Tracking eye position and gaze direction in near-eye volumetric displays

Marek Wernikowski*
West Pomeranian University of Technology, Szczecin
University of Cambridge

Joseph G. March
University of Cambridge

Radosław Mantiuk †
West Pomeranian University of Technology, Szczecin

Ali Özgür Yöntem
University of Cambridge

Rafał K. Mantiuk ‡
University of Cambridge

Abstract

Near-eye volumetric displays, showing multiple focal planes, require knowledge of the accurate position of the nodal point of the eye to correctly render a 3D scene. This is because pixels seen through multiple planes must be accurately aligned with the eye's visual axis to ensure consistency across focal planes. While most eye-tracking methods focus on determining a gaze position within a designated target space, this work aims to track both the eye position and the corresponding gaze direction expressed in coordinates relative to the physical location of the volumetric display planes. To achieve this, we rely on a near-infra-red (NIR) camera image of the pupil and corneal reflections (glints). The existing eye model is used to establish the relationship between the pupil and glint positions in a NIR image and the eye position and rotation in a 3D space. We address the key challenge of robust tracking of the glints in a system that introduces multiple reflections. We also demonstrate that the system reduces the need for recalibration on subsequent uses. Our experiments on a multiple-focal plane display demonstrate that the method can maintain an accurate projection point for volumetric displays.

Index Terms: Eye tracking, gaze tracking, eye model, volumetric displays, multi-focal plane displays.

1 Introduction

Multi-focal-plane volumetric displays create an image in a 3D space by superimposing a stack of images at different depths [29]. The superposition can be achieved with beamsplitters [1, 20, 45], or with tuneable optics [4, 27]. Regardless of the solution, such a multi-plane display can provide "true" 3D capability, which alleviates the vergence and accommodation conflict and provides (near-) correct focus cues. Such volumetric displays are one of the more practical solutions for "true" 3D presentation as they, in general, require much fewer addressable pixels than holographic or light field displays. However, their main limitation is that they require knowledge of the eye position and, in particular, the position of the nodal point of each eye. This information is crucial for rendering a scene from the perspective of a virtual camera, which aligns corresponding pixels with the view axis of the human eye. To ensure such alignment, the existing multi-focal display prototypes resort to stabilizing the user's head with a bite-bar [1, 20], which makes the display very uncomfortable to use. The problem is also present in head-mounted displays, as those tend to move on the head (slippage) during use, therefore invalidating the calibrated eye position. In this paper, we show how the information about the nodal point position can be robustly tracked with a regular near-infra-red near-eye eye tracker, eliminating the need to fix the user's head position.

This work distinguishes between *eye-position tracking* and *gaze direction tracking*, which are both a form of *eye tracking*. Eye-position tracking refers specifically to the measurement of an observer's eye location. This is typically defined by the center points of the pupil and the nodal point of the eye optics, which collectively establish the eye's optical axis. In contrast, *gaze tracking* focuses on determining the *gaze direction*, which aligns with the eye's visual axis, or *gaze position* in the target space (e.g., screen coordinates). In this work we are predominantly interested in robust and accurate eye-position tracking, required for volumetric displays, but we also track the gaze-direction, which is useful for downstream applications.

Both gaze-direction and eye-position tracking are well-established problems and multiple solutions exist (which we review in Section 2). There have also been multiple attempts to introduce eye-position tracking into volumetric displays (reviewed in Section 2). However, none of these works provides sufficient details to reproduce and understand the performance of the system. This paper is meant to fill this gap and provide a practical recipe for implementing eye-position tracking within a near-eye volumetric display.

Our method requires a single (per eye) near-infra-red (NIR) eye tracker, such as those found in most VR headsets. Our prototype system is a part of a bench-mounted haploscope (see Section 4). It establishes a relationship between an eye-tracking camera image (pupil and NIR LED reflections) and the 3D eye parameters (position and rotation) using the eye model of Guestrin et al. [12]. The key challenge that we address in Section 5 is the robust tracking of the glints in a display system that introduces multiple reflections (from mirrors and beam splitters). Since model-based eye-position tracking requires accurate geometric calibration of the system, we explain in Section 6 how it can be done with high precision using computer vision techniques. Finally, in Section 7 we show how a generic eye-model can be calibrated for each individual, taking advantage of multiple focal planes available in our display system. Our results show our system can track the eye nodal point with the accuracy of 1 mm after one-time calibration. The error does not increase on subsequent uses (no re-calibration). Our gaze-direction tracking is robust to slippage and head movements and can track the gaze with the accuracy of 1.5°.

2 Background and Related Work

We review the main method used for tracking the eye and gaze position (Section 2), followed by a discussion of the works that integrate tracking into volumetric displays (Section 2).

Gaze tracking vs. eye position tracking. The predominant approach in contemporary video-based gaze tracking systems, known as "pupil minus corneal reflection" (P-CR) gaze tracking, involves tracking the pupil and corneal reflection, i.e., the reflection of light sources on the anterior surface of the cornea [41][15,

*e-mail: marek.wernikowski@zut.edu.pl
†e-mail: radoslaw.mantiuk@zut.edu.pl
‡e-mail: rafal.mantiuk@cl.cam.ac.uk

Sect.2.5.3]. The vector between the center of the pupil and the center of the reflection spot (i.e., the first Purkinje image called *glint*) changes with the rotation of the eye. This vector can be mapped from camera image coordinates to the gaze position in a specified target space, such as the pixel coordinates on a display plane. Typically, such mapping is modeled with a multivariate polynomial, which is fitted to the data collected in a calibration procedure in which the user is asked to gaze at several calibration points. It is worth noting that this calibration step calibrates for individual's intrinsic eye parameters, such as the cornea curvature (7.79 ± 0.27 mm [7]), refraction of the cornea surface and the effective refractive index of cornea and aqueous humor (ranging from 1.332 to 1.3375 [7]), or distance between the pupil center and the cornea curvature center [21, 40, 24, 25]. The main advantage of the P-CR tracking methods is that they are simple and relatively robust to the *slippage* — slight head movements in relation to the camera and light source, especially, if the glint is located near the cornea center [18]. This characteristic holds practical significance as minor head movements cannot be avoided in eye-tracking systems without resorting to impractical head constraints, such as a bite bar [30]. A broader analysis of the impact of slippage on the accuracy of the head-worn eye trackers is discussed in Niehorster et al. [26].

It is important to highlight that P-CR gaze tracking cannot determine the position of the eye nodal point (the focus of our work), as the mapping infers only the gaze position in a target space. Likewise, this eye-tracking method is inherently limited to measuring relative gaze directions within the calibration coordinate system. The gaze direction angles are unknown unless we establish the eye's physical position in relation to the display (the target space). We present an approach for addressing this issue in Section 7.3.

Accurately tracking eye position presents a significant challenge due to the necessity of considering both the anatomical structure of the eye and the individual variations in ocular component sizes. *Geometric methods* [13] address this challenge by employing a three-dimensional model of the eye to calculate pupil and/or eye nodal point location (see Section 5.2 for details). The common approaches, discussed in [32, 23, 31, 12, 38, 39, 37, 31, 14], use the multiple light sources and/or multiple cameras to track the relationship between features on the cornea surface and the elements of the tracking system (cameras, light sources). The known location of two light sources relative to the camera allows for obtaining head pose invariance [12]. Knowledge of the intrinsic eye parameters can be bypassed by using the parameters averaged for the human population because, in our setup, the differences have a negligible impact on the accuracy of calculating the eye position (see Section 9). This approach obviates the requirement for a dual-camera setup per eye, facilitating the estimation of individual ocular parameters [32].

Several studies have proposed methodologies for eye-position and gaze direction tracking based on the analysis of pupil ellipse shape variations within eye images [8, 36, 18, 6]. These techniques rely solely on the pupil's shape and offer the advantage of avoiding detecting and tracking glints, which require active IR illumination and are prone to false reflections. However, in setups utilizing co-axis cameras (as in our eye tracker) the magnitude of semi-axial length changes during minute eye movements may be lower than the accuracy of detecting the pupil's shape even for the high-resolution cameras.

