen in Table 5.9 and the personal statements about the preferred and most disliked method is listed in Table 5.10.

The personal opinion was quite similar across all participants, stating that they liked SMC and NFC for being very fast and easy, and saw the calibration time as well as the physical participation as a major drawback for NPC.

Table 5.9: Subjective ranking of the four calibration types of each participant.

	SMC	NFC	NPC Marker	NPC Finger
Participant 1	2	1	3	4
Participant 2	1	2	4	3
Participant 3	1	3	4	2
Participant 4	2	1	4	3
Participant 5	2	4	1	3
Participant 6	3	2	1	4

Table 5.10: Personal statements, why a technique was liked most or least.

Best liked method	
SMC	fast execution, fast and simple
NFC	simple, very simple and fast
NPC Marker	off-screen, quicker than finger, efficient
Least liked method	
SMC	screen
NPC Marker	slow, standing, tiring
NPC Finger	too slow, exhausting for the hand, hard to stare at same point, duration

5.3.5 Discussion

As a conclusion, to the system evaluation, the remaining parts of the research question Q1, as well as Q2, are answered in the following.

Do gained results of Pursuit Calibration [Pfe+13] apply to Natural Pursuit Calibration, in terms of attention detection, target velocity, **path** and **sample size**?

Pfeuffer et al. [Pfe+13] suggests that best calibration accuracy is reached when the target is moving along the border in a rectangular way. The evaluation of the path design conducted within the system test, compared rectangular movement at the border of the user's FOV, as well as circular and random movement, and no significant effect on the accuracy was found. Furthermore, additional investigation of the sample size is necessary. So far, a stopping criterion was applied that ensures an equal distribution of reference points, nevertheless the proposed system is capable of on-line accuracy detection by comparing the estimated gaze point to detected motion trajectories, but this is part of possible future work.

Is Natural Pursuits Calibration improving **subjective usability**, in terms of acceptance, user demand and time required, **without a significant loss of accuracy**, compared to Natural Feature Calibration?

Within the evaluated scenario, it must be emphasized that the results gained with NPC are competitive to those achieved with SMC and NFC when it regards calibration accuracy.

However, the calibration time is a major drawback so far and caused a decrease in subjective usability for the users, as this was the main argument they stated when asked what they liked the least. Additionally, the proposed Natural Pursuit Calibration requires a lot more active participation from the user, especially physical movement, which is reflected in the NASA TLX.

Moreover, the fact that participants were not told about further considerations, such as the need for a second person when performing NFC, or the necessity of an additional screen device for SMC, explicitly, should be mentioned once again, as these could cause a shift in user perception. Participants, apart from one, has had very few experience with eye tracking before. Therefore they may not have been aware of the advantages explicit NPC provides. Namely, not requiring any digital user interface, any annotation of the environment or assistance of a supervisor. Additionally, another great potential, namely implicit NPC, was not evaluated during this study.

6 Future Work and Conclusion

Future Work

In order to evaluate the whole potential of the proposed technique, a long-term field study, combining explicit and implicit scenarios, is necessary. Especially, the implicit scenario is not at all evaluated within this work, but seems to be very promising in order to improve user acceptance, as it can be executed in the background without the participant noticing. Additionally, there is a chance that permanent application emphasizes the advantages provided, namely, performing calibration on one's own, without assistance or any supporting device or annotation.

Furthermore, the system is capable of on-line accuracy calculation, which is also not included in the evaluation scenario, and holds great potential in various ways. Firstly, the user can be notified if calibration is distorted and an explicit calibration process, to update the calibration model, is required. Moreover, the need for an explicit stopping criterion gets obsolete, as calibration can be terminated if a certain accuracy is reached. More investigation is also necessary, to highlight the advantages of on-going calibration. The appliance of the concept led to good accuracy results within the proposed evaluation scenario, but this only addresses the short-time behavior.

In addition to more extensive evaluation, the system design can be enhanced in future work as well, in order to increase the robustness of the system, which could then lead to a time improvement.

