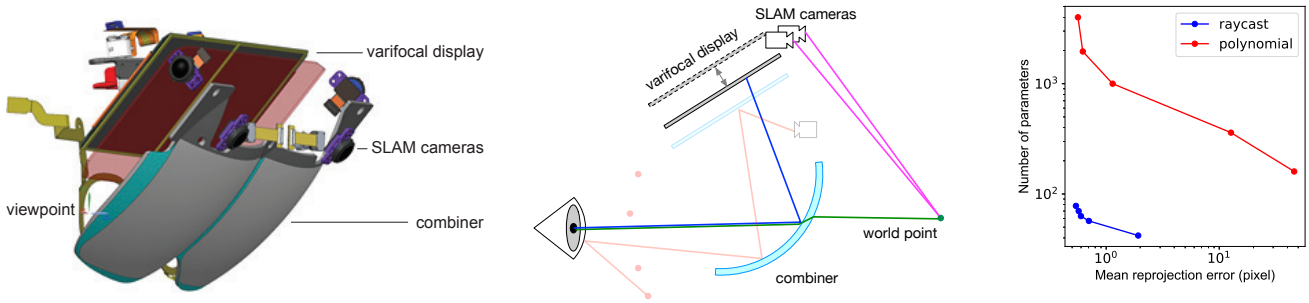
Raycast Calibration for Augmented Reality HMDs with Off-Axis Reflective Combiners

Qi Guo, Huixuan Tang, Aaron Schmitz, Wenqi Zhang, Yang Lou, Alexander Fix, Steven Lovegrove, and Hauke Malte Strasdat

Abstract—Augmented reality overlays virtual objects on the real world. To do so, the head mounted display (HMD) needs to be calibrated to establish a mapping between 3D points in the real world with 2D pixels on display panels. This distortion is a high-dimensional function that also depends on pupil position and varifocal settings. We present *Raycast calibration*, an efficient approach to geometrically calibrate AR displays with off-axis reflective combiners. Our approach requires a small amount of data to estimate a compact, physics-based, and ray-traceable model of the HMD optics. We apply this technique to automatically calibrate an AR prototype with display, SLAM and eye-tracker, without user in the loop.

Index Terms—augmented reality, raycast calibration, off-axis reflectors



(a) Head mounted display prototype (b) Raycast calibration model (c) Raycast vs. polynomial model

Fig. 1: To render virtual objects overlaid with the real world, head-mounted displays (HMDs) need to be accurately calibrated to accommodate form and mounting errors in its optics. (a) Optomechanical design of an AR HMD prototype used in this paper. The device consists of inside-out tracking, varifocal display, and eye tracking. (b) The proposed Raycast model of the HMD prototype. It is physically interpretable, and can be ray-traced. The model is decomposed into a SLAM-and-display system (opaque), and an eye tracker (semi-transparent). (c) Quantitative comparison between the proposed Raycast model and the polynomial model used in many state-of-the-art HMDs [1], [2], [3]. The Raycast model uses 10 times fewer parameters than the polynomial one to achieve sub-pixel calibration accuracy.

1 Introduction

Augmented reality (AR) delivers novel experiences by simultaneously presenting to users the real world and overlaid virtual objects [4]. Among many designs for see-through displays, those based on off-axis reflective combiners [5], [6] are of particular interest to recent industrial applications [7], [8], [9], [10], [11] due to their low fabrication cost. For these AR head mounted displays (HMDs), two fundamentals are accurate localization of the HMD with respect to the environment, and realistic display that augments the real scene.

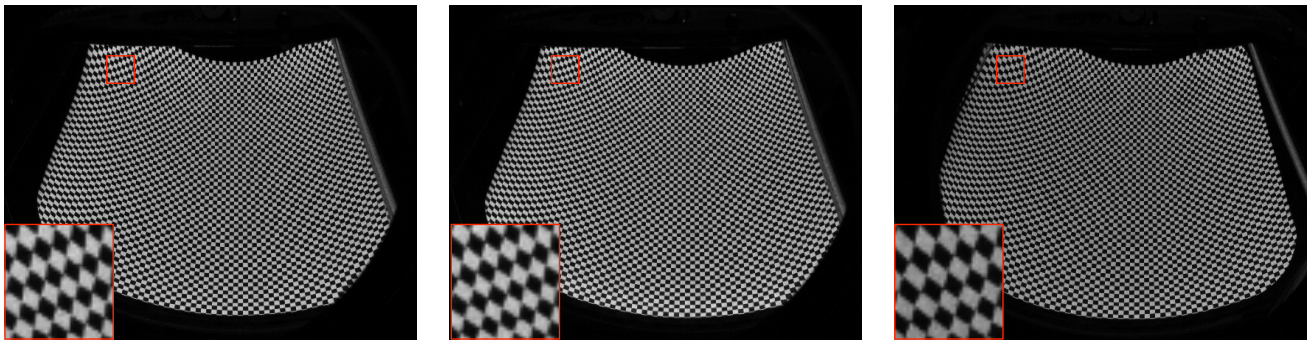
Fig. 1a shows one such AR prototype which consists of a Simultaneous Localization and Mapping (SLAM) system to enable Inside Out-Tracking [12], [13], an optical see-through AR display for rendering virtual objects and an eye tracker. The display uses an off-axis combiner with free-form optical surfaces. The display also has mechanical varifocal in order to resolve vergence accommodation conflict [14]. Specifically the display panel can translate along a linear axis so that the its virtual image is focused at different depths. We use this prototype as a running example throughout this paper.