Commercial Pupil Labs eye trackers also employ an algorithm based on detecting distortions of the pupil ellipse. This approach infers pupil position by first identifying its shape and constructing an eye model. However, the manufacturer recommends excluding cases where the aspect ratio of the pupil ellipse exceeds 0.8 from real-time model updates [3]. In contrast, our system observes a near-circular pupil shape across most eye rotation scenarios.

Eye position tracking in multi-focal-plane displays. Stengel et al. [34] proposed a virtual model-based method for eye position tracking within a head-mounted display (HMD). Their approach utilized a virtual eye rendered from various angles to estimate the 3D pupil center based on captured camera images. While this method achieved good correspondence with the physical system, it neglected eye slippage within the HMD and lacked validation for diverse eye sizes and positions.

Rathinavel et al. [28] emphasized a near-eye display design requiring real-time pupil tracking but acknowledged that their system assumed prior knowledge of the pupil position, so rendering the real-time tracking aspect became redundant. Similarly, Yu et al. [43] presented a multi-focal display that restricted light-field rendering to the pupil area, relying on pre-existing gaze direction data captured by a P-CR eye tracker under the assumption of a static head position. This approach mirrors the work of Jang et al. [16], where eye position tracking served to estimate pupil position for the pupil-tracked light projection on retina in a near-eye display. Neither approach addressed scenarios involving free head movement. Zhong et al. [45] employed head tracking to enhance image convergence in their multi-focal high dynamic range (HDR) display. Participants wore glasses equipped with IR LEDs, which were tracked by a high frame rate machine vision camera for real-time head position monitoring. While this approach is simple, it cannot track the eye's nodal point movement and is not robust to the slippage of the glasses. Ebner et al. [9] proposed a novel mixed-reality display architecture that aimed to address limitations in gaze tracking accuracy, particularly regarding depth perception in varifocal displays. They utilized a P-CR eye tracker to capture gaze position and approximate gaze direction. However, the extent to which eye position tracking data is integrated with multi-focal rendering remains unclear. The effect of small eye movements in multifocal displays can be partially compensated for by finding a multi-focal decomposition that is robust to such movements [19]. This, however, comes at a cost of degraded image quality and is limited to very small pupil movements.

The work the most closely related to ours is that of Mercier et al. [22], who implemented eye-position tracking in a multi-focal display. However, their work focuses on multi-focal-display decomposition and rendering, and it does not report any details or results on the performance of their eye tracking.

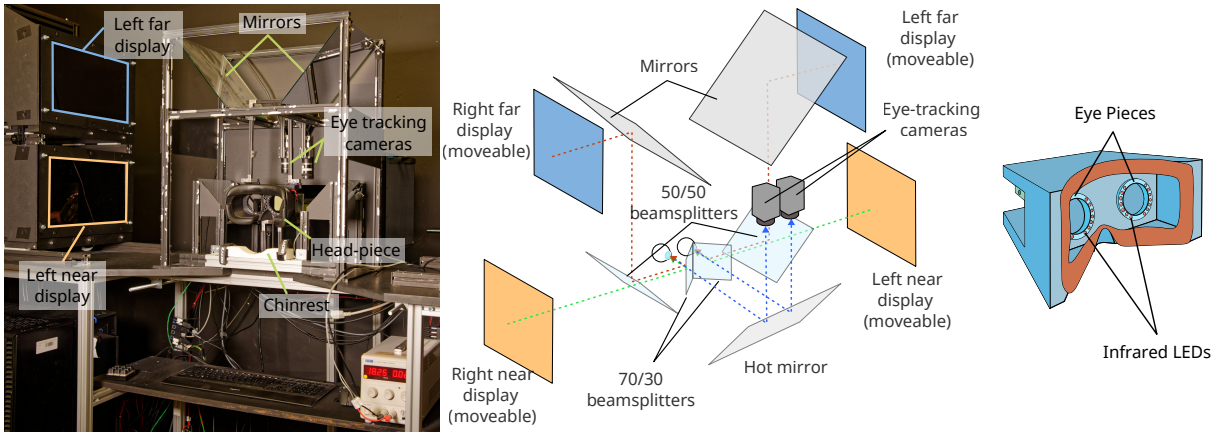
None of the works listed above provide sufficient details to reproduce the eye-tracking system and analyze its performance. This paper is meant to fill this gap and provide a practical recipe for implementing eye-position tracking within a near-eye volumetric display.

3 System Overview

Our eye position tracking is integrated into a high-dynamic-range multi-focal stereo display, such as the one described in Zhong et al. [45]. However, the methods explained here are equally applicable to any near-eye display, including head-mounted displays. We will first outline our hardware setup (Section 4), then we will explain how we adapted the eye-tracking model of Guestrin and Eizenman [12] to estimate both the position of the eye nodal point and the gaze direction (Section 5). The eye-tracking model requires accurate geometric calibration of the entire display and camera system, which we explain in Section 6. The eye-tracking model we use relies on fixed intrinsic eye parameters, which may differ from one user to another. For that reason, we need to perform per-user calibration (Section 7). Finally, knowing the position of the nodal point of each eye, we can compute projection matrices for our multi-focal rendering system (Section 8).

4 Hardware Configuration

The schematics and the photograph of our display system are shown in Figure 1. Our display is a haploscope, which offers two focal planes per eye. The two bottom display planes (shown in orange



(a) Photograph of the system (b) Schematic view of the system (c) Schematic view of the head-piece

Figure 1: (a) The photograph of the system, seen from the front. (b) The schematics of the high-dynamic-range multi-focal stereo display with the eye position tracking cameras, seen from the back. The dashed green and red lines represent the optical paths to the near and far display planes. The blue dashed lines represent the optical paths of the eye-tracking cameras. The schematic is shown without the head-piece, which is shown in (c).

in Figure 1) have a shorter optical path to the eye than the two upper displays (shown in blue in Figure 1) and appear nearer to an observer. The position of each display can be adjusted to control the viewing distance. When rendering 3D content on such a multi-focal display, we distribute pixel intensities between the two planes so that both the vergence and the accommodation of the eye are driven to the right distance [1].

As the display is meant to reproduce ultra-realistic 3D scenes, each display plane is a custom-built high-dynamic-range (HDR) display, which combines a 4k color LCD panel (15.6" IPS 3840×2160 4K LCD LQ156D1JX02) without a backlight and a 1080p projector (Acer P5535) providing the backlight. To focus the light from the projector on each eye, a Fresnel lens and a diffuser are placed behind each LCD panel. The HDR displays follow the same design and calibration procedure as those explained in [45] except that we use higher resolution panels and projectors.

The two small 70/30 (reflection/transmittance) beamsplitters in front of the eyes (see Figure 1) let the observer see a real scene in front of them, allowing us to simulate an optical-see-through AR display. We will later use this capability to calibrate the system to the global 3D coordinates in the real world. Another benefit of this configuration is that we can put a hot mirror (reflecting only infrared light, Edmund Optics 64–472) behind the beamsplitters and direct two eye-tracking cameras (IDS UI-3140CP-M-HQ Rev.2, Fujinon HF25HA-1B 1.4/25 mm lens, diagonal angle-of-view of 10°58'), one per eye, directly toward the eyes. This minimizes geometric distortions and extra reflections found in off-axis systems. The eye position tracking cameras are monochromatic and sensitive in the IR range, have a resolution of 1280 × 1024 pixels, and the nominal frame rate of 169 Hz.

