3D gaze estimation in the scene volume with a head-mounted eye tracker

Carlos Elmadjian
Computer Science Department
University of São Paulo
elmad@ime.usp.br

Pushkar Shukla
Computer Science Department
University of California, Santa Barbara
pushkarshukla@umail.ucsb.edu

Antonio Diaz Tula
Computer Science Department
University of São Paulo
diaztula@ime.usp.br

Carlos H. Morimoto
Computer Science Department
University of São Paulo
hitoshi@ime.usp.br

ABSTRACT

Most applications involving gaze-based interaction are supported by estimation techniques that find a mapping between gaze data and corresponding targets on a 2D surface. However, in Virtual and Augmented Reality (AR) environments, interaction occurs mostly in a volumetric space, which poses a challenge to such techniques. Accurate point-of-regard (PoR) estimation, in particular, is of great importance to AR applications, since most known setups are prone to parallax error and target ambiguity. In this work, we expose the limitations of widely used techniques for PoR estimation in 3D and propose a new calibration procedure using an uncalibrated head-mounted binocular eye tracker coupled with an RGB-D camera to track 3D gaze within the scene volume. We conducted a study to evaluate our setup with real-world data using a geometric and an appearance-based method. Our results show that accurate estimation in this setting still is a challenge, though some gaze-based interaction techniques in 3D should be possible.

CCS CONCEPTS

• **Human-centered computing** → **Interaction devices**; • **Computing methodologies** → *Machine learning approaches*;

KEYWORDS

Head-mounted eye tracking, calibration, gaze estimation, 3D dataset

ACM Reference Format:
Carlos Elmadjian, Pushkar Shukla, Antonio Diaz Tula, and Carlos H. Morimoto. 2018. 3D gaze estimation in the scene volume with a head-mounted eye tracker. In *COGAIN '18: Workshop on Communication by Gaze Interaction, June 14–17, 2018, Warsaw, Poland*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3206343.3206351

1 INTRODUCTION

Gaze-based interfaces are part of an established way in itself of human-computer interaction. They have been employed in specific scenarios where traditional mechanisms of manual input are not possible due to user or context constraints. More recently, with the growth of virtual (VR) and augmented reality (AR) applications, gaze-based techniques have been considered as a means to refine and improve 3D interaction in such domains.

Head-mounted eye trackers are designed to be wearable, which means that a calibration procedure between the eyes and a front-facing camera is required in order to accurately estimate the user's gaze. This procedure generally involves a mapping between gaze data and targets on a planar surface, with the input either being image features (such as the pupil projected position) or a 3D model of the eye, which is used to determine the optical axis [Hansen and Ji 2010]. However, since interaction in VR and AR domains occurs in a 3D volume, traditional estimation approaches might be limiting in the sense that they cannot accurately predict the user's point-of-regard (PoR) in 3D without taking vergence into account.

This issue is particularly evident when there is a partial occlusion or collinearity of 3D objects in the user's line of sight, as shown in Figure 1. Most known approaches in this case rely on determining the fixation target by computing the intersection of gaze direction with scene objects. This inevitably leads to a biased outcome, as the nearest objects in the line of sight will tend to be hit more often, as shown in Figure 2. One way of overcoming this issue is through "parallax tricks", such as requiring the user to fixate at the desired object from different perspectives.

However, we argue that establishing a way to perform gaze estimation in the whole scene frustum would be a more interesting course of action, as it would allow for gaze-contingent volumetric interfaces, which could increase the level of control and visualization of current 3D interfaces.

Although gaze estimation in scene volumes has barely been explored in the literature, there have been some attempts to solve this problem imposing several constraints. In the case of remote

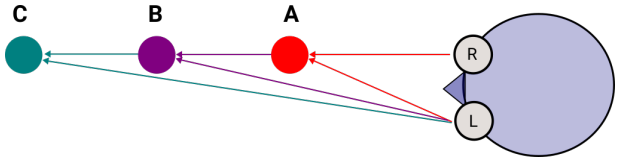

Figure 1: When a set of objects (A, B, C) are relatively collinear with the visual axis of one of the eyes (R), the other one (L) is necessary to determine gaze depth.

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

ACM ISBN 978-1-4503-5790-6/18/06…$15.00
https://doi.org/10.1145/3206343.3206351

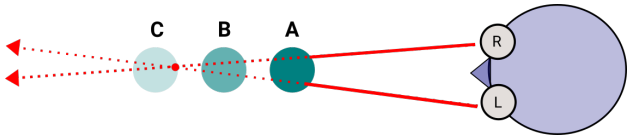

Figure 2: Ray-object intersection tends to create a bias in favor of closer objects (A), instead of farther or actual objects (C) being gazed by the user.

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

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

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

- a calibration procedure for the scene volume;
- a dataset for gaze estimation in the scene volume;
- new estimation techniques for this setting.

Additionally, we also explore other minor problems, such as estimating the eyeball position through angular disparities between different camera coordinate systems, and we provide a thorough discussion about the advantages and shortcomings of each technique in this scenario.

2 PREVIOUS WORKS

One of the first works to explore the use of gaze in 3D environments was provided by [Tanriverdi and Jacob 2000]. Others followed, with the increasing number of applications related to virtual and augmented reality.