To precisely align the geometry between the real and virtual worlds, the AR system needs to understand how 3D world points correspond to pixels in each display panel. Notably, this mapping is highly nonlinear and rotationally asymmetric unlike conventional camera distortion models. Moreover, it varies significantly according to the user's pupil positions due to *pupil swim* [15], [16], and to the varifocal position as well. The asymmetry, non-linearity, and pupil swim is especially pronounced for HMDs with off-axis combiners due to their large optical aberrations, as shown in Fig. 2.

All together, the projection from a world point $\mathbf{w}$ to a display pixel $\mathbf{d}$, subject to pupil position $\mathbf{p}$ and varifocal

- *Q. Guo is with Harvard University. H. Tang, A. Schmitz, W. Zhang, Y. Lou, A. Fix, S. Lovegrove, and H. M. Strasdat are with Facebook Inc. E-mail: qguo@seas.harvard.edu, {hxtang, aaronschmitz, roach, louy, alexander.fix, stevenlovegrove, strasdat}@fb.com*
- *This work was performed when Q. Guo interned at Facebook Reality Labs.*

978-1-7281-5230-1/20/$31.00 ©2020 IEEE



$\mathbf{p} = (-4, 0, 0)$ mm, f = 1mm | $\mathbf{p} = (-4, 0, 0)$ mm, f = 7mm | $\mathbf{p} = (4, 0, 0)$ mm, f = 1mm

Fig. 2: The display panel showing a rectangular checkerboard, viewed from the eyeball camera. The pattern is severely distorted due to the off-axis combiner. This distortion also varies with eyeball position $\mathbf{p}$ and varifocal position f, as highlighted in the closeup window.

position f is a 7D to 2D function

$$\mathbf{d} = \Pi(\mathbf{w}, \mathbf{p}, f) \qquad \mathbf{d} \in \mathbb{R}^2, \mathbf{w} \in \mathbb{R}^3, \mathbf{p} \in \mathbb{R}^3, f \in \mathbb{R} \qquad (1)$$

In theory, the above mapping can be computed from optical prescriptions of the HMD. But in practice, form and mounting errors are unavoidable. HMD components may be fabricated and assembled differently than the original prescriptions. Moreover, the display combiners often get deformed during mounting. Ignoring these errors would result in noticeable misalignment between the real and virtual world, causing serious user discomfort [17]. Therefore, *calibration* is necessary to measure the actual world-to-display projections.

We propose an automatic and efficient approach to AR system calibration with off-axis, see-through displays. The method generates a physics-based model for the HMD as in Fig. 1b. Accordingly, we can calibrate the world-to-display mapping using data captured at some pupil and varifocal position while predicting the mapping at other positions, by virtually translating the corresponding components, and redo ray tracing.

Although there is a big pile of HMD parameters to calibrate (listed in Table 1), we can estimate them in a few sub-steps. In each step, we fix parameters already determined in prior steps, and capture camera observations to estimate parameters of a few uncalibrated HMD components.

Many of the sub-steps involves a target, a camera and a few optical surfaces between them. We propose *Raycast calibration* to build a "white box" model for such an optical system. Specifically, it computes the relative poses (extrinsic parameters) among components, as well is how each component transports light (intrinsic parameters). Because such models have relatively small numbers of parameters, a small amount of data is sufficient to fit to such models, while being able to produce optical distortions of high complexity. Empirically, the low dimensionality of these models also allows calibration to converge quickly with a small search space. As we will show, this technique applies to a variety of calibration problems.

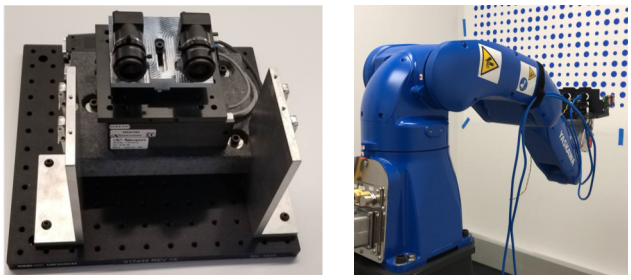
Related work

Early AR calibration approaches such as Single Point Active Alignment Method (SPAAM) and its variations [18], [19], [20] often require user interactions to calibrate the projection from world to display. Such methods are imprecise, prone to human errors, and places a heavy burden on users. Recent methods try to reduce the user's effort using cameras tracking eye gazes [21], [22], but they still leave calibration results user-dependent because they don't factor out variation in inter-pupil distance, eye relief, or even where the HMD sits on the user's face.

Previously, many AR calibration methods have exploited camera distortion models to represent the world to display mapping [23], [24]. However, distortions produced by off-axis curved combiners fits poorly to such models due to large amounts of non-linearity and asymmetry. Alternatively, non-parametric models such as 2D-to-2D polynomials [1], [2], [3] do not assume any underlying structure of the mapping. Yet calibration using such models require long data acquisition time, since it needs to scan pupil positions and varifocal positions in Eq. (1) exhaustively to capture the full 7D variation of the mapping. Fig. 1c shows a sample comparison between the proposed Raycast model and this polynomial model on calibrating the display and the combiner of the HMD in Fig. 1a. The Raycast model reduces the number of parameters ten times while still achieving sub-pixel reprojection accuracy. Our approach is in similar spirit to the calibration method of Project North Star [25], which also builds a physics-based model of the optical system. The key difference is that Project North Star implements an objective function which requires hardware in the loop and real-time measurements. Thus it is expensive to evaluate the derivative of the objective function which is required by most modern optimization techniques. In comparison, our approach decouples data acquisition and optimization completely.