Each eye is illuminated with a ring of IR LEDs (L-34SF4BT Kingbright, 880 nm), as shown in Figure 1(c). There are 14 IR LEDs in total, positioned in two groups of 7 on each side of the eye. Such positioning was selected to avoid shadows cast by eyelids and eyelashes. The LEDs also provide a source of reflections (glints), which are essential for our eye-position estimation algorithm. While the eye model we use requires just two LEDs, a larger number of LEDs improves the robustness and accuracy of the method and provides more uniform illumination of the eye.

5 Eye Tracking

To track both the position of the eye (its nodal point) and the gaze direction, we rely on the eye model of Guestrin and Eizenman [12]. The model builds the relationship between the position of the pupil and glints in the frames captured by the eye-tracking camera and the position and rotation of the eye in the 3D space. The main weakness of the model is that its accuracy depends on how well we can detect and track at least two glints in the images. The model also requires very accurate geometric calibration of each system component, which is often impossible in practice. Finally, the model relies on the intrinsic parameters of the "typical" eye, which may not be suitable for each individual observer. Since the original model is not well explained in the literature, we included a full step-by-step description in the supplementary document. In the following sections, we first focus on tracking the pupil and glints, which is specific to our system. Then, we explain the improvements made to address the main weaknesses of the original model.

Notation. We will use lower-case bold symbols to denote points, bold upper-case bold symbols for matrices, lower-case letters with an arrow for normalized directional vectors, and lower-case letters for scalars. Points and directional vectors are assumed to be column vectors.

5.1 Tracking pupil and glints

Reliable tracking of the pupil and glints in a multi-focal plane display system is non-trivial as such systems introduce multiple reflections (due to beamsplitters and mirrors) as those shown in Figure 2. Moreover, the glints from the top LEDs are often obscured by the eyelid or are blurred on the corneal surface. Below, we outline the main steps taken to ensure robust tracking.

Pupil detection and tracking. As the pupil is typically the darkest part of the image (it is a light trap), its shape can be detected by thresholding the input image. To determine the position of the pupil, we employ contour detection techniques [2, 35]. Given that the image may contain darker regions that could erroneously be classified as the pupil, certain contours must be discarded. We calculate the area of each contour and divide it by the area of the minimum enclosing circle. Based on the empirical tests, we measured the minimum and maximum pupil sizes in pixels, which allowed us to reject contours with diameters falling outside the expected range. We then select the contour with the highest resulting ratio (the most

circular) and mark the pupil center as the center of the enclosing circle.

To improve the stability of estimates across frames, we apply Kalman filters on the estimated pupil position [17]. The filter operates with a measurement dimensionality of 2, where the state dimensionality encompasses both the position and velocity of each parameter. We set the process noise covariance to 2 and the measurement noise covariance to 1.

Glint detection and tracking. To detect glints, we utilize a 9×8 image template, which is matched against every 9×8 pixel area within the image to identify the most probable glint positions. However, due to the presence of numerous secondary reflections, identifying the correct points is non-trivial. To address this challenge, we leverage temporal information and employ a sampling consensus approach akin to RANSAC [10].

We want to find a set of valid primary reflections and reject any higher-order reflections or other incorrectly detected bright features. The primary reflections are typically the brightest and form a circular shape, due to the arrangement of the IR LEDs on a ring. However, not all 14 primary reflections may be detected, and extra spurious reflections can be falsely detected. To ensure robust detection, we use a RANSAC procedure to fit a circle into the detected glint candidates.

In the given frame, we retain only the 20 reflections closest to the centroid of valid glints from the previous frame. These retained reflections are then grouped into triplets, as this is the minimum number of points required to fit a circle. We examine all possible combinations of such triplets (a total of $\binom{n}{3}$ combinations, where n is the number of remaining reflections). Then, a circle center $\boldsymbol{p}(p_x, p_y)$ and radius r are fitted using the Bayesian estimator:

$$\begin{aligned} \underset{\boldsymbol{p},r}{\operatorname{argmin}} \frac{\sum_{i=1}^{m}[(p_x - g_{\mathrm{x},i})^2 + (p_y - g_{\mathrm{y},i})^2 - r^2]^2}{m\,\sigma_g} + \\ + \frac{(p'_x - p_x)^2 + (p'_y - p_y)^2}{\sigma_c} + \frac{(r' - r)^2}{\sigma_r}, \end{aligned} \tag{1}$$

where $g_{\mathrm{x},i}, g_{\mathrm{y},i}$ are the x- and y- coordinates of the i-th glint in a triplet, and $\boldsymbol{p}'$, r' are the estimations from the previous frame. $\sigma_g = 2$ is the expected pixel error of the glint position, $\sigma_c = 2$ is the error of the glint center, and $\sigma_r = 5$ is the error of the radius (all in pixels). $m = 3$ when fitting the triplets.

Subsequently, after estimating the circle parameters for a given triplet using the Eq. (1), we calculate the distance of each remaining glint to the circle and we count the number of glints lying within a 3 pixel distance from the ring (determined empirically by testing for a range of eye rotations). This process is repeated for every glint triplet, and the triplet with the highest count of inliers is chosen. These glints, along with the original triplet, are then designated as the primary reflections, the circle is refitted to them and used for subsequent stages of the algorithm.

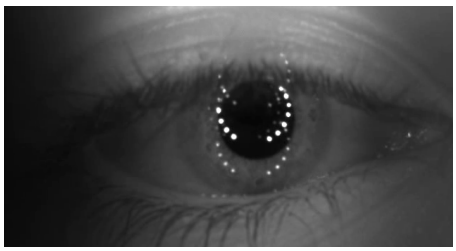


Figure 2: Example frame captured by the eye-tracking camera. The primary reflections, which are the brightest spots in the image, are partially obscured by the eyelid. Additionally, secondary reflections can be observed both below and above the primary reflections.

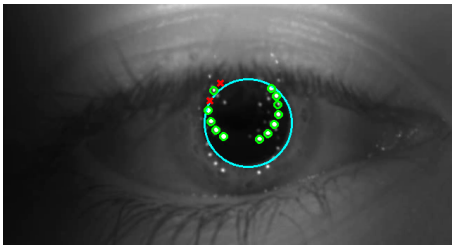


Figure 3: Camera image with the detected features. The blue circle indicates the detected contour of the pupil (its center is used for the calculations). Detected glints are denoted by green circles. Any glints that are absent due to being obscured by the eyelid or being too dark are represented by their expected positions marked with a red cross.

Identifying glints. In the final step of the algorithm, we identify specific glints and associate them with individual IR LEDs. Initially, we divide these glints into two groups: those located to the left of the Bayesian-estimated circle center and those to the right. Within each group, we sort the glints from bottom to top.

In all tested conditions, the bottom glints were consistently visible due to the placement of the LEDs. Hence, we utilized them as reference points. First, we calculated the angle $\Delta\theta$ determined by the first two glints and the ellipse center. Second, we computed the mean distance of all glints to the ellipse center $\bar{r}$ for each side separately. These calculated values were then employed to estimate the most likely locations of other glints using polar coordinates:

$$\begin{aligned} \theta_i &= \theta_1 + (i-1)\Delta\theta, \\ r_i &= \bar{r}, \end{aligned} \tag{2}$$

where (θ_i, r_i) represent the polar coordinates of the i-th glint.

Finally, all previously detected glints are compared to their estimated positions. If they lie in close proximity, they are linked to their corresponding LEDs and marked as detected. Conversely, if they are further away than the established threshold, they are rejected. Any empty spots lacking corresponding glints are marked as undetected. An example of the results of such detection is illustrated in Figure 3.

5.2 Eye model

Once we know the position of the pupil and glints in the image, we can use the model of Guestrin and Eizenman [12] to estimate the position and rotation of the eye. The model is described in detail in the supplementary document. Here, we focus on our improvements.