There are numerous known techniques for PoR estimation in 2D surfaces [Hansen and Ji 2010], but only a few aiming 3D environments. Yet, many of these latter techniques are limited to gaze direction, with no accurate report depth information. Datasets for this purpose are also lacking, since there has been only one work that considered scene depth knowledge, but from the perspective of a remote eye tracker [Mora and Odobez 2014].

Mardanbegi and Hansen developed a method that enables the user to interact with planar displays in a 3D environment using a head-mounted eye tracker [Mardanbegi and Hansen 2011]. This method partially resorts to known calibration methods, but it also assumes there is a homographic mapping between the screen on the scene camera image to the actual screen coordinates due to planarity constraints. A further development, using image features and less restrictions, was presented by [Lander et al. 2015].

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

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

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

Essig et al. presented a feature-based approach that relied only on estimates generated by a neural network [Essig et al. 2006]. Measuring binocular gaze angles, their reported results showed significant improvement in comparison to a geometrical solution, specially regarding depth, but in a very controlled environment. More recently, [Itoh and Klinker 2014] developed a technique to estimate gaze for HMDs using the Świrski and Dodgson algorithm [Swirski and Dodgson 2013], which computes the optical axis by assuming the eyeballs as perfect spheres. Others focused on the nature of gaze depth and its estimation [Duchowski et al. 2014, 2011; Lee et al. 2017]. A general theory for 3D PoR estimation was provided by [Pirri et al. 2011].

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

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

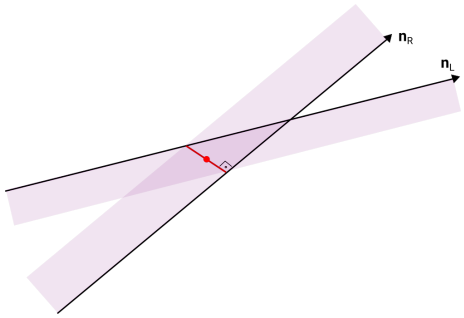

Figure 3: Since two estimated vectors in 3D space coming from the right (n_R) and left (n_L) eyes will most likely not intersect, the midpoint of the shortest segment between gaze rays (in red) is a common measure of gaze estimation for geometric-based models.

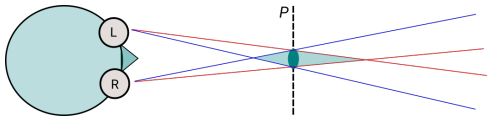

Figure 4: The angular error (green ellipse) of gaze estimation barely affects the correct positioning on the facing plane (P), but it impacts greatly in terms of depth error, as it can be seen by the largest axis of the diamond-shaped area.

[McMurrough et al. 2012] and [Paletta et al. 2013] have both used a head-mounted setup with an RGB-D egocentric camera, but they limited themselves to perform only a 2D calibration step for posterior analysis of gaze data overlaid in depth images, thus incurring on ambiguities associated with the lack of calibration to the scene volume.

3 GAZE ESTIMATION MODELS

Gaze estimation techniques can be classified either as geometric-based or appearance-based [Hansen and Ji 2010]. In general, geometric models are known to be robust to eye tracker slippage and are also able to compensate the parallax error, making them a suitable choice for head-mounted eye trackers. Also, since geometric models provide gaze vectors through a rigid transformation of the eyeballs, they require in theory just one screen target to perform a user calibration, though in practice more points are necessary to improve accuracy. In fact, geometric models are known to perform worse than appearance-based ones due to several simplifications, such as assuming that the eyeball is a perfect sphere.

Appearance-based models, on the other hand, rely on tracking specific features on the eye image, such as the projected pupil center. Since the eyeball rotation occurs in 3D space and the features are captured through their projection, displacement of attributes such as the pupil centers will not be linear on the image plane. Thus, a nonlinear regression function capable of mapping tracked features to targets on the scene camera must be found, which can be done through several techniques, such as polynomial fitting, support vector regression, Gaussian processes, and artificial neural networks. Generally, this procedure tends to yield more accurate results than a purely geometric model, as the regressor learns more intrinsic information about the input data, including noise and sensing biases. In practice, this means that a large and representative set of points should be chosen from the target surface to achieve a good calibration.

Figure 5: The Pupil binocular eye tracker coupled with an Intel Realsense R200 camera used as head-mounted setup.

In 3D, geometric models are a natural choice for gaze estimation, as one can assume that fixation in space occurs when there is an intersection between both vectors aligned with the visual axis. Because the two estimated vectors will most likely not intersect in 3D, a reasonable substitute for the required intersection could be the midpoint of the shortest segment separating these two lines [Hennessey and Lawrence 2009], as shown in Figure 3. Still, absence of high-resolution cameras and simplifications in the 3D model of the eyeball may account for errors that compromise in a significant way gaze depth estimation, as shown in Figure 4. In this study we investigate how well a simplified geometric model can be calibrated to the scene camera frustum in contrast with an appearance-based approach, given a set o targets in space covering the user's field-of-view (FoV).