2 System Calibration of an AR Headset

We consider two subsystems of the AR HMD prototype in Fig. 1b: a joint display-SLAM subsystem to render world-locked content, and an eye-tracking subsystem to track the pupil positions. In this paper we describe in detail the calibration of the former to illustrate our raycast calibration approach. In Sec. 6, we briefly describe eye tracking calibration as another example of our approach.



(a) (b)

Fig. 3: Calibration fixtures for data acquisition. (a) A linear translation stage with precalibrated eyeball cameras to mimic different pupil positions. It is used for multiview-varifocal calibration. (b) A robot arm that carries and moves the HMD to perform SLAM, display and see-through calibration.

2.1 Geometrical distortions in AR HMDs

The projection $\mathbf{d} = \Pi(\mathbf{w}, \mathbf{p}, f)$ of Eq. (1) maps from a 3D world point $\mathbf{w}$ to a 2D display pixel $\mathbf{d}$. We decoupled this projection into three parts (as shown in Fig. 1c): world-to-eye transformation, see-through distortion and display distortion.

First, real-world points $\mathbf{w}$ are usually reported by inside-out tracking as in some virtual "world" frame that does not move with HMD motion. We transform $\mathbf{w}$ to the eye frame

$$\mathbf{e} = \mathbf{T}_{\mathrm{E:C1}}\mathbf{T}_{\mathrm{C1:W}}\mathbf{w} \qquad \text{(world to eye)}. \tag{2}$$

Here we denote $\mathbf{T}_{\mathrm{C1:W}}$ the world-to-SLAM transformation and $\mathbf{T}_{\mathrm{E:C1}}$ the SLAM-to-eye transformation.[1]

Now consider projecting the 3D point $\mathbf{e}$ in the eye frame to a 2D pixel $\mathbf{x}$ on the retina. Note that users look at the point through the combiner. Light refracts at both combiner surfaces, and leads to a nonlinear projection of the points $\mathbf{e}$ we call *see-through distortion* Π_{S}:

$$\mathbf{x} = \Pi_{\mathrm{S}}(\mathbf{e}, \mathbf{p}) \qquad \text{(see-through distortions)}, \tag{3}$$

which depends on the pupil position $\mathbf{p}$. Finally, *display distortion* is the projection Π_{D} of the 2D retinal pixel $\mathbf{x}$ to the corresponding 2D display pixel $\mathbf{d}$:

$$\mathbf{d} = \Pi_{\mathrm{D}}(\mathbf{x}, \mathbf{p}, f) \qquad \text{(display distortions)}. \tag{4}$$

This distortion is caused by the aberrations on the reflection path of the combiner. Thus, it depends on the pupil position $\mathbf{p}$ and the varifocal position f. The goal of calibration is to model the SLAM-to-eye transformation $\mathbf{T}_{\mathrm{E:C1}}$, the see-through distortion Π_{S} and the display distortion Π_{D}.

To keep humans out of the loop, in calibration we use a pair of cameras near the nominal eye position to observe the display and the see-through distortions. We refer to these cameras as eyeball cameras. Therefore, for each eye, the eye frame E is the local frame of the eyeball camera, and the retinal pixels $\mathbf{x}$ are camera pixels. We use two automatic calibration fixtures for data acquisition (Fig. 3). The first is a translation stage that captures the display distortion Π_{D} at various pupil and varifocal positions. The second is a robot arm that acquires data for calibrating the SLAM, the see-through distortion Π_{S}, and the SLAM-to-eyeball transformation $\mathbf{T}_{\mathrm{E:C1}}$.

1. We denote $\mathbf{T}_{a:b}$ as the relative pose from coordinate frame b to frame a so that $\mathbf{e}_b = \mathbf{T}_{a:b}\mathbf{e}_b$. Without loss of generality, we assume the SLAM system to use the first camera C_1 as its parent frame.

(a) Fixture components

Component	Intrinsic	Extrinsic
Eyeball camera E (robot arm)	κ_{E}	$\mathbf{T}_{\mathrm{E:C1}}$
Eyeball camera E′ (trans. stage)	$\kappa_{\mathrm{E}'}$	$\mathbf{T}_{\mathrm{E}'(\mathbf{p}):\mathrm{S}_1}$
Calibu targets $(\mathrm{T}_1, \mathrm{T}_2, \dots)$		$\mathbf{T}_{\mathrm{C1:T1}}, \mathbf{T}_{\mathrm{C1:T2}}, \dots$

