Wearable Eye Tracker Calibration at Your Fingertips

Mihai Bâce
Department of Computer Science
ETH Zurich, Switzerland
mihai.bace@inf.ethz.ch

Sander Staal
Department of Computer Science
ETH Zurich, Switzerland
staals@inf.ethz.ch

Gábor Sörös
Department of Computer Science
ETH Zurich, Switzerland
gabor.soros@inf.ethz.ch

ABSTRACT

Common calibration techniques for head-mounted eye trackers rely on markers or an additional person to assist with the procedure. This is a tedious process and may even hinder some practical applications. We propose a novel calibration technique which simplifies the initial calibration step for mobile scenarios. To collect the calibration samples, users only have to point with a finger to various locations in the scene. Our vision-based algorithm detects the users' hand and fingertips which indicate the users' point of interest. This eliminates the need for additional assistance or specialized markers. Our approach achieves comparable accuracy to similar marker-based calibration techniques and is the preferred method by users from our study. The implementation is openly available as a plugin for the open-source Pupil eye tracking platform.

CCS CONCEPTS

• **Human-centered computing** → **Interaction techniques**; • **Computing methodologies** → **Computer vision**;

KEYWORDS

eye tracking, calibration, head-mounted eye tracker

ACM Reference Format:
Mihai Bâce, Sander Staal, and Gábor Sörös. 2018. Wearable Eye Tracker Calibration at Your Fingertips. In *ETRA '18: 2018 Symposium on Eye Tracking Research and Applications, June 14–17, 2018, Warsaw, Poland.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3204493.3204592

1 INTRODUCTION

As smart devices become ubiquitous, eye gaze can enhance the way in which we interact with objects around us [Majaranta and Bulling 2014]. Head-mounted eye trackers (e.g., the Pupil [Kassner et al. 2014]) enable mobile and pervasive eye tracking [Bulling and Gellersen 2010] and researchers have already experimented with attention-aware systems which are not constrained to a desktop setting [Bâce et al. 2016]. To infer the point of gaze or where the user is looking, these regression-based eye tracking systems have to go through an initial calibration phase.

Calibration is used to determine a mapping function between the eye's characteristics (e.g., the center of the pupil) and a point in the scene. High-end devices which rely on geometry-based gaze estimation can be used without a calibration phase, however, the high cost hinders the large-scale adoption. Cheaper, more flexible regression-based systems require a careful initial calibration or a re-calibration in case the device moves on the person's head. Typically, users have to collect N samples of gaze points at known locations. In a laboratory setting, a common approach is to have a second person assisting with the calibration by manually clicking on points fixated by the user. An alternative is to use calibration markers, which can be printed or displayed on a screen. They can also be placed in the environment, thus making them reliable 3D position markers. While beneficial for some applications, there are scenarios where the environment cannot be augmented with markers. For e.g., in-the-wild or outdoor studies usually cover large areas and placing markers would be impractical. When focusing on mobile settings, users might not have a screen, a dedicated marker, or assistance available. Additionally, current calibration methods are considered difficult and tedious [Pfeuffer et al. 2013].

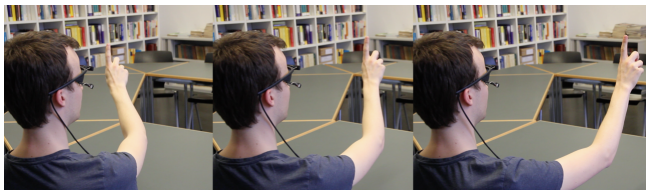

Figure 1: Users can quickly and independently calibrate a head-mounted eye tracker by pointing with their fingertip at locations in the scene. No dedicated marker, display, or additional assistance is required, enabling a calibration technique suitable for mobile and pervasive eye tracking.