To give an example, one should assume that the system is able to detect the fingertip on each frame, just like a visual marker, and a motion path can be built upon this data. Therefore, along with specific context information, a solid object detection algorithm would be required. Lately, promising results for object detection were gained using Neural Networks architectures, like those proposed by Dai et al. [Dai+16], Liu et al. [Liu+16] or He et al. [He+16], to only mention a few. The trend towards highly accurate detection and performance enhancement indicates a high potential to include such techniques into real-time systems, as the proposed one.

Beside object detection issues, additional physical characteristics of the human visual perception system, such as the stereoscopic vision, can be considered. Stereoscopic vision describes the process of the human brain, combining the view of the right and left eye, which differ in perspective and FOV, in order to perceive the environment in a three-dimensional manner. Therefore, a perspective transformation applied to each eye data

sequence before calculating the similarity to any detected motion path could increase the system performance.

Moreover, the steady progression of eye tracking techniques and the increasing amount of collected eye movement data still reveals precious information about human perception, which can be used to enhance visual saliency models, that in return can be used to advance calibration and gaze-based interaction techniques like NPC.

Conclusion

Within the scope of this work, Natural Pursuit Calibration was introduced, which is an unobtrusive calibration technique for mobile eye trackers, that applies smooth pursuit calibration, without guidance or artificial annotation of the environment. This is done by extracting any occurring motion trajectory out of the egocentric video stream and then correlating it to the user's eye movement. NPC can be either performed explicitly, by generating motion trajectories on one's own, or implicitly, without active user participation, depending on the scenario. Additionally, the concept of on-going calibration is proposed, which updates certain areas of the calibration model, whenever reference points are available, instead of rejecting the entire model.

A two-level evaluation process was conducted in order to (i) optimize the parameter settings and detect the system's performance in terms of attention detection. It was, however, also executed to (ii) gain insights about resulting calibration accuracy, time and user acceptance employing an explicit NPC scenario. Due to calibration time and the active and physical participation required for the explicit scenario, usability does not increase. Nevertheless, The detection rate of the proposed system performs similarly to a state-of-the-art calibration approaches up to a certain speed. Moreover, it is quite competitive with common calibration techniques, regarding calibration accuracy.

List of Figures

2.1	Eight possible stroke components in the right, and three sample gestures in the left image [DS07].	15
2.2	The concept of Pursuits: a person tracing motion trajectories with the eye in order to interact [VBG13].	16
2.3	A grid-like calibration pattern on the left, versus the motion trajectory of the dragonfly illustrated on the right [Ren+11].	17
2.4	Different application scenarios for Pursuit Calibration: a moving dot (a,b), shooting stars (c) and floating words (d) [Pfe+13].	18
3.1	Person wearing an eye tracker is watching a moving car, which triggers an update of the calibration model.	19
3.2	Schematic overview roughly representing the system design.	19
3.3	Conceptual illustration of the continuous calibration model, where the dark gray dots represent the original model data. The light gray dots demonstrate outdated data, that is replaced by a new match (blue dots).	20
3.4	Explicit interaction scenario, where the movement of the finger acts as gesture guidance.	21
4.1	A binocular Pupil Labs eye tracker, with two infrared cameras, which are capturing each eye, and a wide-range world camera, that is recording the egocentric view.	23
4.2	Communication between Pupil Labs framework and the implemented system via ZeroMQ.	24
4.3	Detailed overview of the system architecture.	24
4.4	Features are identified, traced over a number of frames and motion noise is filtered, in order to extract valid motion trajectories.	25
4.5	Examples of squared binary fiducials.	26
4.6	Two different frames where corresponding features are graphically associated, using the Pyramid Kanade-Lucas-Tomasi feature tracker.	27
4.7	A lot of features in non-salient regions on the left image, versus a very reduced number of features on the right, to demonstrate visual saliency dropout.	28
4.8	The interface used for validation of clustering algorithm on the left and the corresponding extracted motion trajectories on the right.	29
4.9	A lot of noise is filtered out, while no non-noise trajectory is eliminated.	29
4.10	Motion paths and corresponding eye movements are compared, in order to detect motion matches.	30