(b) HMD components

Component	Intrinsic	Extrinsic
SLAM $(\mathrm{C}_1, \mathrm{C}_2, \dots)$	$\kappa_{\mathrm{C1}}, \kappa_{\mathrm{C2}}, \dots$	$\mathbf{T}_{\mathrm{C1:C2}}, \dots$
Inner combiner S_1	prescription $(\rho_1, \eta_1, \alpha_1, \gamma_1)$ deformation β_1	$\mathbf{T}_{\mathrm{E:S}_1}$
Outer combiner S_2	prescription $(\rho_2, \eta_2, \alpha_2, \gamma_2)$ deformation $\boldsymbol{\beta}_2$	$\mathbf{T}_{\mathrm{S}_1:\mathrm{S}_2}$
Display D	-	$\mathbf{T}_{\mathrm{S}_1:\mathrm{D}(f)}$

TABLE 1: Parameters of HMD components. Note that the pose of the display is a function of varifocal position f, and the pose of the eyeball camera on the translation stage is a function of pupil position $\mathbf{p}$. Intrinsic parameters of eyeball cameras κ_{E}, $\kappa_{\mathrm{E}'}$ are calibrated ahead of time. Combiner prescriptions as well as the pose between its two surfaces $\mathbf{T}_{\mathrm{S}_1:\mathrm{S}_2}$ are held as constants during calibration.

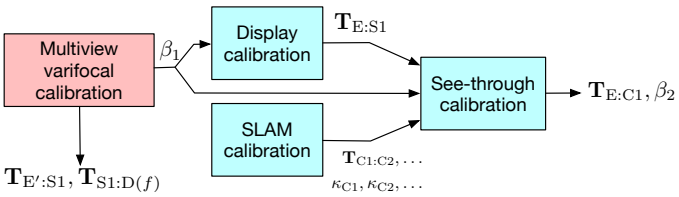


Fig. 4: Calibration workflow for the HMD prototype in Fig. 1a. The color of the blocks indicates the fixtures (see Fig. 3) for data acquisition. The output of all steps collectively gives model parameters in Table 1.

Although the see-through and display distortions are functions of pupil positions $\mathbf{p}$, they can be accurately modelled via the proposed method without using data captured at a dense set of pupil positions. In fact, we only use two pupil positions on the translation stage to capture display distortion and a single position on the robot arm to capture see-through distortion in our experiment. We achieve this by estimating a physics-based model of the HMD.

As Table 1 summarizes, we model the HMD and the fixture as a collection of optical components. Each component is parameterized by a set of intrinsic and extrinsic parameters. The intrinsic parameters represent how the component transports light, while the extrinsic parameters allow us to relate the local coordinate frames of the components with each other. Therefore, for example, moving the pupil position is as simple as changing the eyeball camera's extrinsic parameter $\mathbf{T}_{\mathrm{C1:E}}$.

2.2 System calibration workflow

Fig. 4 shows the proposed workflow to estimate all parameters in Table 1.

Multi-view varifocal calibration: In this step we estimate the display distortion Π_{D} in Eq. 4. Using the translation stage in Fig. 3a, we capture the distortion at two pupil

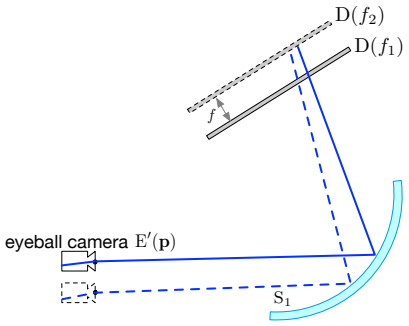


Fig. 5: Geometric model for multi-view varifocal calibration on translation stage (Fig. 3a). The model consists of the eyeball camera E′, the display D and the inner surface of the combiner S_1. The eyeball camera translates to different pupil positions $\mathbf{p}$ and the display translates linearly to different varifocal positions f.

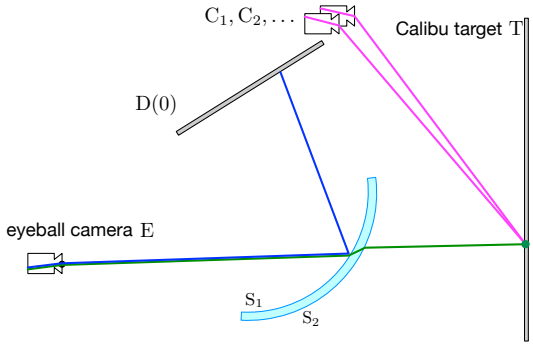


Fig. 6: Geometric model for display (blue), SLAM (pink) and see-through (green) calibration on robot arm (Fig. 3b). Light paths of the three calibration steps are visualized in different colors. Display calibration involves the fixed eyeball camera E, the inner combiner surface S_1 and the display D at $f = 0$. See-through calibration covers the eyeball camera E, both combiner surfaces S_1, S_2 and the Calibu target T [26].

positions. At each pupil position, we translate the display to two varifocal positions. The distortion is measured as the correspondence between eyeball camera pixels $\mathbf{x}$ and display pixels $\mathbf{d}$. We collect such correspondences by capturing eyeball camera images while a sequence of checkerboard is displayed (see Appendix A.1 for detailed approach). Fig. 5 shows the display distortion resulting from the combiner reflection. The goal of this calibration is to estimate intrinsics and extrinsics of the combiner's inner surface S_1, the eyeball camera $E'(\mathbf{p})$ and the display $D(f)$, for every pupil position $\mathbf{p}$ and display position f.