In this paper, we propose a novel method which enables users to quickly and independently calibrate a head-mounted eye tracker. Users do not need any specialized markers, they can use their own fingertips (Figure 1). By pointing at locations in the scene and fixating on their own fingertip, users can easily collect calibration samples in different environments. For distances of about an arm's length, the proposed method achieves a comparable accuracy to standard marker-based calibration. We also provide an implementation of this method as a plugin for the open-source Pupil platform.[1]

2 RELATED WORK

The user-dependent calibration [Duchowski 2007] is regarded as one of the key challenges which hinders the wide adoption of eye trackers. Researchers have tried to define guidelines to help with

ETRA '18, June 14–17, 2018, Warsaw, Poland

ACM ISBN 978-1-4503-5706-7/18/06…$15.00
https://doi.org/10.1145/3204493.3204592

[1] **https://github.com/mihaibace/fingertip-calibration**

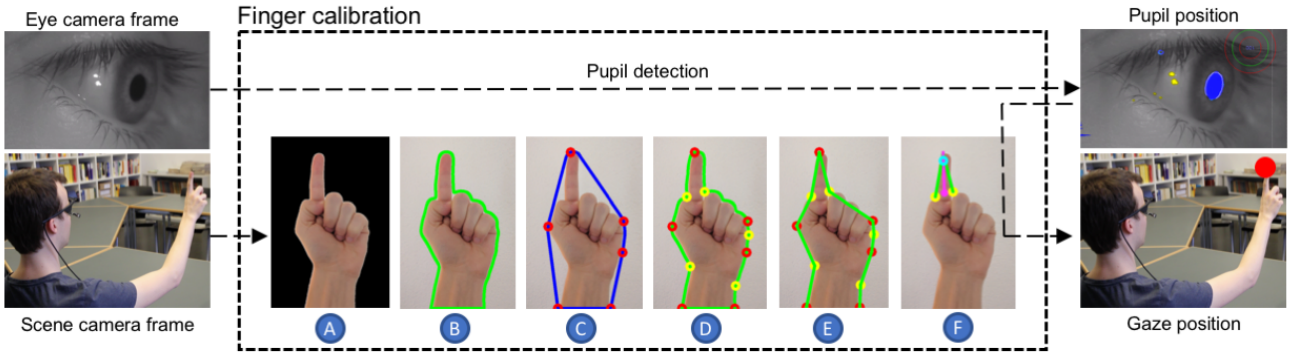

Figure 2: Method overview: Finger calibration requires hand segmentation and fingertip detection. The hand is segmented based on color. Finger candidates are points on the convex hull enclosing the hand contour. A hull point is a finger candidate if the angle between this point and two adjacent convexity defect points is smaller than a threshold ($\alpha = 60°$).

selecting the number of points to use, their location, or the type of mapping function [Kasprowski et al. 2014].

The Pupil open-source platform [Kassner et al. 2014] supports several calibration methods: Markers displayed on a screen, printed markers, or natural features. Pursuit calibration proposes the use of moving targets with a known trajectory [Celebi et al. 2014; Pfeuffer et al. 2013] and can achieve an angular error as low as 0.6°. A different approach is to leverage user events and possible interactions with a PC [Huang et al. 2016; Kasprowski and Harezlak 2016]. Egocentric visual saliency can also be used for a continuous self-calibrating eye tracker [Sugano and Bulling 2015]. CalibMe [Santini et al. 2017] is a marker-based calibration approach which facilitates fast and unsupervised collection of a large number of calibration points. Others have tried to reduce the number of re-calibrations necessary and make them more time-efficient [Lander et al. 2016].

While the above methods provide good results in different scenarios, most of them rely on a display to show the markers, a second person to assist with the procedure, or an initial calibration. We aim to simplify the initial calibration step without the use of a screen or additional assistance.

3 METHOD OVERVIEW

A gaze calibration algorithm collects sample points of known eye gaze locations in order to estimate a gaze-to-output space mapping function. We tackle this problem in a novel way. Users only have to hold their hand and fingertip in the scene camera's field of view and fixate the tip of the finger. Once a fingertip has been detected, the system will output an audio feedback to let users know that multiple sample points are being collected from that specific location. A second sound will inform users that the sampling process has finished. Users can then move their fingertip to point towards a different location to further collect calibration samples.

3.1 Hand and fingertip detection

Fingertip calibration relies on computer vision methods to segment the hand and detect the tip of the fingers. Figure 2 gives an overview of our method. The first step is to apply a blur filter on the input image to remove noise. The hand can be segmented from the background with a binary mask which is obtained from color-based segmentation in the HSV color space. In this space, only the hue channel encodes the actual color information which makes thresholding easier. In our implementation, we use a hue range from 0° to 40°, a saturation range from 12% to 60%, and a value range from 24% to 100%. The above parameters work well with light-colored skin, however, there are more sophisticated skin segmentation methods for general application [Bambach et al. 2015]. We fill small holes in the binary mask through dilation. We can then use the resulting mask to segment the hand and remove the background (Figure 2A).