The model estimates the extrinsic parameters of the eye: 3D position of the eyeball center $\boldsymbol{e}$, the pupil center $\boldsymbol{p}$, and the position of the eye nodal point $\boldsymbol{n}$ (see Figure 4). These three points define the direction of the *eye's optical axis*. For simplicity, the center of the cornea is assumed to be coincident with the nodal point $\boldsymbol{n}$. The visual axis, representing the actual viewing direction, deviates from the optical axis by specific vertical and horizontal angles, which can vary among individuals [33, 11].

The model requires a minimum of two light sources whose reflections from the cornea's surface (glints) are captured by the eye-tracking camera. In the initial step, the line between the camera's nodal point $\boldsymbol{c}$ and the eye's nodal point $\boldsymbol{n}$ is computed. This line can be determined using a pair of planes, each defined for the i-th light source by three points: $\boldsymbol{c}$, $\boldsymbol{l'}_i$, and $\boldsymbol{l}_i$ (see Figure 4). In our system, we have up to 14 glints instead of two, which can give us $\binom{14}{2} = 91$ pairs we can use for a more robust estimate. To reduce that number, we select only the pairs that are likely to give us robust estimates; those that form a pair of planes at the angle of 45 degrees

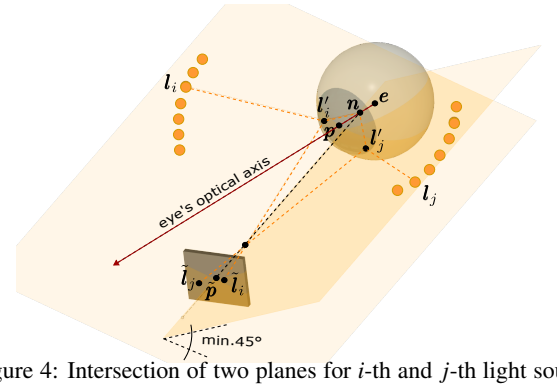


Figure 4: Intersection of two planes for i-th and j-th light sources. Two planes are defined by two tuples of three points each: camera nodal point $\boldsymbol{c}$, position of the light source ($\boldsymbol{l}_i$ or $\boldsymbol{l}_j$), and the center of the light glint in the image ($\tilde{\boldsymbol{l}}_i$ or $\tilde{\boldsymbol{l}}_j$).

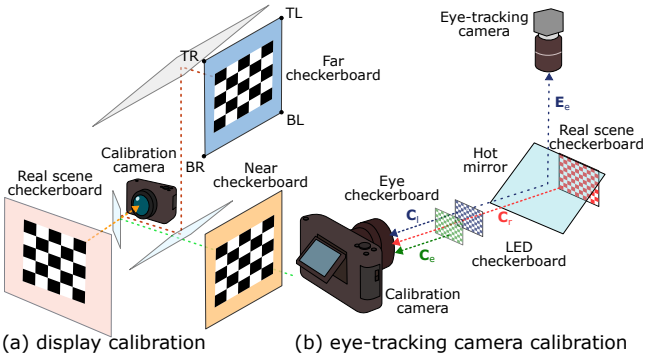


Figure 5: (a) Schematic view of the display calibration process (right- hand side only). The points {TL, TR, BL, BR} are marked on the far display. (b) Schematic view of the eye-tracking camera calibration process. The arrows indicate which checkerboard is used to calculate the corresponding matrix.

or more. The final estimate is computed by taking the median (of the directional vector) across all estimated $\boldsymbol{c}$–$\boldsymbol{n}$ lines.

In the next step, we follow an optimization procedure to compute the $\boldsymbol{n}$, as elaborated in the supplementary material. The estimation error is quantified as the sum of differences between the incidence and reflection angles of all glints. The calculated $\boldsymbol{n}$ is then utilized to estimate the position of the $\boldsymbol{e}$, following the same methodology as in the original model.

The final step of the model involves the estimation of the gaze direction. The optical axis is determined as the direction between $\boldsymbol{n}$ and $\boldsymbol{e}$. To calculate the visual axis, we rotate the optical axis by specific angles α and β, corresponding to horizontal and vertical rotations, respectively. The values for these angles were obtained as an average across the population [33, 11] and are detailed in Table 1 of the supplementary document.

6 System Geometric Calibration

The accurate 3D positions of our system components are required for both eye position tracking and multi-focal rendering. To accurately measure those, we employ a calibration procedure that relies on detecting multiple checkerboards with a high-resolution mirrorless camera (Sony α7R3 with a Sony SEL35F18F FE 1.8/35 lens). The camera operated with the electronic shutter since we found the mechanical (global) shutter was causing small pixel shifts in captured images.

6.1 Display plane positions

To accurately measure the position of screens corresponding to the near and far display planes, we position the camera in front of the display so that its nodal point is located near the nodal point of the corresponding eye (left or right), as shown in Figure 5a. We first establish the pose of the camera in the (real) world coordinates. Because all planes of our display are transparent, we can position a physical checkerboard behind the front 70/30 beamsplitter (see "Real scene chechkerboard" in Figure 5a) and estimate the extrinsic camera matrix $\boldsymbol{C}_\mathrm{r}$. This checkerboard will serve as a point-of-reference for the world coordinate system. Then, we display a checkerboard pattern spanning the size of each display plane and use it to estimate an extrinsic camera matrix with respect to each display plane $\boldsymbol{C}_{\mathrm{d}i}$, where $i \in \{\mathrm{LN}, \mathrm{LF}, \mathrm{RN}, \mathrm{RF}\}$ denotes the display plane, left-near, left-far, right-near or right-far. The position of four corners of each display plane in the world coordinates can then be found as:

$$\boldsymbol{d}_{i,j} = \boldsymbol{C}_\mathrm{r}^{-1} \boldsymbol{C}_{\mathrm{d}i} \boldsymbol{r}_{i,j}\,, \quad j \in \{\mathrm{TL}, \mathrm{TR}, \mathrm{BL}, \mathrm{BR}\}\,, \tag{3}$$

where $\boldsymbol{r}_{i,j} = [x_\mathrm{c}\ \mathrm{y}_c\ 0]^T$ is the position of a display corner in mm (local coordinates of the displayed checkerboard). j corresponds to the index of the corner: TL – top left, TR – top right, BL – bottom left, and BR – bottom right, with the top-left corner at the local coordinates $\boldsymbol{r}_{i,1} = [0\,0\,0]^T$.

6.2 Calibration of eye-tracking camera

The eye-tracking model requires knowledge of the pose of the eye-tracking cameras and the positions of the IR LEDs in the world space. Because the eye tracking cameras have no visibility of the real scene checkerboard or IR LEDs, the camera poses and LED positions need to be estimated indirectly using the same calibration camera as the one used to estimate display plane positions (Section 6.1).

As in Section 6.1, the real-scene checkerboard let us establish the extrinsic matrix of the calibration camera, $\boldsymbol{C}_r$ (Figure 5b). Then, we position another two-sided checkerboard so that the front side lies close to the nodal point of a typical observer's eye (green "eye checkerboard" in Figure 5b). This checkerboard lets us determine the extrinsic matrix of the calibration ($\boldsymbol{C}_\mathrm{e}$) and eye-tracking camera ($\boldsymbol{E}_\mathrm{e}$) with respect to the coordinate system of this checkerboard. To calculate the extrinsic matrix of the eye-tracking camera with respect to the real scene coordinates, we can chain the transformations:

$$\boldsymbol{M}_\mathrm{et} = \boldsymbol{E}_\mathrm{e} \boldsymbol{C}_\mathrm{e}^{-1} \boldsymbol{C}_\mathrm{r} \tag{4}$$

To estimate the positions of the IR LEDs, we use the calibration camera to capture both the LEDs and another "LED checkerboard" (see the blue checkerboard in Figure 5b). We need a separate checkerboard since focusing the calibration camera on the "eye checkerboard" would cause the LEDs to be out of focus. The LED positions are then manually marked on the image. The extrinsic matrix, $\boldsymbol{C}_\mathrm{l}$ is estimated, and local LED positions, $\boldsymbol{a}_i$ are derived from the image-space position of the markers in relation to the checkerboard. We then obtain each LED position, $\boldsymbol{l}_i$ in world space:

$$\boldsymbol{l}_i = \boldsymbol{C}_\mathrm{r}^{-1} \boldsymbol{C}_\mathrm{l} \boldsymbol{a}_i \tag{5}$$

7 Per-user Calibration

The eye model estimations made in Section 5.2 might not be universally applicable due to several assumptions about eye parameters that may not hold true for every individual. Factors such as variations in the horizontal angle between optical and visual axes (which can deviate by up to 5 degrees from the mean [42, 11, 5, 33]), the presence of corneal irregularities or inaccuracies in measuring display element positions could affect accuracy. Therefore, a calibration needs to be performed for each individual. Its purpose is to

estimate the position of the center of eye rotation ($\boldsymbol{e}$) and gaze direction. Then, this position is used in the eye-tracking model to fine-tune the predictions.