4.11 A Python plugin, implemented within the Pupil Labs framework, takes reference points via an API and triggers the calibration process.	32
5.1 In order to discover appropriate parameter settings subjects are located in front of a display, following a black moving dot.	34
5.2 Accuracy for three matching distances and six parameter settings.	37
5.3 Avoidance of false positives is crucial for NPC, therefore maximal precision and specificity are crucial.	38
5.4 Sensitivity for three matching distances and six parameter settings.	39
5.5 Accuracy for three window sizes and six parameter settings.	40
5.6 Avoidance of false positives is crucial for NPC, therefore maximal precision and specificity are crucial.	40
5.7 Sensitivity for three window sizes and six parameter settings.	41
5.8 Overall detection rates for different velocities and window sizes (1000 ms and 2000 ms).	43
5.9 Overall detection rates for different velocities for window size 2500 ms. . .	43
5.10 Average detection rates over all window sizes for each velocity.	44
5.11 Average detection rates for different velocities, over all window sizes and parameter settings, including the standard error of the mean.	44
5.12 Overall detection rates for different velocities, compared the Pursuit Calibration results [Pfe+13].	45
5.13 Illustrating the effect of displacement of the pupil and the egocentric camera mounted on the eye tracker.	46
5.14 Study setup: Three different calibration areas and one evaluation area, where the participant is positioned at arm's length distance.	47
5.15 Screen Marker Calibration Area, position illustrated in the left, and an image of the real setup in the right. Calibration is performed on a screen.	47
5.16 Natural Feature Calibration Area, position illustrated in the left and an image of the real setup in the right. Nine numbered labels are used as an assistance.	48
5.17 Natural Pursuit Calibration Area, position illustrated in the left and an image of the real setup in the right. A visual marker placed on the finger is used for performing NPC Marker calibration.	49
5.18 Natural Pursuit Calibration Area, position illustrated in the left and an image of the real setup in the right. Only the person's finger is used for calibration.	50
5.19 Calibration Evaluation Area, position illustrated in the left and an image of the real setup in the right. All objects are approximately at arm's length distance, if the user is placed appropriately on the desk.	50
5.20 Accuracy evaluation using visual markers placed within a fictional working place.	53
5.21 Calibration accuracy, reported as the displacement of the estimated gaze point in degrees of visual angle - the lower the better.	54
5.22 Detailed calibration accuracy, for each calibration technique and each participant, over all runs.	55
5.23 Calibration accuracy, for each participant, each run and each calibration type.	55

5.24 Calibration accuracy, over all participants, for each calibration technique, split into first, second and third run.	56
5.25 Completion time measurement, for each calibration technique	56
5.26 Completion time measurement, for each calibration technique and for each run, over all participants.	57
5.27 Evaluation of three different trajectory designs: calibration accuracy on the left and completion time on the right, both over all participants and runs. .	57
5.28 Users subjective ranking of the performed trajectory designs.	58
5.29 Results of the NASA TLX questionnaire, which was performed for each calibration technique separately.	59

List of Tables

5.1	Three different velocities chosen for parameter evaluation.	34
5.2	Six different parameter settings used for matching distance, speed and window size evaluation.	36
5.3	Results for distance evaluation. * values are reported as average values over all six parameter settings.	38
5.4	Results for window size evaluation.* values are reported as average values over all six parameter settings.	39
5.5	TP and FN values for parameter setting 1 to 6, for the three different velocities and participants, over all window sizes.	42
5.6	Duration and number of evaluation slots of single recordings for each participant and velocity.	42
5.7	Detection rates for different window sizes and velocities.	42
5.8	Total number of executions for each calibration technique, taking several variations into account. The order of techniques and variations is just for demonstration issues, and varied for each participant in the actual sessions.	52
5.9	Subjective ranking of the four calibration types of each participant.	60
5.10	Personal statements, why a technique was liked most or least.	60