Once the image has been segmented, the contour with the largest area and its convex hull are computed (Figure 2B). Since the convex hull spans the entire contour, the edge points of the convex hull can be very dense. In order to group these dense edge points together (meaning that only a single edge point belongs to a fingertip), each edge point on the convex hull is assigned to a cluster. Two points are assigned to the same cluster, if the corresponding Euclidean distance is smaller than a certain threshold d. In our implementation, we chose $d = 50\,px$ (frame resolution is $1280 \times 720\,px$) and the assignment and union of clusters is done through disjoint sets. After assigning each edge point to a cluster, the points closest to the center of their respective cluster are chosen to be the new edge points representing their cluster. Figure 2C shows the outcome of this reduction step.

To detect whether a hull point could represent a fingertip, we first calculate the convexity defects of the convex hull. Figure 2D shows the convex hull points in red, while convexity defects are shown in yellow. α represents the angle between the two vectors which are spanned by a hull point and two neighboring convexity defects. In our implementation, if the angle α is smaller than 60°, the hull point is identified as a finger (Figure 2E).

The last step of the finger detection algorithm is to correct the center of the fingertip. In its current state, the algorithm selects a convex hull point, but this point is located on the finger's contour. The hull points, which were detected as fingers, are shifted towards the center of the fingertip. The direction for this translation is obtained by looking at the angle bisector of α, which approximates

the direction of the finger. Figure 2F shows the new positions of the hull points after translating them along the finger direction towards the fingertip.

3.2 Complete System

In addition to sampling calibration points, a complete eye tracking system requires a pupil detection algorithm and a gaze mapping function. In our implementation, we use the 2D pupil detector provided with the Pupil software [Kassner et al. 2014]. This method is robust to users who wear contact lenses or eyeglasses because it does not rely on corneal reflections. The gaze mapping or the transfer function which maps pupil positions to the scene space is estimated through calibration. It has been shown that simpler polynomial functions work better when the number of sample points is low [Kasprowski et al. 2014]. Our prototype uses two bivariate polynomials (Eq. 1):

$$\begin{aligned} x_s &= A_x x_e^2 + B_x y_e^2 + C_x x_e + D_x y_e + E_x \\ y_s &= A_y x_e^2 + B_y y_e^2 + C_y x_e + D_y y_e + E_y \end{aligned} \tag{1}$$

where (x_e, y_e) are the (x, y) coordinates in the eye space and (x_s, y_s) are the target coordinates in the scene space. Mapping a two-dimensional space to a one-dimensional space can be reduced to the problem of surface fitting. Calculating the coefficients of the two polynomials is done with the Levenberg-Marquardt algorithm which solves non-linear least square problems.

4 EVALUATION

We conducted several experiments to evaluate the proposed finger calibration method. The accuracy of a calibration is measured as the average angular offset between fixation locations and the corresponding ground truth targets. The more samples are collected, the better the gaze mapping function can be estimated. Besides accuracy, the usability of an eye tracker is influenced by the calibration time.

Our experiments were performed using the Pupil eye tracker from Pupil Labs in a monocular set-up. The device was equipped with one eye camera (60 Hz) and one scene camera (30 Hz). Our finger calibration method was evaluated similarly to a typical 9-point calibration where the users were instructed to collect sample points from 9 different locations in a grid like pattern. In all the experiments, users first performed a calibration and then an accuracy test and had the freedom to choose the 9 points as they wanted.

4.1 Number of samples versus duration

In this first experiment, we analyze the number of samples necessary per location or fixation target. The Pupil open-source platform uses 30 sample points. A 9-point calibration would lead to 270 samples before removing any outliers.

The experiment was performed by an expert with more than one year experience in working with the Pupil eye tracker. The tested parameter is the number of samples per fixation. For each value of this parameter (10, 20, 30, 40, or 50), we performed three sessions. Each session involved a 9-point finger calibration followed by a 9-point finger accuracy test.