7.1 Eye-position calibration

The eye position calibration leverages the two focal planes offered by our display. During this step, we show a rectangular grid on the near and far planes of the display (see Figure 6). The user can drag with a mouse the corners of the grid displayed on the near plane. The user's task is to align the grid of the far plane with the grid of the near plane so that the grid lines overlap.

The difficulty here is that when the user performs this task, the rays originating from $\boldsymbol{o}_i$ in the direction $\vec{w}$ (see Figure 7a) are aligned with the user's visual axis. The visual axis does not cross the center of eye rotation, $\boldsymbol{e}$, but it instead crosses the nodal point. As the eye rotates when looking at each corner, the nodal point moves. Our task is to solve for the eye rotation center, $\boldsymbol{e}$, knowing the angle between the visual and optical axes [33, 11].

The display plane calibration (Section 6.1) gave us the 3D positions of the display planes in the world coordinates. We can use those to find the 3D positions of the grid corners in each display and, therefore, rays with the origins $\boldsymbol{o}_i$ and directions $\vec{w}_i$ in the world coordinates.

The position of each nodal point $\boldsymbol{n}_i$ can be expressed as:

$$\boldsymbol{n}_i = \boldsymbol{e} + k_{ne}\,\vec{v}_i \tag{6}$$

where k_{ne} represents the fixed distance between the nodal point and

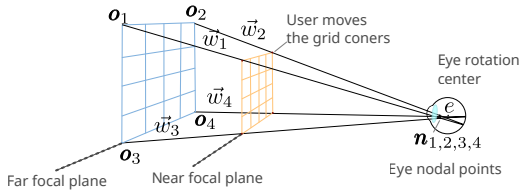


Figure 6: Schematic view of the calibration grids used in per-user calibration. The corners of the near-focal-plane grid (orange) could be dragged with a mouse. The rays originating from the corners $\boldsymbol{o}$ in the direction $\vec{w}$ are aligned with the visual axis and cross the nodal point of the eye, which is different depending on which corner the user looks at.

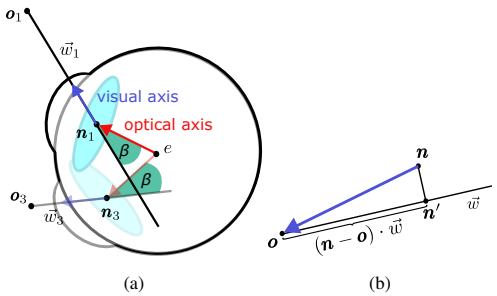


Figure 7: (a) The location of the eye rotation center ($\boldsymbol{e}$) in relation to the first end third corners of the grid. Two blended eye images are depicted: one where the user is looking in the direction $-\vec{w}_1$, and another where they are looking in the direction $-\vec{w}_3$. After rotating the gaze direction (i.e., the visual axis) by the specific angle β at the nodal points $\boldsymbol{n_1}$ and $\boldsymbol{n_3}$, the obtained optical axes intersect at $\boldsymbol{e}$. Both the eye rotation and the angle β are shown larger than in reality for better visualization. (b) The calculation used to find the projection of the estimated eye nodal point $\boldsymbol{n}$ on the visual axis. During the optimization, we minimize the distance between $\boldsymbol{n}$ and $\boldsymbol{n}'$.

the center of rotation (see Table 1 of the supplementary document). $\vec{v}$ denotes the optical axis, which is obtained by rotating the measured visual axis, $-\vec{w}$, by angles α and β listed in Table 1 of the supplementary document.

Two rays $\vec{w}$ are sufficient to triangulate the eye rotation center, but we use all four rays to improve the accuracy. We first find the projection of the nodal point $\boldsymbol{n}_i$ on the measured visual axis $\vec{w}$ (see Figure 7b):

$$\boldsymbol{n}_i' = \boldsymbol{o}_i + \vec{w}_i\,[(\boldsymbol{n}_i - \boldsymbol{o}_i)\cdot\vec{w}] = \boldsymbol{o}_i + \vec{w}_i\,[(\boldsymbol{e} + k_{ne}\,\vec{v}_i - \boldsymbol{o}_i)\cdot\vec{w}] \tag{7}$$

Then, we want to find the eye rotation center, $\boldsymbol{e}$, for which the projections are closest to the nodal point defined by the optical axis (Eq. (6)):

$$\boldsymbol{e} = \underset{\boldsymbol{e}}{\operatorname{argmin}} \sum_{i=1}^{4} ||\boldsymbol{n}_i' - \boldsymbol{n}_i|| = \underset{\boldsymbol{e}}{\operatorname{argmin}} \sum_{i=1}^{4} ||\boldsymbol{n}_i' - \boldsymbol{e} - k_{ne}\,\vec{v}_i|| \tag{8}$$

7.2 Gaze-direction calibration

After completing the eye position calibration, users proceed with a gaze direction calibration. They are presented 15 markers (crosses) appearing successively in various parts of the screen, covering an $8^\circ \times 8^\circ$ area. Their task is to look at these markers without changing their head position. Leveraging the knowledge of the eye rotation center, we can calculate the reference angles of gaze direction. We then use the RANSAC algorithm to fit a polynomial function. This involves randomly selecting 10,000 sets of 6 samples from the captured marker data. Each set undergoes polynomial fitting, and its error against the remaining samples is calculated. The set with the most samples below a 0.5-degree error threshold is chosen, and a refined polynomial is fitted specifically to it. This approach helps mitigate outliers and ensures robust calibration.

7.3 Eye model fine-tuning

The two calibration steps explained in the previous sections let us fine-tune the model and improve its accuracy. Such a fine-tuning step compensates for any deviation from the model assumptions (e.g., intrinsic eye parameters) and for any inaccuracies of geometric calibration.

To fine-tune eye-position predictions (the eye rotation center), we utilize both the eye-position (Section 7.1) and gaze-direction data (Section 7.2). We assume that the rotation center stays fixed during the gaze direction calibration but the nodal point changes with the gaze direction. We use the measured eye rotation center, $\boldsymbol{e}$, and Eq. (6) to estimate the nodal point for each gaze-direction marker. Then, we calculate the difference between the nodal point positions calculated in this calibration step and those predicted by the model. We use a robust mean of those differences (excluding measurements 2σ outside the mean) as the offset to the eye rotation center predicted by the model. Note that we do not directly use the difference between the model prediction of $\boldsymbol{e}$ and $\boldsymbol{e}$ estimated in Eq. (8) because the gaze-direction calibration offers more measurements and we can estimate the nodal point position without the need to track the pupil.

While adding an offset to the estimated eye position produced satisfactory results (as demonstrated in Section 9.1), such a straightforward adjustment did not yield desirable outcomes for gaze direction estimation. This limitation stems from the fact that the expected offset varies depending on the viewing direction; the further the gaze is from the center, the greater the error. Therefore, we adopt the polynomial fitting method employed by the P-CR eye tracking. It is noteworthy that, unlike P-CR scenarios, the actual (absolute) gaze directions passing through the eye nodal points are known in our case.