4 METHODOLOGY

4.1 Architecture description

A binocular head-mounted eye tracker from Pupil Labs was used to collect gaze data at 30 Hz with a resolution of 480p. The Intel Realsense R200 RGB-D camera was adapted to the tracker frame as the scene camera. The R200 device was configured to run at 60 Hz and capture one RGB image and one depth image from scene at each frame at 480p. All these devices were connected to a laptop PC in order process and record the streams. The head-mounted setup is shown in Figure 5.

The software used to compute eye features, such as the projected pupil centers and 3D gaze vectors, was a modified version of the one provided by Pupil Labs (v0913) that also allowed us to record eye streams. We developed our own software, using *OpenCV* and *librealsense* libraries, to detect markers in the frustum, identify them, and report information about their position in 3D space. The technique used for marker detection resorted to a similar approach

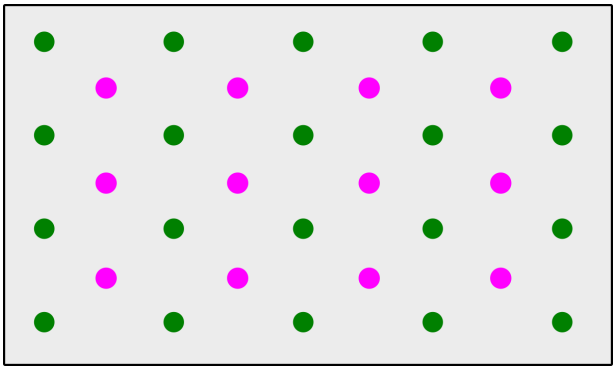

Figure 6: Disposition of training (green) and testing (pink) targets per plane during calibration procedure.

proposed by [Fiala 2005]. Once detected, the coordinates at the center of the marker were projected to 3D using a mapping procedure between the RGB and depth frames. A mean filter was applied over a window of 5 frames to smooth depth noise.

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

4.2 Calibration procedure

We designed our calibration procedure to be conservative about the number of targets that should be employed in the process, as there was no previous information about how the camera frustum should be spatially sampled. Therefore, we decided to use five planes at different depths from the user. Each plane had a grid of 5×4 binary AR markers for training purposes and a 4×3 internal grid used for testing. The relative position of the grids is shown in Figure 6. The size of the planes and their markers also changed in respect to depth in order to fill up the scene camera FoV, maintaining the same angular ratio between them.

The planes were defined by positioning the user at 5 different depths from a projection wall: 0.75 m, 1.25 m, 1.75 m, 2.25 m, and 2.75 m. These distances were considered taking into account some limitations associated with the R200 camera, as its sensing range varies from 0.51 m to 4 m, according to the manufacturer, but our empirical evaluation revealed that the camera was only able to provide reliable data between 0.7 m and 3 m.

The procedure starts by placing the user at 2.75 m from the wall and adjusting the projection center so it can be aligned with the camera scene FoV center. After that, we show all the training markers on the wall and the user is asked to follow a green target that remains static on the center of each marker at a time, while the system gathers 30 synchronized samples from each different camera feed. Samples are only recorded if eye features and markers are properly recognized by detection algorithms. During this stage, participants get a chance to practice and are instructed to move their eyes to the correct target prior to triggering recording.

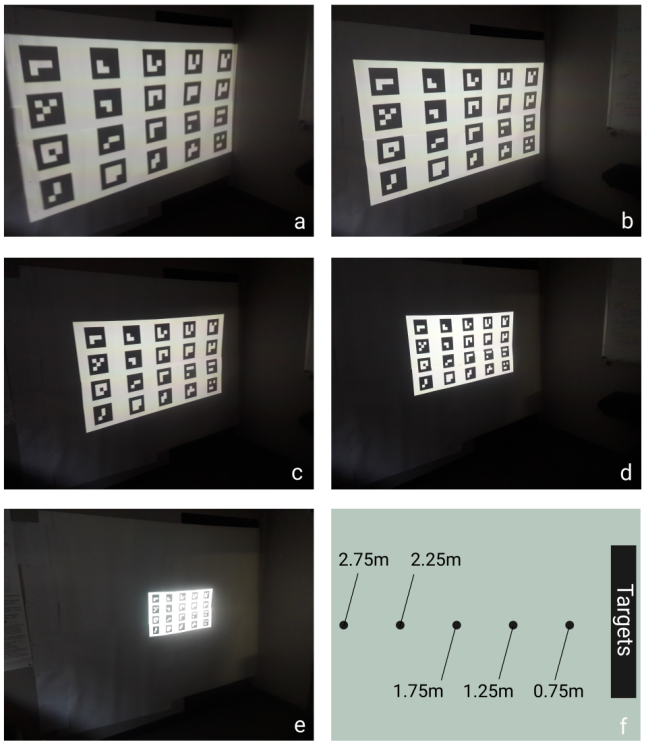

Figure 7: Projection of the targets used in calibration in respect to depth, from 2.75 m to 0.75 m (Figs. a-e). Figure f illustrates the 5 different user positionings in this setup.

Once acquisition of training targets for a plane is complete, the user is asked to remain still and repeat the same procedure now for the testing targets, which are shown in a similar fashion. Following that, we move to a next depth plane by placing the user closer to the projection wall and we repeat the previous routine of centering the screen and showing the markers for data acquisition. An opportunity to redo part of the procedure is offered whenever the individual notices a mistake. Figure 8 shows a diagram that summarizes the whole method, while Figure 7 depicts it.