Figure 3 shows that varying the number of samples per fixation does not significantly influence the calibration accuracy. On

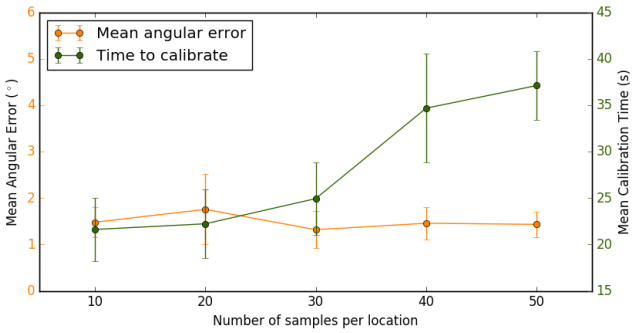

Figure 3: No. of samples per fixation vs. calibration duration.

average, the angular error varies between 1.3° and 1.8° with the minimum value being obtained for 30 calibration samples. One possible explanation for this could be the time required to collect these data points. The longer a user has to fixate a specific target, the higher the chance of a saccade or attention shift which would lead to incorrect samples. A calibration where only 10 samples per fixation are collected requires, on average, around 21 s. In contrast, collecting 50 samples increases the calibration time to around 37 s.

4.2 Preliminary user evaluation

The main goal of the proposed method is to enable users to quickly and independently calibrate an eye tracker. So far, we have evaluated the accuracy of the proposed method in a laboratory setting. In this experiment, we compared finger calibration to standard marker-based calibration. To make the comparison fair, the marker was displayed on a mobile device's screen and the calibration had to be performed similarly (Figure 4).

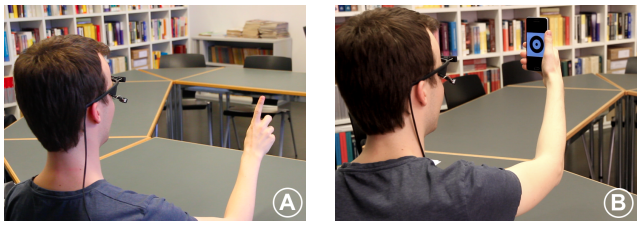

Figure 4: The two conditions compared in the user evaluation. (A) Finger calibration (B) Marker calibration with the marker displayed on a mobile phone's screen.

We designed a pilot user study and gathered quantitative and qualitative data. Twelve adult subjects aged between 23 and 58 years old (M = 35.2, SD = 13.6, 8 male and 4 female) took part in the evaluation. Nine participants were wearing vision correcting glasses. Five users had previously been exposed to eye tracking, however, only two of them had tried to calibrate an eye tracker before. Most of the participants (9 out of 12) have a technical background. Each subject was asked to perform two tasks. The first task was to calibrate an eye tracker with their finger, followed by a separate accuracy test. The second task was to use the standard manual marker calibration, again followed by an accuracy test. Users were

informed to collect samples from 9 locations covering their field of view in a grid-like pattern (similar to a 9-point on-screen calibration). There was no information given on how to spread out the points or how to hold the finger/smartphone. Each calibration session was performed twice. The presentation order of the conditions was counterbalanced. After each task, each user was asked to complete a System Usability Scale (SUS) [Brooke 1996] questionnaire. After completing both tasks, participants had to fill in an additional form which compared the two conditions. The duration of the experiment was between 15 and 20 mins per participant.

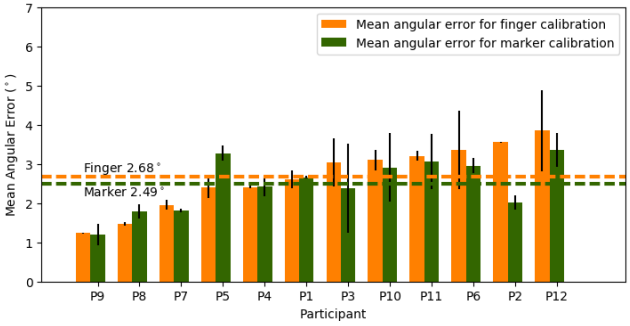

Figure 5: The mean angular error per participant, sorted by finger calibration error. The error bars represent the SD.