We employed a multivariate polynomial for the gaze direction parameters, consisting of one polynomial for the horizontal angle

θ and another for the vertical angle ϕ. We utilized second-degree polynomials:

$$f_1(\theta,\phi) = r_{1,\theta} + r_{2,\theta}\ \theta + r_{3,\theta}\ \theta^2 + r_{4,\theta}\ \phi + r_{5,\theta}\ \phi^2 + r_{6,\theta}\ \theta\phi,$$
$$f_2(\theta,\phi) = r_{1,\phi} + r_{2,\phi}\ \theta + r_{3,\phi}\ \theta^2 + r_{4,\phi}\ \phi + r_{5,\phi}\ \phi^2 + r_{6,\phi}\ \theta\phi, \tag{9}$$

where $r_{i,x}$ and $r_{i,\phi}$ represent the polynomial coefficients for horizontal and vertical angle, respectively, while θ and ϕ are the angles estimated directly by the model. The mapping between the estimation and reference (e.g., the horizontal angle θ) can be expressed in matrix form as follows:

$$\begin{bmatrix} 1 & \theta_1 & \theta_1^2 & \phi_1 & \phi_1^2 & \theta_1\phi_1 \\ 1 & \theta_2 & \theta_2^2 & \phi_2 & \phi_2^2 & \theta_2\phi_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \theta_N & \theta_N^2 & \phi_N & \phi_N^2 & \theta_N\phi_N \end{bmatrix} \begin{bmatrix} r_{1,\theta} \\ r_{2,\theta} \\ \vdots \\ r_{6,\theta} \end{bmatrix} = \begin{bmatrix} \theta'_1 \\ \theta'_2 \\ \vdots \\ \theta'_N \end{bmatrix}, \tag{10}$$

where θ'_i represents the reference angle for sample i, and N is the total number of samples obtained during the calibration process. The polynomial coefficients can be estimated by computing the pseudoinverse of the variable matrix.

8 Multi-focal plane rendering

For geometrically accurate rendering of 3D objects on a multi-focal plane (MFP) display, we need to find projection matrices that convert world coordinates into pixel coordinates of each display plane. This projection is more complicated than the standard projection used in computer graphics because display planes are typically slanted and not orthogonal to the visual axis.

First, we find a matrix projecting 3D world coordinates into the display plane with respect to the eye nodal point $\boldsymbol{n}$. Such projection matrix for the display plane i can be computed as:

$$\boldsymbol{M}_{\mathrm{p},i} = \begin{bmatrix} \boldsymbol{n} \\ 1 \end{bmatrix} \begin{bmatrix} \vec{n}_{\mathrm{d}i} & -(\boldsymbol{d}_{i,\mathrm{TL}} \cdot \vec{n}_{\mathrm{d}i}) \end{bmatrix} + \left((\boldsymbol{d}_{i,\mathrm{TL}} - \boldsymbol{n}) \cdot \vec{n}_{\mathrm{d}i} \right) \boldsymbol{I}_4, \tag{11}$$

where $\cdot$ is the dot product and $\boldsymbol{I}_4$ is a 4×4 identity matrix. The normal of the display plane, $\vec{n}_{\mathrm{d}i}$, can be found from the display corners (see Eq. (3)):

$$\vec{n}_{\mathrm{d}i} = \frac{(\boldsymbol{d}_{i,\mathrm{TR}} - \boldsymbol{d}_{i,\mathrm{TL}}) \times (\boldsymbol{d}_{i,\mathrm{BL}} - \boldsymbol{d}_{i,\mathrm{TL}})}{\left\| (\boldsymbol{d}_{i,\mathrm{TR}} - \boldsymbol{d}_{i,\mathrm{TL}}) \times (\boldsymbol{d}_{i,\mathrm{BL}} - \boldsymbol{d}_{i,\mathrm{TL}}) \right\|}, \tag{12}$$

where $\times$ is a cross product.

Finally, we find a transformation from the world coordinates to the pixel coordinates. We can do it by translating, rotating, and scaling the coordinates:

$$\boldsymbol{M}_{\mathrm{x},i} = \boldsymbol{S}\left(w_{\mathrm{pix}}/w_{\mathrm{mm}}\right) \boldsymbol{R}\boldsymbol{T}(-\boldsymbol{d}_{i,\mathrm{TL}}) \boldsymbol{M}_{\mathrm{p},i}, \tag{13}$$

where $\boldsymbol{T}(\cdot)$ and $\boldsymbol{S}(\cdot)$ are 4×4 translation and scaling matrices. We pass to the scaling matrix the ratio of display width in pixels and millimeters (assuming square pixels). The rotation matrix is obtained from the normalized vectors aligned with the display edges and the normal:

$$\boldsymbol{R} = \begin{bmatrix} \frac{\boldsymbol{d}_{i,\mathrm{TR}} - \boldsymbol{d}_{i,\mathrm{TL}}}{\|\boldsymbol{d}_{i,\mathrm{TR}} - \boldsymbol{d}_{i,\mathrm{TL}}\|} & \frac{\boldsymbol{d}_{i,\mathrm{BL}} - \boldsymbol{d}_{i,\mathrm{TL}}}{\|\boldsymbol{d}_{i,\mathrm{BL}} - \boldsymbol{d}_{i,\mathrm{TL}}\|} & \vec{n}_{\mathrm{d}i} & \boldsymbol{0}_{3,1} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{14}$$

Matrix $\boldsymbol{M}_{\mathrm{x},i}$ lets us project a point in the world coordinates into the pixel coordinates of i-display plane. For multi-focal plane rendering, we may also need to know the distance from one of the focal planes to a given point in the world space (e.g., for depth blending [1]). This can be trivially computed as the distance between the given point and its projection on one of the planes (Eq. (11)).

Table 1: The accuracy of the eye (nodal point) position estimation and the ablation studies. The reported values are means and standard deviations of the error in millimeters.

Ablation	Session 1	Session 2	Session 3
Complete system	1.08 ± 0.89	1.30 ± 1.16	1.11 ± 0.85
w/o glint RANSAC	7.30 ± 22.63	3.59 ± 4.69	7.80 ± 17.33
using two glints	9.82 ± 49.24	7.76 ± 37.43	5.41 ± 30.74
uncalibrated	20.23 ± 9.83	20.54 ± 23.11	20.30 ± 21.37

9 Testing and validation

We first report the numerical error of the eye and gaze-position estimation for our system, its ablations and the P-CR approach. We then demonstrate an example of how our system can compensate for the eye-movement in a multi-focal plane display.

9.1 Accuracy and precision

We conducted two experiments to assess the accuracy of eye position and gaze tracking through a user study. Six volunteers, four male and two female with ages in range 23 – 45, completed our experiments. The departmental ethics panel approved the experiment.

Eye position To estimate the error of eye position prediction, participants were asked to move their head to align in their vision square patterns shown on the near and far planes of the display and press a button to confirm the alignment. Such an alignment lets us estimate the position of the eye (nodal point) using similar methods as those explained in Section 7.1. Then, we could compare the measured position and that predicted by the model. In total, we tested 18 positions spread across a 8×8 mm eye box of our display.

The experiment consisted of three sessions. In the first session, alignment was preceded by eye-position and gaze-direction calibration, used to fine-tune the model for each participant (Section 7). In the second and third sessions, participants used the same calibration profile without any re-calibration. In the third session, participants were instructed to move their head away from the display after each alignment, which let us measure the model accuracy when the eye cannot be continuously tracked.

As there is no alternative system we could compare with, we used the collected results for ablation studies. We disabled the following components in the ablations:

- *w/o glint RANSAC* — the component responsible for removing outliers from the glint candidates, explained in "Glint detection and tracking" in Section 5.1.
- *using two glints* — using two bottom glints instead of up to 14 glints, as explained in Section 5.2.
- *uncalibrated* — the model prediction without fine-tuning for each participant (Section 7.3).