SLAM calibration: We calibrate the SLAM subsystem on the HMD using data captured on the robot arm. The fixture carries the HMD and moves it continuously. In the meantime, the SLAM cameras co-observe a few calibration targets and the IMU sensors in the SLAM system record accelerations of the HMD. We use Calibu targets [26], which are a grid of large and small dots in a planar surface. Each dot is uniquely identified by the sizes (large or small) of its neighboring dots, and thus allows us to efficiently establish correspondences.

In our implementation, we use the method by Lovegrove et al. [27] for SLAM calibration. SLAM calibration is a problem orthogonal to the focus of this paper and we omit the relevant discussion from now on.

See-through calibration: As Fig. 6 shows, the see-through light path (green) gets refracted twice on the combiner. Thus, the see-through distortion Π_S is determined by the intrinsics and extrinsics of the eyeball camera E, and both surfaces S_1, S_2 of the combiner.

To calibrate the see-through distortion, the robot arm is used to simultaneously move the HMD and the eyeball cameras along a certain trajectory. Meanwhile, the SLAM and eyeball cameras capture images of the Calibu targets T_1, T_2...,. Using the observations from SLAM cameras, we can localize the pose of each target relative to the SLAM frame $\mathbf{T}_{C1:T_j}$. This provides correspondences between points on the target in the SLAM coordinate $\mathbf{t}$ and pixels on the eyeball camera $\mathbf{x}$. The correspondences can be used to estimate the relative pose between the SLAM and the eyeball $\mathbf{T}_{C1:E}$, and unknown parameters of the surfaces S_1, S_2.

Display calibration: As mentioned above, the goal of this step is to calibrate the refraction by the inner combiner surface S_1. Since it is impossible to place calibration targets between the two combiner surfaces, it is hard to measure the refraction directly. However, we can estimate the shape of the inner combiner by capturing correspondences between eyeball pixels and display pixels (Fig. 6 blue). Given the shape of the inner combiner, rays can be traced from eyeball pixels to the inner combiner, and the surface normal at the intersection can be computed. Then, tracing along the see-through path only requires Snell's law, assuming known refractive index of the combiner material.

Comparing display calibration (Fig. 6 blue) and the multi-view varifocal calibration (Fig. 5), the only difference is that the former uses a fixed pupil position E and a fixed varifocal position $f = 0$. As shown in Fig. 4, this is because the display calibration reuses the intrinsics of the inner combiner S_1 calibrated from the multi-view varifocal calibration, which greatly reduces the parameter space to optimize.

After all calibration is done, all poses are transformed to a unified frame. For example, we can transform the extrinsic parameters of combiner surfaces and displays to be relative to the first SLAM camera C_1 by $\mathbf{T}_{C1:S_1} = \mathbf{T}_{C1:E}\mathbf{T}_{E:S_1}$, $\mathbf{T}_{C1:S_2} = \mathbf{T}_{C1:S_1}\mathbf{T}_{S_1:S_2}$, and $\mathbf{T}_{C1:D(f)} = \mathbf{T}_{C1:S_1}\mathbf{T}_{S_1:D(f)}$.

3 Modeling Geometric Distortion by the Combiner

The projection functions Π_D and Π_S formulated in Sec. 2 both involve modeling the geometric distortion caused by the combiner. In this section we describe how we characterize such distortions with a model of the optical system.

3.1 Optical model

Consider an optical system as shown in Fig. 7, in which a camera C looks at a target T through multiple refractive or reflective surfaces $S_1, S_2, \ldots S_n$.

Without loss of generality we specify the poses of all components as relative to the camera C. Thus the camera itself has the trivial pose $\mathbf{I}$ and is fully specified by its intrinsic κ.

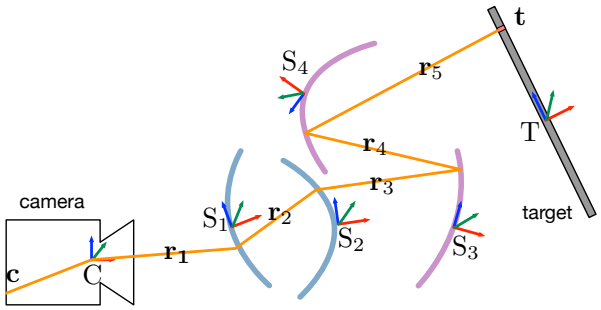


Fig. 7: Graphical illustration of the raycast model. Blue curved surfaces are refractive and purple curved surfaces are reflective.

The n-th lens surface is specified by the tuple $(\mathrm{z}_n, \gamma_n, \mathbf{T}_{\mathrm{C:S}n})$. With $\mathbf{T}_{\mathrm{C:S}n}$ defining the local frame of the surface, the 2D function $\mathrm{z}_n(x, y)$ defines the sag (*a.k.a.* shape) of the surface. The scalar γ denotes the ratio between the refractive index of the medium before and after the surface [2]. A target is treated as a special surface with sag $\mathrm{z} = 0$, with no reflection or refraction occurring on the surface.

We assume the sag function $\mathrm{z}_n(x, y)$ for the n-th lens surface to take the following parametric form

$$\mathrm{z}_n(x, y) = \mathrm{q}(x, y; \rho_n, \eta_n) + \sum_{p=1}^{P} \boldsymbol{\alpha}_{n,p} \mathrm{b}_p(x, y) + \sum_{q=1}^{Q} \boldsymbol{\beta}_{n,q} \mathrm{b}_q(x, y). \tag{5}$$

The first two terms comes from the optical prescription of the HMD (See Appendix A.2 for detailed definition). The third term models the combiner deformation. The second and third terms are represented with Zernike polynomial sequences [28], where $\mathrm{b}_p(x, y)$ and $\mathrm{b}_q(x, y)$ are Zernike polynomials. This formulation allows us to represent continuous sag functions in free forms.