List of Abbreviations

ACC	Accuracy.	36
ANOVA	analysis of variance.	56
DTW	dynamic time warping.	30
FN	False Negatives.	28
FOV	field of view.	11
FP	False Positives.	28
FPR	False Positive Rate.	36
HCI	human-computer interaction.	11
IoT	Internet of Things.	11
KLT	Kanade-Lucas-Tomasi.	26
N	Negatives.	36
NFC	Natural Feature Calibration.	48
NPC	Natural Pursuit Calibration.	19
NPC Finger	Natural Pursuit Finger Calibration.	48
NPC Marker	Natural Pursuit Marker Calibration.	48
NPI	Natural Pursuit Interaction.	21
P	Positives.	36
PC	Pursuit Calibration.	34
PPV	Positive Predictive Value.	36
SMC	Screen Marker Calibration.	47
TN	True Negatives.	28
TNR	True Negative Rate.	36
TP	True Positives.	36
TPR	True Positive Rate.	36

Bibliography

- [APS09] T. Abeel, Y. V. d. Peer, Y. Saeys. “Java-ml: A machine learning library.” In: *Journal of Machine Learning Research* 10.Apr (2009), pp. 931–934 (cit. on pp. 31, 37).
- [Abe16] P. Abeles. *BoofCV*. <http://boofcv.org/>. 2016 (cit. on pp. 25, 26).
- [Ach+08] R. Achanta, F. Estrada, P. Wils, S. Süsstrunk. “Salient region detection and segmentation.” In: *Computer Vision Systems* (2008), pp. 66–75 (cit. on p. 27).
- [BRT09] A. Bulling, D. Roggen, G. Tröster. *Wearable EOG goggles: eye-based interaction in everyday environments*. ACM, 2009 (cit. on p. 15).
- [CCL04] F. Chang, C.-J. Chen, C.-J. Lu. “A linear-time component-labeling algorithm using contour tracing technique.” In: *computer vision and image understanding* 93.2 (2004), pp. 206–220 (cit. on p. 25).
- [CJ11] J. Chen, Q. Ji. “Probabilistic gaze estimation without active personal calibration.” In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, pp. 609–616 (cit. on p. 11).
- [Cor+10] F. Corno, A. Gale, P. Majaranta, K.-J. Räihä. “Eye-based direct interaction for environmental control in heterogeneous smart environments.” In: *Handbook of ambient intelligence and smart environments*. Springer, 2010, pp. 1117–1138 (cit. on p. 14).
- [Cor01] A. Corradini. “Dynamic time warping for off-line recognition of a small gesture vocabulary.” In: *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on*. IEEE. 2001, pp. 82–89 (cit. on p. 31).
- [Cor14] iMatix Corporation. *ZMQ Distributed Messaging*. <http://zeromq.org>. 2014 (cit. on p. 23).
- [Cym+14] D. H. Cymek, A. C. Venjakob, S. Ruff, O. H.-M. Lutz, S. Hofmann, M. Roetting. “Entering PIN codes by smooth pursuit eye movements.” In: *Journal of Eye Movement Research* 7.4 (2014) (cit. on p. 16).
- [DAH12] M. L. Dybdal, J. S. Agustin, J. P. Hansen. “Gaze input for mobile devices by dwell and gestures.” In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM. 2012, pp. 225–228 (cit. on pp. 14, 15).
- [DD90] L. F. Dell’Osso, R. B. Daroff. “Eye movement characteristics and recording techniques.” In: *Neuroophthalmology* 2 (1990), pp. 279–297 (cit. on p. 13).