4.3 Estimation approaches

One of the goals of this study was to assess the performance of geometric and appearance-based calibration methods for the camera frustum. Following, we describe the estimation techniques that were employed in this work and we detail the modifications that were adopted to address some of the peculiarities of estimation in the scene volume.

4.3.1 Geometric model. Our geometric model is constituted by two normal vectors to the pupil centers provided by the Pupil tracking software. These vectors are a result of a 3D eyeball model that is built from multiple observations of projected pupil contours, which are approximated to ellipses. Assuming a camera pinhole model and a weak perspective projection, the centers of these ellipses are considered to be part of a sphere surface, which is regarded as a

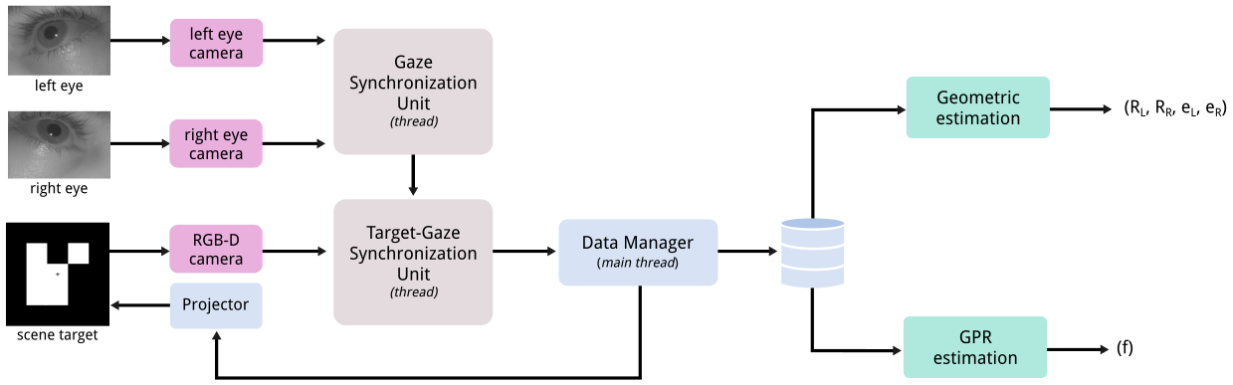

Figure 8: Architecture of the calibration procedure. A *Data Manager* routine controls the experiment, requesting the projector to show training or testing targets and recording synchronized data from scene and eye cameras. Recorded data is used later for gaze estimation algorithms. While GPR goal is to determine a function f, the geometric procedure needs to find the position of eyeballs (e_L and e_R) and associated rotation matrices (R_L and R_R).

rough approximation of the eyeball. A detailed explanation of this method can be found in [Swirski and Dodgson 2013].

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

With no constraints and prior knowledge about the user anatomy, finding the eyeball position in the scene camera space can be treated as an optimization problem. [Mansouryar et al. 2016] proposed a technique that minimizes the angular disparity between computed gaze vectors on eye camera space and targets on scene camera space. However, this approach is computationally expensive, as the search involves determining 6 parameters simultaneously (3 Euler angles and the coordinates for the eyeball position). Additionally, dependency between parameters might not always yield the optimal outcome, as this is a typical non-convex problem, with many local minima.

To reduce the complexity of the search space, we propose an approach that breaks the optimization procedure into two steps: first we compute the eyeball position and then we determine the rotation that places its gaze vectors into the scene camera space. This computation assumes that angular disparity patterns are roughly preserved among gaze vectors between eye and scene camera spaces. Therefore, given a set of angles between gaze vectors in the eye camera space and a set of targets sampled in the scene camera frustum, the eyeball position is determined by minimizing the squared disparities between associated angles in both coordinate systems, as shown in Equation 1.

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

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

Once a reasonable estimate is determined for both eyeballs, we proceed to compute the rotation that transforms the gaze vector in eye camera space to the appropriate orientation and position in the scene camera coordinate system. This is summarized by the parametric equation (2), where λ is a free parameter. Again, this is an optimization problem where we want to compute the rotation matrix (R) and the translation (T) that minimize the cosine distance between transformed gaze vectors n_i and corresponding normalized vectors with origin in e and pointing toward the target t_i. This can be expressed by (3). During non-linear minimization iterations, the parameter β can be used to penalize larger dissimilarities, speeding up convergence, if achievable.

$$e_{cam} + T + \lambda R\mathbf{n} \tag{2}$$

$$f(R) = \sum_{i=1}^{N} \left(1 - Rn_i \cdot \frac{t_i - e}{||t_i - e||} \right)^{\beta} \tag{3}$$

Finally, the PoR in 3D is computed as the midpoint of the shortest segment between both rotated gaze rays n_l and n_r, with respective origins in e_l and e_r. Assuming that this segment r is perpendicular to both rays and given the parametric equations of each ray, we solve for λ_l and λ_r in order to determine the midpoint m of this shortest segment, as shown in Equations 4, 5 6, and 7. A diagram illustrating the geometric estimation pipeline is shown in Figure 9.