Thus the optical system is fully specified by intrinsics and extrinsics of all components

$$\boldsymbol{\Theta} = \begin{pmatrix} \kappa, \rho_1, \eta_1, \boldsymbol{\alpha}_1, \boldsymbol{\beta}_1, \gamma_1, \mathbf{T}_{\mathrm{C:S1}}, \\ \rho_2, \eta_2, \boldsymbol{\alpha}_2, \boldsymbol{\beta}_2, \gamma_2, \mathbf{T}_{\mathrm{C:S2}}, \ldots \mathbf{T}_{\mathrm{C:T}} \end{pmatrix} \tag{6}$$

3.2 Geometric distortion model

The geometric distortion between the camera and the target is defined by the chief ray, the light path through the optical center of the camera's pupil. While tracing the chief ray from a target point to the camera is hard, inversely tracing the ray from camera to target can be easily done following laws of geometric optics.

Ray casting

We can define the mapping from a 2D camera pixel $\mathbf{c}$ to a 2D target point $\mathbf{t}$

$$\mathbf{t} = \mathrm{raycast}(\mathbf{c}; \boldsymbol{\Theta}) \tag{7}$$

by tracing the ray backwards from the camera to the target, as Figure 7 shows. We call this process *ray casting*.

2. Reflection is a special case with $\gamma_n = -1$.

We define ray casting as

$$\mathbf{r}_1 = \mathrm{unproject}(\mathbf{c}; \kappa) \quad \text{(camera unproject)} \tag{8}$$

$$\mathbf{r}_{n+1} = \mathrm{deflect}(\mathbf{r}_n; \mathrm{z}_n, \gamma_n, \mathbf{T}_{\mathrm{C:S}_n}) \quad \text{(ray deflection)} \tag{9}$$

$$\mathbf{t} = \mathrm{intersect}(\mathbf{r}_n; \mathbf{T}_{\mathrm{C:S}_n}) \quad \text{(plane intersection).} \tag{10}$$

First, we map from pixel $\mathbf{c}$ to the camera-side chief ray $\mathbf{r}_1$ according to the camera's intrinsic parameter κ. Then we trace the ray iteratively through N optical surfaces from $\mathbf{r}_1$ to target side ray $\mathbf{r}_{N+1}$. Finally we compute the target point $\mathbf{t}$ as the intersection between $\mathbf{r}_n$ and the target plane. This requires knowing the relative pose between the camera and the target.

A key operation here is the $\mathrm{deflect}(\cdot)$ function that transforms the chief ray from $\mathbf{r}_n$ to $\mathbf{r}_{n+1}$. At a high level, this is done by first finding the ray intersection with the surface and then computing ray refraction or reflection. We show in Appendix A.3 and A.4 how both operations can be efficiently performed with parametric optical surfaces as in Eq. (5). The final intersection function $\mathrm{intersect}(\cdot)$ is a special case of the deflection function, as it computes the ray intersection with a planar surface $\mathrm{z} = 0$.

Ray tracing

Tracing the ray from a target point $\mathbf{t}$ to a camera pixel $\mathbf{c}$ cannot be computed analytically in optical systems with free form surfaces. This is due to the difficulty in finding the chief ray for each target point that goes through the camera's aperture center. Thus, we compute the ray tracing by searching for the camera pixel $\mathbf{c}$ that minimizes reprojection error on the target plane, i.e.

$$\mathrm{raytrace}(\mathbf{t}; \boldsymbol{\Theta}) = \arg\min_{\mathbf{c}} |\mathrm{raycast}(\mathbf{c}; \boldsymbol{\Theta}) - \mathbf{t}|^2 \tag{11}$$

Visibility

For certain configurations, a camera pixel may not see any points on the target because the corresponding chief ray cannot hit one or more optical surfaces on its light path.

To quantify such visibility, we define

$$\mathrm{a}_n(\mathbf{c}) = \min_t |z_o + w \cdot t - \mathrm{z}_n(x_o + u \cdot t, y_o + v \cdot t)|^2 \tag{12}$$

as the nearest distance in z between the surface z_n and the ray $\mathbf{r}_n$ cast from camera pixel $\mathbf{c}$ in the coordinate of z_n. Here we denote the ray to have origin (x_o, y_o, z_o) and direction (u, v, w). So a camera pixel is considered visible to aperture n if $\mathrm{a}_n(\mathbf{c}) = 0$, and invisible otherwise.

4 Calibration Algorithms

The raycast model in Sec. 3 allows us to generate correspondences $\mathcal{D}$ between target points $\mathbf{t}$ and camera pixels $\mathbf{c}$ from a physics-based model $\boldsymbol{\Theta}$. But, our goal is to solve the inverse problem that estimates $\boldsymbol{\Theta}$ from data of correspondences. We call this process *raycast calibration*.

In this section, we first show raycast calibration with a fixed camera and target, and its application in display calibration. As will be discussed in Sec. 6, it can also be applied to eye-tracking calibration. Then, we extend the formulation to handle moving camera and targets, which is used for multiview varifocal calibration and see-through calibration.