The results from the eye position experiment, presented in Table 1, show that each component of the system was essential for achieving acceptable accuracy. RANSAC for glint candidates was necessary to exclude falsely detected glints. Using only two glints made the result unstable. And per-user calibration was essential to account for individual differences and any inaccuracy of the geometric calibration. Another noteworthy observation is that the error did not increase in sessions 2 and 3, which were run without re-calibration. The differences in errors are not statistically significant: between session 1 and 2 (2-sided t-test, $p = 0.11$, $t(214) = -1.62$), and session 1 and 3 ($p = 0.81$, $t(214) = -0.24$). This demonstrates that the system needs to be calibrated only once per user.

Table 2: The accuracy of gaze estimation in degrees for our system, simple offset correction, and the P-CR approach.

Ablation	Our system	Offset correction	P-CR
Complete system	1.53 ± 1.17	2.03 ± 1.61	2.76 ± 2.23
w/o glint RANSAC	10.16 ± 17.37	10.81 ± 7.44	5.48 ± 5.14
using two glints	1.66 ± 1.39	2.01 ± 1.55	2.62 ± 1.59
uncalibrated	7.32 ± 3.30	—	—

Gaze direction The gaze-direction error was measured in the fourth and final session of the experiment, relying on the gaze-direction calibration (Section 7.2) performed before the first session. The gaze direction was measured from four different eye positions to simulate "slippage". To bring the eye to the known position, the participants were asked to align square patterns on both planes, as in the eye-position measurements above. Then, they were presented with the same 15 markers as in the gaze direction calibration (Section 7.2), but the measured directions were used for the validation rather than calibration.

We performed the same ablations as for the eye position experiment. We compare our system, with a simplified fine-tuning (offset correction), and with the standard P-CR approach. In the "offset correction" configuration, we replaced the 2-nd degree polynomial (Eq. (9)) with a simple offset of angles (0-degree polynomial). For the P-CR approach, we estimated the vertical and horizontal distance between the pupil and the center of an ellipse formed by the glints during calibration. This distance was fitted to a polynomial, which mapped it to a specific location on the screen.

The results from the gaze direction experiment are presented in Table 2. Compared to the PC-R approach, our system resulted in a significantly lower error (2-sided t-test, $p < 0.001$, $t(644) = -8.74$). It should be noted that both our and PC-R approaches were tested without recalibration (both PC-R and our system can achieve a smaller error when recalibrated for each session and eye position). Using a simple "offset correction" resulted in a small increase in the error ($p < 0.001$, $t(644) = -4.45$). The ablations confirmed that RANSAC is necessary to remove falsely detected glint and the system needs to be calibrated for each participant. Interestingly, using more than 2 glints did not improve the accuracy of the gaze direction estimation ($p = 0.22$, $t(644) = -1.23$).

9.2 System testing

We finally test whether our tracking system can successfully compensate for changes in eye position in a multi-focal rendering system. Because we could not measure the reference eye position for people participating in the experiment, we cannot report numerical results. Instead, we encourage watching the supplementary video demonstrating how the eye-position compensation works in practice. In Figure 8, we show a photo of the rendered scene, illustrating the effect of misalignment.

We employ an online eye tracking algorithm for real-time processing. The algorithm operates on individual frames with a resolution of 1280x1024 pixels. The entire processing pipeline, from frame acquisition to eye parameter estimation, takes approximately 15 milliseconds, corresponding to a frame rate of 66 fps. A breakdown of the processing time for key steps is provided: frame preprocessing: 6 ms, pupil detection: 0.5 ms, glint detection: 5.0 ms, nodal point and gaze direction estimation: 3.5 milliseconds. It is important to note that the C and CUDA code used for this implementation was not specifically optimized for performance. The timings reported here were collected on an Intel i9-9900K CPU and a GeForce GTX 1080 GPU.

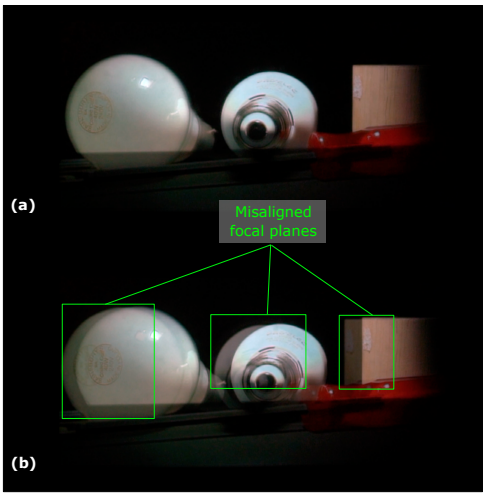


Figure 8: Example of the rendered output of our system with eye-tracking enabled (a) and without (b). Note the visible misalignment between the focal planes in (b).

10 Discussion and Limitations

Our system was tested in the on-axis camera configuration, in which the camera's and eye's optical axes are similar and the pupil is circular. It is possible to adapt the system to the off-axis camera configuration (more compact, no need for a hot mirror). However, the on-axis configuration is a better choice for accurate tracking of the eye position in the two directions that are crucial for the plane alignment in a multi-focal-plane display — vertical and horizontal. When the off-axis camera is placed below the eye, the tracking will be less sensitive to the vertical eye movement.

The work of Lee at al. [19] demonstrates how a multi-focal system can be drastically reduced in size with the help of the holographic optical element (HOE) to be suitable for a head-mounted AR display. A similar HOE with an additional layer can be used to redirect the infrared light toward the eye-tracking camera [44]. In the context of those designs, our method is especially attractive as a way to compensate for slippage in HMDs. It is also possible to combine our tracking with the multi-focal decomposition that is robust to eye movement [19] to reduce the tracking latency and accuracy requirements.

11 Conclusions

Tracking of the eye position and gaze direction is a necessity for many volumetric displays that require alignment of focal planes. Such tracking can also benefit displays with a small eye box, which require compensation for the pupil position. This work reports on a workable solution for tracking the eye position and gaze direction in near-eye displays. It robustly tracks multiple glints and the pupil and infers from those the eye position and rotation. The inference model requires very accurate geometric calibration, which we achieve using computer vision techniques. The inference is then corrected based on a per-user calibration procedure. Unlike other popular solutions (e.g., P-CR eye tracking), we require one-time calibration per user that can be reused across the sessions.

Acknowledgements

We would like to thank Akshay Jindal, Jize Sha, and Dmitry Lubyako for the help with the display and in the early stages of the project, and Dongyeon Kim for his feedback. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement N° 725253–EyeCode).