$$r = e_l - e_r \tag{4}$$

$$\lambda_l = \frac{(n_l \cdot n_r)(n_r \cdot r) - (n_l \cdot r)(n_r \cdot n_r)}{(n_l \cdot n_l)(n_r \cdot n_r) - (n_l \cdot n_r)(n_l \cdot n_r)} \tag{5}$$

$$\lambda_r = \frac{(n_l \cdot n_l)(n_r \cdot r) - (n_l \cdot r)(n_l \cdot n_r)}{(n_l \cdot n_l)(n_r \cdot n_r) - (n_l \cdot n_r)(n_l \cdot n_r)} \tag{6}$$

$$m = \frac{(e_l + \lambda_l n_l + \lambda_r n_r + e_r)}{2} \tag{7}$$

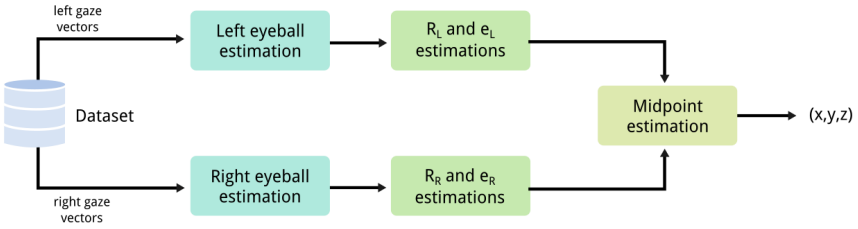

Figure 9: Pipeline of the geometric method.

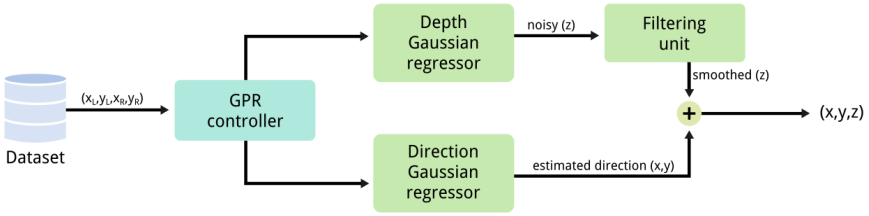

Figure 10: Pipeline of the Gaussian processes regressor method.

4.3.2 Regression-based model. Despite the myriad of appearance-based models that have been proposed in the literature, we opted for a Gaussian processes regression (GPR) due to its wide applicability and reportedly good results with similar problems [Sesma-Sanchez et al. 2016; Sugano et al. 2013].

A Gaussian process is a stochastic process that assumes that every finite set of its random variables has a linear combination with a normal distribution. This strong assumption makes them surprisingly flexible as a tool to model different sets of problems [Rasmussen and Williams 2006]. In a way, they remind of Support Vector Machines, in the sense that they are kernel machines. Thus, Gaussian processes are primarily governed by their covariance function, as it encodes the similarity between data input and targets.

For the purpose of finding a regression between gaze data and scene targets, we selected a Squared Exponential Kernel, which is shown in Equation 8, with initial parameters $\sigma = 1.5$ and scale factor $l = 1.0$, as this configuration has demonstrated better generalization properties during our preliminary trials.

$$K(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right) \tag{8}$$

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

Considering these limitations, we approached the problem of depth regression separately, i.e., building a Gaussian processes regressor for gaze depth and another one for gaze direction in the projected scene camera plane. Sequential observations of z-values were used as a filtering step to improve prediction. Results of both regressors were combined later to perform gaze estimation. A diagram illustrating the process is shown in Figure 10.

5 DATASET DESCRIPTION

We collected gaze data from 11 subjects (5 women) with ages ranging between 22 to 35 years. All of them had normal or corrected-to-normal vision during the procedure, which followed the protocol fully described in section 4.2. The dataset comprehend information about both left and right normalized pupil centers in each image (*LE2D*, *RE2D*), as well as both normalized gaze vectors acquired through 3D eyeball modeling (*LE3D*, *RE3D*), and the ground truth targets in 3D coordinates provided by the RGB-D camera (*RS3D*). During this process, we also acquired grayscale frames from both eyes and scene cameras for debugging purposes, although this information was not integrated into the dataset due to size limitations. Therefore, a valid sample was defined as:

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

The number of samples was fixated on 30 by target, which accounts for roughly 1 second of observation with our current architecture. This value was chosen considering subject's likely extenuation due to a large amount of targets in the experiment. Additionally, in order to minimize individual errors and increase comfort during data acquisition, each participant received a device to activate the moment of recording at each target being gazed. Table 1 summarizes information about the number of samples for training and testing targets. The dataset is publicly available at: *https://github.com/elmadjian2/3D_gaze_dataset*.