Figure 5 shows the mean angular error per participant. On average, the finger calibration error ($M = 2.68°$, $SD = 0.820°$) is comparable to the marker calibration error ($M = 2.49°$, $SD = 0.673°$). A student's t-test reveals that the difference between them is not statistically significant ($t(22) = 0.640$, $p = 0.53$, $Cohen's\ d = 0.253$).

In terms of usability, the average SUS score for finger calibration was 78.8 compared to 76.3 for marker calibration. The SUS does not offer a guideline on how to compare scores, however, given its wide use, the average score is considered to be 68. A student's t-test shows that the difference between finger ($M = 78.75$, $SD = 16.93$) and marker calibration ($M = 76.25$, $SD = 16.63$) is not statistically significant ($t(22) = 0.365$, $p = 0.719$, $Cohen's\ d = 0.149$).

Qualitative feedback. At the end of the experiment, participants had to fill in a questionnaire in which the two calibration methods were compared in terms of preference, perceived speed, ease of understanding, and the feeling of being in control. In terms of perceived speed, half of the participants saw no difference between the two methods, while 4 of them said that the finger calibration seemed faster. 9 out of 12 participants found the two experiments equally easy to understand, which was in line with our expectation. In terms of control, 7 participants said that the finger gave them a better sense of being in control. Overall, 7 out of 12 participants expressed that they preferred the finger over the marker, 4 were in favor of the marker, and 1 had no preference.

4.3 Hand segmentation

In this experiment, we quantitatively evaluated the hand segmentation on the EgoHands dataset [Bambach et al. 2015]. The dataset contains 48 videos taken with a Google Glass from an egocentric perspective, similar to a head-mounted eye tracker. It focuses on the interaction between two people and contains 4800 manually annotated frames with pixel-level masks for hands present in the scene (up to four segmentation masks, one for each person's hand).

Our algorithm was designed to identify one segmentation mask, the one with the largest contour. In 4177 images (87%), our segmentation overlapped with one of the manually segmented hands. For such cases, the overlap in pixels was, on average, 60%. For 61 images (1.3%), our method correctly identified that no annotated hands were present in the scene. For the remaining 562 images (11.7%) our algorithm failed to detect any. These results show that the proposed method is simple and good enough for this application.

4.4 Runtime analysis

We evaluated the runtime of the finger detection algorithm on a laptop with an i5 CPU @ 2.3 GHz. Small images ($320 \times 240\ px$) need ~7.7 ms (~130 FPS), medium images ($1280 \times 720\ px$) need ~18 ms (~55 FPS), and large images ($1920 \times 1080\ px$) need ~33 ms (~30 FPS). This shows that our method is fast enough to accommodate the Pupil's scene camera frame rate.

5 DISCUSSION

Parallax Effect. The proposed method collects calibration samples at about an arm's length away from the user. If the distance to the point of regard is different than the calibration distance, there will be a parallax error. This happens due to the position of the camera relative to the eye. This type of error is found on most video-based monocular mobile gaze trackers and can also influence the results of our method. [Mardanbegi and Hansen 2012] studied this effect for different calibration and fixation distances. Methods like CalibMe [Santini et al. 2017] allow calibrating at varying distances but, to minimize the parallax error, require recalibration for every target fixation plane. Our method is well suited for mobile scenarios, but the calibration distance is limited. The best results are obtained when the interaction happens closer to the user (e.g., in interaction scenarios where the objects are within the user's reach).

Area covered during calibration. In the user evaluation, for both the calibration and the accuracy test, participants had to sample 9 locations within their field of view in a grid-like pattern. To simulate a realistic scenario, they were not told how or where to place their finger/smartphone. An analysis of these locations has shown that points which fall outside of the calibration area will have larger errors due to extrapolation.

6 CONCLUSION AND FUTURE WORK

We presented a novel calibration technique which simplifies the initial calibration step for head-mounted eye trackers. Users collect calibration samples by pointing with their fingers and fixating the fingertip. No additional assistance or specialized calibration markers are necessary, thus enabling quick initialization for pervasive and mobile eye tracking in any environment. The proposed method achieves comparable accuracy to similar marker-based calibration. Our preliminary user evaluation highlighted that the majority of the participants preferred finger calibration over traditional markers. In the future, we plan to investigate and incorporate ways to mitigate the parallax effect in situations where the fixation plane is further away from the user.