4.1 Single-view raycast calibration

Consider a raycast model described in Sec. 3, with the camera and the target fixed. A set of noisy correspondences $\mathcal{D} = \{(\mathbf{c}_i, \mathbf{t}_i)\}$ between camera pixels $\mathbf{c}_i$ and target points $\mathbf{t}_i$ are acquired. The goal is to estimate the model $\mathbf{\Theta}$ that generates $\mathcal{D}$. Often it is not necessary to optimize all parameters in $\mathbf{\Theta}$, but a low dimension reparameterization θ of $\mathbf{\Theta}$. For example, for display calibration the intrinsic parameters of the inner combiner are already known, thus θ only contains extrinsic parameters.

The most straight-forward way to invert θ is to solve the following minimization problem:

$$\begin{aligned} \arg\min_{\theta} \sum_i |\text{raytrace}(\mathbf{t}_i, \mathbf{\Theta}(\theta)) - \mathbf{c}_i|^2 \quad &\text{(reprojection)} \\ \textbf{s.t. } \forall i, n : \text{a}_n(\mathbf{c}_i) = 0 \quad &\text{(visilibity)} \end{aligned} \tag{13}$$

The objective function effectively computes the maximum likelihood estimation (MLE) of θ assuming Gaussian noise on the image reprojection error. The non-linear constraints enforce that all target points in the data are visible in the estimated model. This is equivalent to restricting chief rays to pass through the aperture of all optical surfaces.

Two challenges arise in the above formulation: First, ray tracing is computationally expensive as indicated in Sec. 3. Empirically, ray tracing is $10\times$ computationally expensive than ray casting. To solve this problem, we instead evaluate errors on the target plane, rather than on the image plane. This modification significantly improves the efficiency of optimization, though the original ray tracing statistically gives the Maximum-Likihood estimation of θ under a Gaussian noise assumption [3]. Second, the visibility constraint defines a complicated, non-convex feasible region for the problem, making the optimization difficult to solve. So we relax Eq. (13) into an unconstrained optimization problem:

$$\arg\min_{\theta} \sum_i |\varepsilon(\mathbf{c}_i, \mathbf{t}_i; \mathbf{\Theta})|^2 + \sum_{n,i} |\text{a}_n(\mathbf{c}_i; \mathbf{\Theta})|^2 \tag{14}$$

where the reprojection error ε takes into account both visible and invisible pixels

$$\varepsilon(\mathbf{c}, \mathbf{t}; \mathbf{\Theta}) = \begin{cases} \text{raycast}(\mathbf{c}, \mathbf{\Theta}) - \mathbf{t} & \text{if } \forall n : \text{a}_n(\mathbf{c}, \mathbf{\Theta}) = 0 \\ \tau & \text{otherwise} \end{cases}. \tag{15}$$

The constant τ penalizes invisible pixels. In our implementation we set $\tau = 10$, but we find the optimization is insensitive to τ as the converged θ always allows all target points in $\mathcal{D}$ to be visible.

This form of unconstrained optimization can be solved with existing optimization packages. In our implementation, we use the Ceres Solver [29], an open source toolbox in C++ for optimization problems. We use Ceres to perform auto-differentiation for the loss function, and solve the optimization with the Levenberg-Marquart algorithm. The optimization usually converges in fewer than 10 iterations.

Display calibration

The single-view raycast calibration can be applied to display calibration. Recall the light path of display calibration (Fig. 6 blue), where the display $\text{D}(0)$ and the eyeball camera E correspond to T and C respectively in Fig. 7. The optical model is:

$$\mathbf{\Theta} = (\kappa_\text{E}, \rho_1, \eta_1, \boldsymbol{\alpha}_1, \boldsymbol{\beta}_1, \gamma_1, \mathbf{T}_{\text{E:S}_1}, \mathbf{T}_{\text{E:D}(0)}). \tag{16}$$

Multiple parameters in $\mathbf{\Theta}$ are already known. The eyeball camera intrinsic κ_E is pre-calibrated. The combiner sag z_1 is determined by multiview varifocal calibration. The refractive index at the surface is $\gamma = -1$ since the surface is reflective. Therefore, the parameters to be calibrated are

$$\theta = (\mathbf{T}_{\text{E:S}_1}, \mathbf{T}_{\text{S}_1\text{:D}(0)}). \tag{17}$$

In our implementation, we represent each transformation $\mathbf{T}$ using a vector in $\mathbb{R}^6$, which is a parameterization in the tangential space of the **SE(3)** Lie group.

4.2 Multi-view raycast calibration

Consider the camera and the target in Fig. 7 moving to different positions from time to time. The optical system can then be represented as a list of optical models with different camera and target positions, $(\mathbf{\Theta}_1, \mathbf{\Theta}_2, \dots)$. At "time" t, the system is described by $\mathbf{\Theta}_t$, and a set of pixel correspondences $\mathcal{D}_t = \{(\mathbf{c}_{t,i}, \mathbf{t}_{t,i})\}$ is acquired.