- [DHI17] W. Delamare, T. Han, P. Irani. “Designing a gaze gesture guiding system.” In: *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM. 2017, p. 26 (cit. on pp. 16, 21).
- [DLWD07] A. De Luca, R. Weiss, H. Drewes. “Evaluation of eye-gaze interaction methods for security enhanced PIN-entry.” In: *Proceedings of the 19th australasian conference on computer-human interaction: Entertaining user interfaces*. ACM. 2007, pp. 199–202 (cit. on p. 14).
- [DS07] H. Drewes, A. Schmidt. “Interacting with the computer using gaze gestures.” In: *IFIP Conference on Human-Computer Interaction*. Springer. 2007, pp. 475–488 (cit. on p. 15).
- [DS09] H. Drewes, A. Schmidt. “The MAGIC touch: Combining MAGIC-pointing with a touch-sensitive mouse.” In: *IFIP Conference on Human-Computer Interaction*. Springer. 2009, pp. 415–428 (cit. on p. 14).
- [Dai+16] J. Dai, Y. Li, K. He, J. Sun. “R-fcn: Object detection via region-based fully convolutional networks.” In: *Advances in neural information processing systems*. 2016, pp. 379–387 (cit. on p. 63).
- [Est+15] A. Esteves, E. Velloso, A. Bulling, H. Gellersen. “Orbits: Gaze interaction for smart watches using smooth pursuit eye movements.” In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM. 2015, pp. 457–466 (cit. on pp. 16, 19, 30).
- [Eva+12] K. M. Evans, R. A. Jacobs, J. A. Tarduno, J. B. Pelz. “Collecting and analyzing eye tracking data in outdoor environments.” In: *Journal of Eye Movement Research* 5.2 (2012), p. 6 (cit. on pp. 17, 46).
- [Fuh+16] W. Fuhl, M. Tonsen, A. Bulling, E. Kasneci. “Pupil detection for head-mounted eye tracking in the wild: an evaluation of the state of the art.” In: *Machine Vision and Applications* 27.8 (2016), pp. 1275–1288 (cit. on p. 17).
- [GE08] E. D. Guestrin, M. Eizenman. “Remote point-of-gaze estimation requiring a single-point calibration for applications with infants.” In: *Proceedings of the 2008 symposium on Eye tracking research & applications*. ACM. 2008, pp. 267–274 (cit. on p. 17).
- [HRM13] J. Hales, D. Rozado, D. Mardanbegi. “Interacting with objects in the environment by gaze and hand gestures.” In: *Proceedings of the 3rd International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction*. 2013, pp. 1–9 (cit. on p. 14).
- [HS81] B. K. Horn, B. G. Schunck. “Determining optical flow.” In: *Artificial intelligence* 17.1-3 (1981), pp. 185–203 (cit. on pp. 19, 25, 26).
- [Han+16] J. P. Hansen, H. Lund, F. Biermann, E. Møllenbach, S. Sztuk, J. S. Agustin. “Wrist-worn pervasive gaze interaction.” In: *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*. ACM. 2016, pp. 57–64 (cit. on p. 15).

- [Har06] S. G. Hart. "Nasa-Task Load Index (NASA-TLX); 20 Years Later." In: *Human Factors and Ergonomics Society Annual Meeting*. Vol. 50. 9. 2006, pp. 904–908 (cit. on pp. 52, 58).
- [Has+17] M. Haslgrübler, P. Fritz, B. Gollan, A. Ferscha. "Getting Through - Modality Selection in a Multi-Sensor-Actuator Industrial IoT Environment." In: *Proceedings of the 7th International Conference on the Internet of Things*. 2017 (cit. on p. 23).
- [He+16] K. He, X. Zhang, S. Ren, J. Sun. "Deep residual learning for image recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on p. 63).
- [Iso00] P. Isokoski. "Text input methods for eye trackers using off-screen targets." In: *Proceedings of the 2000 symposium on Eye tracking research & applications*. ACM. 2000, pp. 15–21 (cit. on p. 15).
- [Ist+10] H. Istance, A. Hyrskykari, L. Immonen, S. Mansikkamaa, S. Vickers. "Designing gaze gestures for gaming: an investigation of performance." In: *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. ACM. 2010, pp. 323–330 (cit. on p. 15).
- [JS16] R. Jacob, S. Stellmach. "What you look at is what you get: gaze-based user interfaces." In: *interactions* 23.5 (2016), pp. 62–65 (cit. on pp. 11, 14).
- [Jac91] R. J. Jacob. "The use of eye movements in human-computer interaction techniques: what you look at is what you get." In: *ACM Transactions on Information Systems (TOIS)* 9.2 (1991), pp. 152–169 (cit. on pp. 3, 5, 11).
- [Jua+05] K. Juang, F. Jasen, A. Katrekar, J. Ahn, A. T. Duchowski. "Use of eye movement gestures for web browsing." In: *Computer Science Department, Clemson University, Available as early as Jan 1* (2005), p. 7 (cit. on p. 15).
- [Jun+17] F. Jungwirth, M. Murauer, M. Haslgrübler, A. Ferscha. "Eyes are faster than Hands: An Analysis of Mobile Gaze-based Interaction with IoT Devices." In: *Proceedings of Handling the Internet of Things: Human-Computer Interaction Perspectives on IoT*. 2017 (cit. on p. 14).
- [KE08] Y. Kondou, Y. Ebisawa. "Easy eye-gaze calibration using a moving visual target in the head-free remote eye-gaze detection system." In: *Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2008. VECIMS 2008. IEEE Conference on*. IEEE. 2008, pp. 145–150 (cit. on p. 18).
- [KH16] P. Kasprowski, K. Harezlak. "Implicit calibration using predicted gaze targets." In: *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*. ACM. 2016, pp. 245–248 (cit. on pp. 11, 17).
- [KM03] I. Kang, J. G. Malpeli. "Behavioral calibration of eye movement recording systems using moving targets." In: *Journal of neuroscience methods* 124.2 (2003), pp. 213–218 (cit. on p. 18).