REFERENCES

Mihai Bâce, Teemu Leppänen, David Gil de Gomez, and Argenis Ramirez Gomez. 2016. ubiGaze: Ubiquitous Augmented Reality Messaging Using Gaze Gestures. In *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications (SA '16)*. ACM, New York, NY, USA, Article 11, 5 pages. https://doi.org/10.1145/2999508.2999530

Sven Bambach, Stefan Lee, David J. Crandall, and Chen Yu. 2015. Lending A Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions. In *The IEEE International Conference on Computer Vision (ICCV)*.

John Brooke. 1996. SUS: A "quick and dirty" usability scale. In *Usability Evaluation in Industry*, Ian Lyall McClelland Bernard Weerdmeester Patrick W. Jordan, B. Thomas (Ed.). Taylor & Francis, London.

Andreas Bulling and Hans Gellersen. 2010. Toward Mobile Eye-Based Human-Computer Interaction. *IEEE Pervasive Computing* 9, 4 (October 2010), 8–12. https://doi.org/10.1109/MPRV.2010.86

Feridun M. Celebi, Elizabeth S. Kim, Quan Wang, Carla A. Wall, and Frederick Shic. 2014. A Smooth Pursuit Calibration Technique. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '14)*. ACM, New York, NY, USA, 377–378. https://doi.org/10.1145/2578153.2583042

Andrew T. Duchowski. 2007. *Eye Tracking Methodology: Theory and Practice*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Michael Xuelin Huang, Tiffany C.K. Kwok, Grace Ngai, Stephen C.F. Chan, and Hong Va Leong. 2016. Building a Personalized, Auto-Calibrating Eye Tracker from User Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5169–5179. https://doi.org/10.1145/2858036.2858404

Pawel Kasprowski and Katarzyna Harezlak. 2016. Implicit Calibration Using Predicted Gaze Targets. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA '16)*. ACM, New York, NY, USA, 245–248. https://doi.org/10.1145/2857491.2857511

Pawel Kasprowski, Katarzyna Harężlak, and Mateusz Stasch. 2014. Guidelines for the Eye Tracker Calibration Using Points of Regard. In *Information Technologies in Biomedicine, Volume 4*, Ewa Piętka, Jacek Kawa, and Wojciech Wieclawek (Eds.). Springer International Publishing, Cham, 225–236.

Moritz Kassner, William Patera, and Andreas Bulling. 2014. 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 (UbiComp '14 Adjunct)*. ACM, New York, NY, USA, 1151–1160. https://doi.org/10.1145/2638728.2641695

Christian Lander, Frederic Kerber, Thorsten Rauber, and Antonio Krüger. 2016. A Time-efficient Re-calibration Algorithm for Improved Long-term Accuracy of Head-worn Eye Trackers. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA '16)*. ACM, New York, NY, USA, 213–216. https://doi.org/10.1145/2857491.2857513

Päivi Majaranta and Andreas Bulling. 2014. *Eye Tracking and Eye-Based Human–Computer Interaction*. Springer London, London, 39–65. https://doi.org/10.1007/978-1-4471-6392-3_3

Diako Mardanbegi and Dan Witzner Hansen. 2012. Parallax Error in the Monocular Head-mounted Eye Trackers. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12)*. ACM, New York, NY, USA, 689–694. https://doi.org/10.1145/2370216.2370366

Ken Pfeuffer, Melodie Vidal, Jayson Turner, Andreas Bulling, and Hans Gellersen. 2013. Pursuit Calibration: Making Gaze Calibration Less Tedious and More Flexible. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 261–270. https://doi.org/10.1145/2501988.2501998

Thiago Santini, Wolfgang Fuhl, and Enkelejda Kasneci. 2017. 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 (CHI '17)*. ACM, New York, NY, USA, 2594–2605. https://doi.org/10.1145/3025453.3025950

Yusuke Sugano and Andreas Bulling. 2015. Self-Calibrating Head-Mounted Eye Trackers Using Egocentric Visual Saliency. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. ACM, New York, NY, USA, 363–372. https://doi.org/10.1145/2807442.2807445