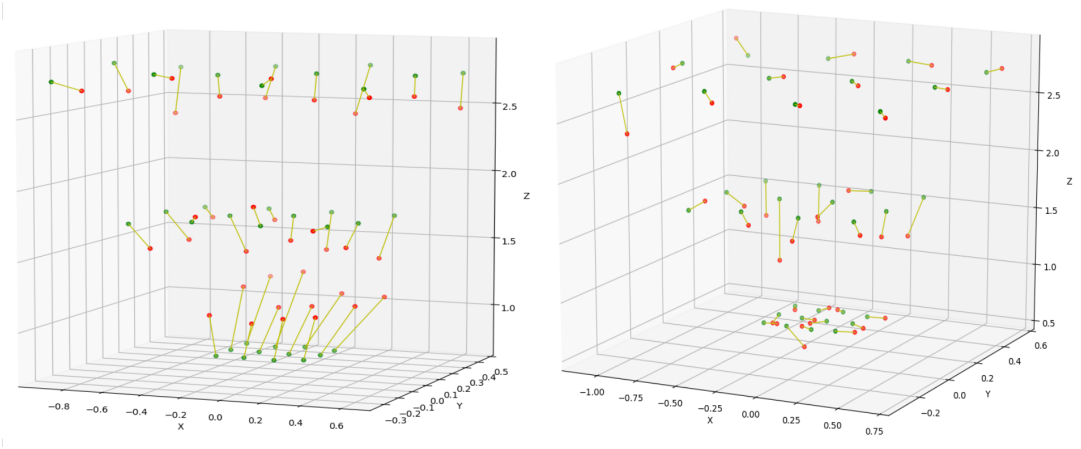

Figure 11: Top-down visualization of the gaze estimation from a participant using the geometric approach (left) and the regression-based one (right). Green points indicate the ground truth, while red dots are the corresponding estimates.

Table 1: Summary of the number of training and testing targets collected per user.

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

Table 2: Summary of the results for the three metrics using 5 planes for calibration.

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

6 EVALUATION AND RESULTS

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

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

We evaluated accuracy in terms of depth error, angular error and Euclidean distance to ground truth. Figure 12 summarizes the basic statistical results in respect to each metric by plane using all the 5 planes for prediction. These results are compiled in Table 2. Figure 13 portrays the impact of the number of training planes for gaze estimation. Since there was no significant difference on using 3 to 5 planes for training, we report only the analysis of variance (ANOVA) results for the latter.

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

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

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

7 DISCUSSION

Our results show that gaze estimation within a scene frustum is still an open challenge, specially in terms of depth. A remarkable observation is that the number of calibration planes seems to have a noticeable impact on estimation. Therefore, even through a geometric approach, a single-plane calibration does not suffice to provide useful gaze estimation in the scene volume.

It was also noted that the closer plane (0.75 m) showed higher angular errors, particularly with the appearance-based method. This might contradict the expectation that the farther plane (2.75 m) should yield worse results. Yet, one possible explanation for this outcome could be the effect of vergence angular disparity on planes, which is only strong in the first one (0.75 m).

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

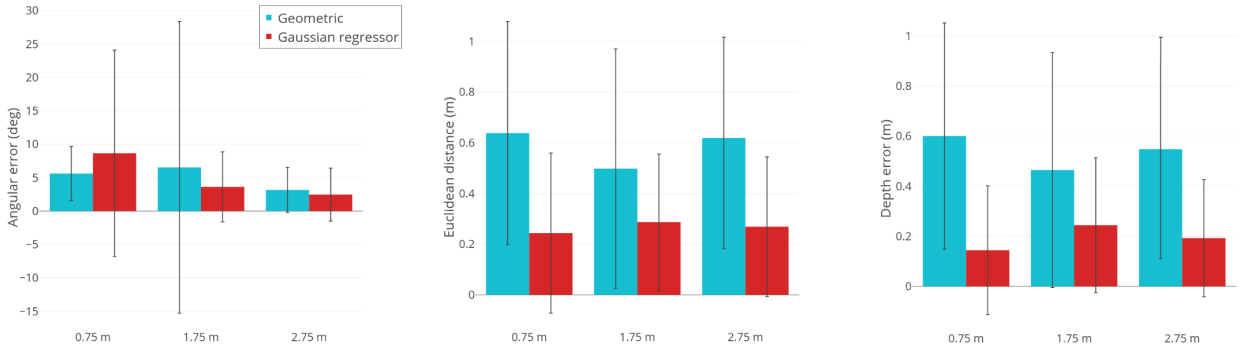

Figure 12: Average angular error, Euclidean distance, and depth error for each testing plane separately.

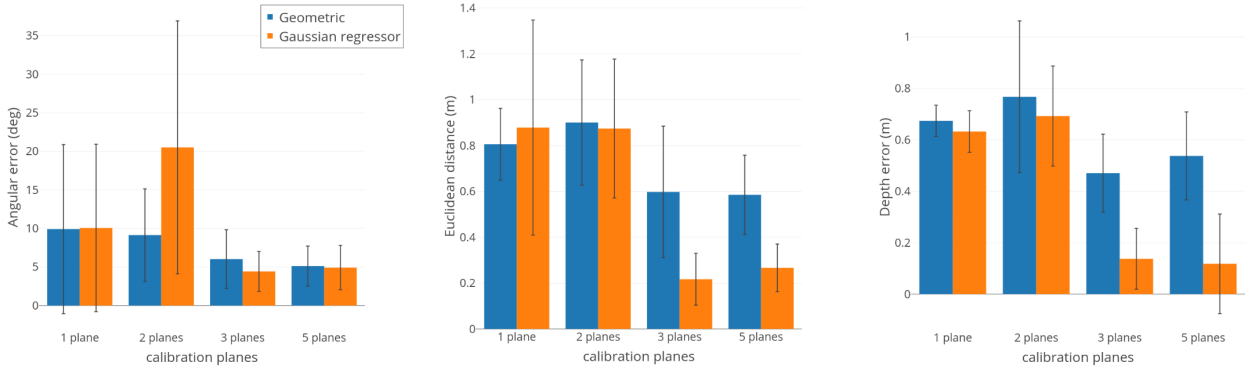

Figure 13: Average angular error, Euclidean distance, and depth error in respect to the number of planes used for calibration.

amount of noise. However, a larger number of participants should be considered in future studies in order to confirm or generalize all these findings.