- [KPB14] M. Kassner, W. Patera, A. Bulling. “Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction.” In: *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication*. ACM. 2014, pp. 1151–1160 (cit. on p. 45).
- [Kan+14a] J. Kangas, D. Akkil, J. Rantala, P. Isokoski, P. Majaranta, R. Raisamo. “Gaze gestures and haptic feedback in mobile devices.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2014, pp. 435–438 (cit. on p. 15).
- [Kan+14b] J. Kangas, D. Akkil, J. Rantala, P. Isokoski, P. Majaranta, R. Raisamo. “Using gaze gestures with haptic feedback on glasses.” In: *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*. ACM. 2014, pp. 1047–1050 (cit. on p. 15).
- [Kha+16] M. Khamis, O. Saltuk, A. Hang, K. Stolz, A. Bulling, F. Alt. “Textpursuits: Using text for pursuits-based interaction and calibration on public displays.” In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2016, pp. 274–285 (cit. on p. 18).
- [LK+81] B. D. Lucas, T. Kanade, et al. “An iterative image registration technique with an application to stereo vision.” In: (1981) (cit. on p. 26).
- [Liu+15] D. Liu, B. Dong, X. Gao, H. Wang. “Exploiting eye tracking for smartphone authentication.” In: *International Conference on Applied Cryptography and Network Security*. Springer. 2015, pp. 457–477 (cit. on p. 16).
- [Liu+16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg. “Ssd: Single shot multibox detector.” In: *European conference on computer vision*. Springer. 2016, pp. 21–37 (cit. on p. 63).
- [MAŠ09] P. Majaranta, U.-K. Ahola, O. Špakov. “Fast gaze typing with an adjustable dwell time.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2009, pp. 357–360 (cit. on p. 14).
- [MB14] P. Majaranta, A. Bulling. “Eye tracking and eye-based human–computer interaction.” In: *Advances in physiological computing*. Springer, 2014, pp. 39–65 (cit. on pp. 11, 13, 15).
- [MBE10] L. Muda, M. Begam, I. Elamvazuthi. “Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques.” In: *arXiv preprint arXiv:1003.4083* (2010) (cit. on p. 31).
- [MR02] P. Majaranta, K.-J. Rähkä. “Twenty years of eye typing: systems and design issues.” In: *Proceedings of the 2002 symposium on Eye tracking research & applications*. ACM. 2002, pp. 15–22 (cit. on p. 14).
- [Maj+06] P. Majaranta, I. S. MacKenzie, A. Aula, K.-J. Rähkä. “Effects of feedback and dwell time on eye typing speed and accuracy.” In: *Universal Access in the Information Society* 5.2 (2006), pp. 199–208 (cit. on p. 14).