REFERENCES

[1] K. Akeley, S. Watt, A. Girshick, and M. Banks. A stereo display prototype with multiple focal distances. *ACM Transactions on Graphics (TOG)*, 23(3):804–813, 2004. 1, 3, 7

[2] E. M. Arvacheh and H. R. Tizhoosh. Iris segmentation: Detecting pupil, limbus and eyelids. In *2006 International Conference on Image Processing*, pp. 2453–2456. IEEE, 2006. 3

[3] K. Barkevich, R. Bailey, and G. J. Diaz. Using deep learning to increase eye-tracking robustness, accuracy, and precision in virtual reality. *arXiv preprint arXiv:2403.19768*, 2024. 2

[4] J.-H. R. Chang, B. V. K. V. Kumar, and A. C. Sankaranarayanan. Towards multifocal displays with dense focal stacks. *ACM Trans. Graph.*, 37(6), dec 2018. doi: 10.1145/3272127.3275015 1

[5] M. Corbett, N. Maycock, E. Rosen, and D. O'Brart. *Corneal topography: principles and applications*. Springer, 2019. 5

[6] K. Dierkes, M. Kassner, and A. Bulling. A novel approach to single camera, glint-free 3d eye model fitting including corneal refraction. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, pp. 1–9, 2018. 2

[7] M. Dubbelman, V. Sicam, and G. Van der Heijde. The shape of the anterior and posterior surface of the aging human cornea. *Vision research*, 46(6-7):993–1001, 2006. 2

[8] Y. Ebisawa. Realtime 3d position detection of human pupil. In *2004 IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2004.(VCIMS).*, pp. 8–12. IEEE, 2004. 2

[9] C. Ebner, S. Mori, P. Mohr, Y. Peng, D. Schmalstieg, G. Wetzstein, and D. Kalkofen. Video see-through mixed reality with focus cues. *IEEE Transactions on Visualization and Computer Graphics*, 28(5):2256–2266, 2022. 2

[10] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981. doi: 10.1145/358669.358692 4

[11] A. G. Gale. A note on the remote oculometer technique for recording eye movements. *Vision research*, 22(1):201–202, 1982. 4, 5, 6

[12] E. D. Guestrin and M. Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on biomedical engineering*, 53(6):1124–1133, 2006. 1, 2, 3, 4

[13] D. W. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):478–500, 2009. 2

[14] C. Hennessey, B. Noureddin, and P. Lawrence. A single camera eye-gaze tracking system with free head motion. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, pp. 87–94, 2006. 2

[15] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer. *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford, 2011. 2

[16] C. Jang, K. Bang, S. Moon, J. Kim, S. Lee, and B. Lee. Retinal 3d: augmented reality near-eye display via pupil-tracked light field projection on retina. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017. 2

[17] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960. 4

[18] C.-C. Lai, S.-W. Shih, and Y.-P. Hung. Hybrid method for 3-d gaze tracking using glint and contour features. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(1):24–37, 2014. 2

[19] S. Lee, J. Cho, B. Lee, Y. Jo, C. Jang, D. Kim, and B. Lee. Foveated Retinal Optimization for See-Through Near-Eye Multi-Layer Displays. *IEEE Access*, 6:2170–2180, Dec. 2017. doi: 10.1109/ACCESS.2017.2782219 2, 8

[20] K. J. MacKenzie, R. A. Dickson, and S. J. Watt. Vergence and accommodation to multiple-image-plane stereoscopic displays: "real world" responses with practical image-plane separations? *Journal of Electronic Imaging*, 21(1):1 – 9, 2012. doi: 10.1117/1.JEI.21.1.011002 1

[21] J. Merchant, R. Morrissette, and J. L. Porterfield. Remote measurement of eye direction allowing subject motion over one cubic foot of space. *IEEE Trans. Biomed. Eng.*, 21(4):309–317, July 1974. doi: 10.1109/TBME.1974.324318 2

[22] O. Mercier, Y. Sulai, K. Mackenzie, M. Zannoli, J. Hillis, D. Nowrouzezahrai, and D. Lanman. Fast gaze-contingent optimal decompositions for multifocal displays. *ACM Transactions on Graphics*, 36(6):1–15, nov 2017. doi: 10.1145/3130800.3130846 2

[23] C. H. Morimoto, A. Amir, and M. Flickner. Detecting eye position and gaze from a single camera and 2 light sources. In *2002 International Conference on Pattern Recognition*, vol. 4, pp. 314–317. IEEE, 2002. 2

[24] C. H. Morimoto, D. Koons, A. Amir, and M. Flickner. Pupil detection and tracking using multiple light sources. *Image and vision computing*, 18(4):331–335, 2000. 2

[25] C. H. Morimoto and M. R. Mimica. Eye gaze tracking techniques for interactive applications. *Computer vision and image understanding*, 98(1):4–24, 2005. 2

[26] D. C. Niehorster, T. Santini, R. S. Hessels, I. T. C. Hooge, E. Kasneci, and M. Nyström. The impact of slippage on the data quality of head-worn eye trackers. *Behav. Res. Methods*, 52(3):1140–1160, June 2020. doi: 10.3758/s13428-019-01307-0 2

[27] K. Rathinavel, H. Wang, A. Blate, and H. Fuchs. An extended depth-at-field volumetric near-eye augmented reality display. *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2857–2866, Nov. 2018. Citation Key: Rathinavel2018. doi: 10.1109/TVCG.2018.2868570 1

[28] K. Rathinavel, H. Wang, A. Blate, and H. Fuchs. An extended depth-at-field volumetric near-eye augmented reality display. *IEEE transactions on visualization and computer graphics*, 24(11):2857–2866, 2018. 2

[29] J. P. Rolland, M. W. Krueger, and A. Goon. Multifocal planes head-mounted displays. *Applied Optics*, 39(19):3209, July 2000. doi: 10.1364/AO.39.003209 1

[30] T. Santini, D. C. Niehorster, and E. Kasneci. Get a grip: Slippage-robust and glint-free gaze estimation for real-time pervasive head-mounted eye tracking. In *Proceedings of the 11th ACM symposium on eye tracking research & applications*, pp. 1–10, 2019. 2

[31] S.-W. Shih and J. Liu. A novel approach to 3-d gaze tracking using stereo cameras. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):234–245, 2004. 2

[32] S.-W. Shih, Y.-T. Wu, and J. Liu. A calibration-free gaze tracking technique. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 4, pp. 201–204. IEEE, 2000. 2

[33] A. Slater and J. Findlay. The measurement of fixation position in the newborn baby. *Journal of Experimental Child Psychology*, 14(3):349–364, 1972. 4, 5, 6

[34] M. Stengel, S. Grogorick, M. Eisemann, E. Eisemann, and M. A. Magnor. An affordable solution for binocular eye tracking and calibration in head-mounted displays. In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 15–24, 2015. 2

[35] S. Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985. 3

[36] L. Świrski and N. A. Dodgson. A fully-automatic, temporal approach to single camera, glint-free 3d eye model fitting. *Proceedings of ECEM*, 2013, 2013. 2

[37] A. Villanueva and R. Cabeza. Models for gaze tracking systems. *EURASIP Journal on Image and Video Processing*, 2007:1–16, 2007. 2

[38] A. Villanueva, R. Cabeza, and S. Porta. Eye tracking: Pupil orientation geometrical modeling. *Image and Vision Computing*, 24(7):663–679, 2006. 2

[39] A. Villanueva, J. J. Cerrolaza, and R. Cabeza. Geometry issues of gaze estimation. *Advances in Human Computer Interaction*, p. 513, 2008. 2

[40] K. P. White, T. E. Hutchinson, and J. M. Carley. Spatially dynamic calibration of an eye-tracking system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(4):1162–1168, 1993. 2

[41] A. L. Yarbus. Determination of the absolute velocity of the eye from

photographs. *Biophysics*, 6(2):155–170, 1941. 1

[42] L. R. Young and D. Sheena. Survey of eye movement recording methods. *Behavior research methods & instrumentation*, 7(5):397–429, 1975. 5

[43] H. Yu, M. Bemana, M. Wernikowski, M. Chwesiuk, O. T. Tursun, G. Singh, K. Myszkowski, R. Mantiuk, H.-P. Seidel, and P. Didyk. A perception-driven hybrid decomposition for multi-layer accommodative displays. *IEEE transactions on visualization and computer graphics*, 25(5):1940–1950, 2019. 2

[44] J. Zhao, B. Chrysler, and R. K. Kostuk. Design of a high-resolution holographic waveguide eye-tracking system operating in near-infrared with conventional optical elements. *Opt. Express*, 29(15):24536–24551, July 2021. doi: 10.1364/OE.433572 8

[45] F. Zhong, A. Jindal, A. O. Yöntem, P. Hanji, S. J. Watt, and R. K. Mantiuk. Reproducing reality with a high-dynamic-range multi-focal stereo display. *ACM Trans. Graph.*, 40(6), dec 2021. doi: 10.1145/3478513.3480513 1, 2, 3