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

That said, both methods considered for this study presented different strengths and weaknesses. The geometric model showed a tendency of preserving the spatial relationship between testing targets, although inaccurate estimations of the rotation matrix and the eyeball position clearly compromised the whole system, as gaze estimates tend to appear shifted by a certain degree from the user perspective. The appearance-based method does not suffer from this problem, but estimates seem to be more random. Figure 11 illustrates these phenomena.

Although the Gaussian processes regression might have shown improved results in two metrics, it should be noted that it relies on too many targets for training in order to provide a suitable regression, whereas the geometric model, at least in theory, could be already functional with much fewer targets. This over-reliance on the number of samples by the appearance-based approach can easily be observed in Figure 13.

Finally, although the results regarding angular error are still not comparable to the eye-tracking industry standards, it is possible to devise some uses for the proposed techniques in 3D AR scenarios, such as allowing the user to have different contextual interfaces based on gaze depth, or triggering access to detailed information about scene objects by switching back and forth between the environment and the HMD. With improvement on estimation accuracy, vergence-based controls for 3D interaction could also be feasible.

8 CONCLUSION

To allow for a compelling gaze interaction in 3D environments, it is essential to get a good estimate of the user's PoR in the scene, but gaze depth is still the main challenge to overcome for accurate estimation in the scene volume. Our results suggest that further investigation should be done in order to determine the effect of target positioning in regard to depth on the accuracy of the proposed estimation methods, even for 3D interaction with non-continuous depth. Finally, we expect that the proposed dataset as well as the techniques and procedures presented here might instigate further research on this topic.

ACKNOWLEDGMENTS

We would like to thank Andrew T. Kurauchi and acknowledge partial support of this work by the São Paulo Research Foundation (FAPESP), grants 2016/10148-3 and 2017/06933-0.

REFERENCES

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

Andrew T. Duchowski, Donald H. House, Jordan Gestring, Robert Congdon, Lech Swirski, Neil A. Dodgson, Krzysztof Krejtz, and Izabela Krejtz. 2014. Comparing estimated gaze depth in virtual and physical environments. In *Eye Tracking Research and Applications, ETRA '14, Safety Harbor, FL, USA, March 26-28, 2014*. 103–110. https://doi.org/10.1145/2578153.2578168

Andrew T. Duchowski, Brandon Pelfrey, Donald H. House, and Rui I. Wang. 2011. Measuring gaze depth with an eye tracker during stereoscopic display. In *Proceedings of the 8th Symposium on Applied Perception in Graphics and Visualization, APGV 2011, Toulouse, France, August 27-28, 2011*. 15–22. https://doi.org/10.1145/2077451.2077454

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

Mark Fiala. 2005. ARTag, a Fiducial Marker System Using Digital Techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*. 590–596. https://doi.org/10.1109/CVPR.2005.74

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

Dan Witzner Hansen and Qiang Ji. 2010. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 3 (2010), 478–500. https://doi.org/10.1109/TPAMI.2009.30

Craig Hennessey and Peter D. Lawrence. 2009. Noncontact Binocular Eye-Gaze Tracking for Point-of-Gaze Estimation in Three Dimensions. *IEEE Trans. Biomed. Engineering* 56, 3 (2009), 790–799. https://doi.org/10.1109/TBME.2008.2005943

Yuta Itoh and Gudrun Klinker. 2014. Interaction-free calibration for optical see-through head-mounted displays based on 3D Eye localization. In *IEEE Symposium on 3D User Interfaces, 3DUI 2014, Minneapolis, MN, USA, March 29-30, 2014*. 75–82. https://doi.org/10.1109/3DUI.2014.6798846

Yong-Moo Kwon, Kyeong-Won Jeon, Jeongseok Ki, Qonita M. Shahab, Sangwoo Jo, and Sung-Kyu Kim. 2006. 3D Gaze Estimation and Interaction to Stereo Dispaly. *IJVR* 5, 3 (2006), 41–45. http://www.ijvr.org/sub/issues/issue3/16-1394-KIST-YMKWON-20061021.pdf

Christian Lander, Sven Gehring, Antonio Krüger, Sebastian Boring, and Andreas Bulling. 2015. GazeProjector: Accurate Gaze Estimation and Seamless Gaze Interaction Across Multiple Displays. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, UIST 2015, Charlotte, NC, USA, November 8-11, 2015*. 395–404. https://doi.org/10.1145/2807442.2807479