-
- [Mul11] F. Mulvey. "Eye anatomy, eye movements and vision." In: *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies: Advances in Assistive Technologies* (2011), p. 10 (cit. on pp. 13, 15).
 - [Møl+10] E. Møllenbach, M. Lillholm, A. Gail, J. P. Hansen. "Single gaze gestures." In: *Proceedings of the 2010 symposium on eye-tracking research & applications*. ACM. 2010, pp. 177–180 (cit. on p. 15).
 - [OM04] T. Ohno, N. Mukawa. "A free-head, simple calibration, gaze tracking system that enables gaze-based interaction." In: *Proceedings of the 2004 symposium on Eye tracking research & applications*. ACM. 2004, pp. 115–122 (cit. on pp. 11, 17).
 - [OMY02] T. Ohno, N. Mukawa, A. Yoshikawa. "FreeGaze: a gaze tracking system for everyday gaze interaction." In: *Proceedings of the 2002 symposium on Eye tracking research & applications*. ACM. 2002, pp. 125–132 (cit. on p. 17).
 - [Pfe+13] K. Pfeuffer, M. Vidal, J. Turner, A. Bulling, H. Gellersen. "Pursuit calibration: Making gaze calibration less tedious and more flexible." In: *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM. 2013, pp. 261–270 (cit. on pp. 11, 17–19, 30, 33, 34, 44, 45, 49, 60).
 - [Pfe+14] K. Pfeuffer, J. Alexander, M. K. Chong, H. Gellersen. "Gaze-touch: combining gaze with multi-touch for interaction on the same surface." In: *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM. 2014, pp. 509–518 (cit. on p. 14).
 - [RM03] T. M. Rath, R. Manmatha. "Word image matching using dynamic time warping." In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. Vol. 2. IEEE. 2003, pp. II–II (cit. on p. 31).
 - [Ren+11] P. Renner, N. Lüdike, J. Wittrowski, T. Pfeiffer. "Towards continuous gaze-based interaction in 3d environments-unobtrusive calibration and accuracy monitoring." In: *Proceedings of the Workshop Virtuelle & Erweiterte Realität 2011*. 2011 (cit. on p. 17).
 - [SC07] S. Salvador, P. Chan. "Toward accurate dynamic time warping in linear time and space." In: *Intelligent Data Analysis 11.5* (2007), pp. 561–580 (cit. on pp. 31, 37).
 - [SC78] H. Sakoe, S. Chiba. "Dynamic programming algorithm optimization for spoken word recognition." In: *IEEE transactions on acoustics, speech, and signal processing* 26.1 (1978), pp. 43–49 (cit. on p. 31).
 - [SD12] S. Stellmach, R. Dachsel. "Look & touch: gaze-supported target acquisition." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2012, pp. 2981–2990 (cit. on p. 14).
 - [SFK17] T. Santini, W. Fuhl, E. Kasneci. "CalibMe: Fast and Unsupervised Eye Tracker Calibration for Gaze-Based Pervasive Human-Computer Interaction." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM. 2017, pp. 2594–2605 (cit. on p. 18).