Youngho Lee, Choonsung Shin, Alexander Plopski, Yuta Itoh, Thammathip Piumsomboon, Arindam Dey, Gun A. Lee, Seungwon Kim, and Mark Billinghurst. 2017. Estimating Gaze Depth Using Multi-Layer Perceptron. In *2017 International Symposium on Ubiquitous Virtual Reality, ISUVR 2017, Nara, Japan, June 27-29, 2017*. 26–29. https://doi.org/10.1109/ISUVR.2017.13

Jianfeng Li and Shigang Li. 2014. Eye-Model-Based Gaze Estimation by RGB-D Camera. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2014, Columbus, OH, USA, June 23-28, 2014*. 606–610. https://doi.org/10.1109/CVPRW.2014.93

Morten Lidegaard, Dan Witzner Hansen, and Norbert Krüger. 2014. Head mounted device for point-of-gaze estimation in three dimensions. In *Eye Tracking Research and Applications, ETRA '14, Safety Harbor, FL, USA, March 26-28, 2014*. 83–86. https://doi.org/10.1145/2578153.2578163

Mohsen Mansouryar, Julian Steil, Yusuke Sugano, and Andreas Bulling. 2016. 3D gaze estimation from 2D pupil positions on monocular head-mounted eye trackers. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, ETRA 2016, Charleston, SC, USA, March 14-17, 2016*. 197–200. https://doi.org/10.1145/2857491.2857530

Diako Mardanbegi and Dan Witzner Hansen. 2011. Mobile gaze-based screen interaction in 3D environments. In *NGCA 2011, First Conference on Novel Gaze-Controlled Applications, Karlskrona, Sweden, May 26 - 27, 2011*. 2. https://doi.org/10.1145/1983302.1983304

Diako Mardanbegi and Dan Witzner Hansen. 2012. Parallax error in the monocular head-mounted eye trackers. In *The 2012 ACM Conference on Ubiquitous Computing, Ubicomp '12, Pittsburgh, PA, USA, September 5-8, 2012*. 689–694. https://doi.org/10.1145/2370216.2370366

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

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

Kenneth Alberto Funes Mora and Jean-Marc Odobez. 2014. Geometric Generative Gaze Estimation (G3E) for Remote RGB-D Cameras. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. 1773–1780. https://doi.org/10.1109/CVPR.2014.229

Susan M. Munn and Jeff B. Pelz. 2008. 3D point-of-regard, position and head orientation from a portable monocular video-based eye tracker. In *Proceedings of the Eye Tracking Research & Application Symposium, ETRA 2008, Savannah, Georgia, USA, March 26-28, 2008*. 181–188. https://doi.org/10.1145/1344471.1344517

Lucas Paletta, Katrin Santner, Gerald Fritz, Heinz Mayer, and Johann Schrammel. 2013. 3D attention: measurement of visual saliency using eye tracking glasses. In *2013 ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '13, Paris, France, April 27 - May 2, 2013, Extended Abstracts*. 199–204. https://doi.org/10.1145/2468356.2468393

Thies Pfeiffer and Patrick Renner. 2014. EyeSee3D: a low-cost approach for analyzing mobile 3D eye tracking data using computer vision and augmented reality technology. In *Eye Tracking Research and Applications, ETRA '14, Safety Harbor, FL, USA, March 26-28, 2014*. 195–202. https://doi.org/10.1145/2578153.2578183

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

Carl Edward Rasmussen and Christopher K. I. Williams. 2006. *Gaussian processes for machine learning*. MIT Press. http://www.worldcat.org/oclc/61285753

Laura Sesma-Sanchez, Yanxia Zhang, Andreas Bulling, and Hans Gellersen. 2016. Gaussian processes as an alternative to polynomial gaze estimation functions. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, ETRA 2016, Charleston, SC, USA, March 14-17, 2016*. 229–232. https://doi.org/10.1145/2857491.2857509

Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. 2013. Appearance-Based Gaze Estimation Using Visual Saliency. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 2 (2013), 329–341. https://doi.org/10.1109/TPAMI.2012.101

Lech Swirski and Neil Dodgson. 2013. A fully-automatic, temporal approach to single camera, glint-free 3d eye model fitting. *Proc. PETMEI* (2013).

Kentaro Takemura, Yuji Kohashi, Tsuyoshi Suenaga, Jun Takamatsu, and Tsukasa Ogasawara. 2010. Estimating 3D point-of-regard and visualizing gaze trajectories under natural head movements. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA 2010, Austin, Texas, USA, March 22-24, 2010*. 157–160. https://doi.org/10.1145/1743666.1743705

Vildan Tanriverdi and Robert J. K. Jacob. 2000. Interacting with eye movements in virtual environments. In *Proceedings of the CHI 2000 Conference on Human factors in computing systems, The Hague, The Netherlands, April 1-6, 2000*. 265–272. https://doi.org/10.1145/332040.332443

Xuehan Xiong, Qin Cai, Zicheng Liu, and Zhengyou Zhang. 2014. Eye gaze tracking using an RGBD camera: a comparison with a RGB solution. In *The 2014 ACM Conference on Ubiquitous Computing, UbiComp '14 Adjunct, Seattle, WA, USA - September 13 - 17, 2014*. 1113–1121. https://doi.org/10.1145/2638728.2641694