- [SM90] H. F. Silverman, D. P. Morgan. “The application of dynamic programming to connected speech recognition.” In: *IEEE ASSP Magazine* 7.3 (1990), pp. 6–25 (cit. on p. 31).
- [Sch+16] S. Schenk, P. Tiefenbacher, G. Rigoll, M. Dorr. “Spock: A smooth pursuit oculomotor control kit.” In: *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM. 2016, pp. 2681–2687 (cit. on p. 16).
- [Shi+94] J. Shi et al. “Good features to track.” In: *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR’94., 1994 IEEE Computer Society Conference on*. IEEE. 1994, pp. 593–600 (cit. on pp. 26, 27).
- [TK91] C. Tomasi, T. Kanade. “Detection and tracking of point features.” In: (1991) (cit. on p. 26).
- [VBG13] M. Vidal, A. Bulling, H. Gellersen. “Pursuits: spontaneous interaction with displays based on smooth pursuit eye movement and moving targets.” In: *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM. 2013, pp. 439–448 (cit. on pp. 11, 16, 18, 19, 30, 37).
- [VC08] A. Villanueva, R. Cabeza. “A novel gaze estimation system with one calibration point.” In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 38.4 (2008), pp. 1123–1138 (cit. on p. 17).
- [VCP04] A. Villanueva, R. Cabeza, S. Porta. “Eye tracking system model with easy calibration.” In: *Proceedings of the 2004 symposium on Eye tracking research & applications*. ACM. 2004, pp. 55–55 (cit. on p. 17).
- [Var08] K. Varda. “Protocol buffers: Google’s data interchange format.” In: *Google Open Source Blog, Available at least as early as Jul 72* (2008) (cit. on p. 23).
- [Vel+16] E. Velloso, M. Wirth, C. Weichel, A. Esteves, H. Gellersen. “AmbiGaze: Direct Control of Ambient Devices by Gaze.” In: *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM. 2016, pp. 812–817 (cit. on pp. 16, 19).
- [Wit+16] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016 (cit. on p. 28).
- [Wob+07] J. O. Wobbrock, J. Rubinstein, M. Sawyer, A. T. Duchowski. “Not typing but writing: Eye-based text entry using letter-like gestures.” In: *Proceedings of The Conference on Communications by Gaze Interaction (COGAIN)*. 2007, pp. 61–64 (cit. on p. 15).
- [ZMI99] S. Zhai, C. Morimoto, S. Ihde. “Manual and gaze input cascaded (MAGIC) pointing.” In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM. 1999, pp. 246–253 (cit. on p. 14).
- [ŠM12] O. Špakov, P. Majaranta. “Enhanced gaze interaction using simple head gestures.” In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM. 2012, pp. 705–710 (cit. on p. 14).

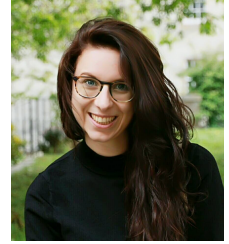
Michaela Murauer

Curriculum Vitae

Karl-Renner-Straße 3/3
4040 Linz

+43(0)664 5055431

✉ michaela.murauer@yahoo.com



Personal Details

Name Michaela Murauer
Date of birth 18.02.1992

Education

- 2016 – present **Master in Computer Science**, *Johannes Kepler University, Linz.*
Major Subject: Pervasive Computing
- 2011 – 2015 **BSc, Bachelor in Computer Science**, *Johannes Kepler University, Linz.*
- 2006 – 2011 **Höhere Technische Bundeslehranstalt, Braunau.**
Electronics and Computer Science, Major Subject: Media Engineering and Communication

Working experience

- 2016 – present **Scientific Staff**, *Institute for Pervasive Computing, Johannes Kepler University, Linz.*
Application of gaze-based interaction within distributed environments
- 2012 – 2016 **Software Engineer**, *GTN – Global Training Network GmbH, Linz.*
Extension for the e-learning platform Moodle and project based assignments

Publications

- 2017 M. Murauer, M. Haslgrübler, A. Ferscha. 2017. Natural Pursuit Calibration: Using Motion Trajectories for Unobtrusive Calibration of Mobile Eye Trackers. Proceedings of the 7th International Conference on the Internet of Things, ACM.
- 2017 M. Haslgrübler, M. Murauer, A. Ferscha. 2017. Gazor: A gaze aware Industrial IoT-based Instructor. Proceedings of the 7th International Conference on the Internet of Things, ACM.
- 2017 M. Murauer, F. Jungwirth, M. Haslgrübler, A. Ferscha. 2017. Gaze-based Action Zones: A universal interaction modality for IoT devices. Workshop on Handling the Internet of Things: Human-Computer Interaction Perspectives on IoT (in conjunction with IoT '17), Editors: to appear.
- 2017 F. Jungwirth, M. Murauer, M. Haslgrübler, A. Ferscha. 2017. Eyes are faster than Hands: An Analysis of Mobile Gaze-based Interaction with IoT Devices. Workshop on Handling the Internet of Things: Human-Computer Interaction Perspectives on IoT (in conjunction with IoT '17), Editors: to appear.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. The electronic copy is consistent with all submitted copies.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Masterarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe. Die vorliegende Masterarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

place, date, signature