The loss function extends Eq. (14) by summing the error terms over all views:

$$\arg\min_{\theta} \sum_{t,i} |\varepsilon(\mathbf{c}_{t,i}, \mathbf{t}_{t,i}; \mathbf{\Theta}_t)|^2 + \sum_{t,i,n} |\text{a}_n(\mathbf{c}_{t,i}, \mathbf{\Theta}_t)|^2 \tag{18}$$

It may appear that a multi-view configuration has many variables to optimize over. In practice, however, most parameters are shared across configurations. The number of parameters in θ may even be sublinear to the number of optical models. We present two examples below.

Multiview Varifocal calibration

The problem is similar to display calibration, with the eyeball camera E' and the display D (Fig. 5) corresponding to T and C in Fig. 7 respectively. But here, the eyeball camera E' moves to K positions $\{\mathbf{p}_1, ..., \mathbf{p}_K\}$ and the display D translates to L positions $\{f_1, ..., f_L\}$. This generates $K \times L$ views, each with different relative poses $\mathbf{T}_{\text{E}'\text{:S}_1}$ and $\mathbf{T}_{\text{E}'\text{:D}}$:

$$\mathbf{\Theta}_{k,l} = (\kappa_\text{E}, \rho_1, \eta_1, \boldsymbol{\alpha}_1, \boldsymbol{\beta}_1, \gamma_1, \mathbf{T}_{\text{E}'(\mathbf{p}_k)\text{:S}_1}, \mathbf{T}_{\text{E}'(\mathbf{p}_k)\text{:D}(f_l)}). \tag{19}$$

On the other hand, note that

$$\underbrace{\mathbf{T}_{\text{E}'(\mathbf{p}_k)\text{:D}(f_l)}}_{K\times L \text{ poses}} = \underbrace{\mathbf{T}_{\text{E}'(\mathbf{p}_k)\text{:S}_1}}_{K \text{ poses}} \underbrace{\mathbf{T}_{\text{S}_1\text{:D}(f_l)}}_{L \text{ poses}}, \tag{20}$$

thus we can fully specify all $K \times L$ configurations with $K + L$ extrinsic parameters. In addition, the sag of the inner combiner S_1 is shared among all $\mathbf{\Theta}_{k,l}$. Among parameters of the sag $(\rho_1, \eta_1, \boldsymbol{\alpha}_1, \boldsymbol{\beta}_1, \gamma_1)$, only the distortion coefficients $\boldsymbol{\beta}_1$ need to be estimated, while others are fixed to the original prescription of the HMD prototype. The refractive index γ_1 is set to -1 as the ray is reflected by S_1. According to the definition in Eq. (5), the first Zernike term $\beta_{1,1}\text{b}_1(x, y)$ is adding a constant offset to the sag, and is equivalent to a translation to the combiner. Thus in our experiment, we fix $\beta_{1,1} = 0$ to reduce redundancy in parameterization. The total dimension of parameters to be estimated is $6(K + L) + Q - 1$:

3. We tried re-weighting the errors using precision matrices, and the difference in results is negligible.

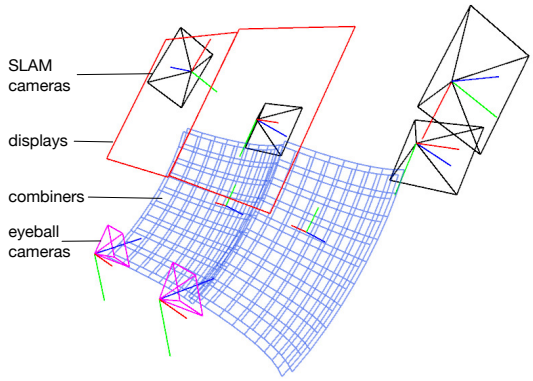


Fig. 8: 3D visualization of the calibrated model of the HMD prototype in Fig. 1a.

$$\theta = \begin{pmatrix} \beta_{1,2} \dots \beta_{1,Q}, \\ \mathbf{T}_{\mathrm{E}'(\mathbf{p}_1):\mathrm{S}_1} \dots \mathbf{T}_{\mathrm{E}'(\mathbf{p}_K):\mathrm{S}_1}, \\ \mathbf{T}_{\mathrm{S}_1:\mathrm{D}(f_1)} \dots \mathbf{T}_{\mathrm{S}_1:\mathrm{D}(f_L)} \end{pmatrix} \tag{21}$$

See-through calibration

In see-through calibration (Fig. 6 green), the eyeball camera E corresponds to the camera C in Fig. 7. The ray undergoes two refractions by the inner and outer combiners S_1 and S_2, thus the model contains two optical surfaces. In practice, there are multiple calibration targets $\mathrm{T}_1, \mathrm{T}_2, \dots$. In this case, we define a separate optical model $\boldsymbol{\Theta}_j$ for each target T_j:

$$\boldsymbol{\Theta}_j = \begin{pmatrix} \kappa_{\mathrm{E}}, \rho_1, \eta_1, \boldsymbol{\alpha}_1, \boldsymbol{\beta}_1, \gamma_1, \mathbf{T}_{\mathrm{E:S}_1}, \\ \rho_2, \eta_2, \boldsymbol{\alpha}_2, \boldsymbol{\beta}_2, \gamma_2, \mathbf{T}_{\mathrm{E:S}_2}, \mathbf{T}_{\mathrm{E:T}_j} \end{pmatrix}. \tag{22}$$

Here the intrinsic of the eyeball camera κ_{E} is known, and the sag parameters are shared across all $\boldsymbol{\Theta}_j$. The intrinsics of the inner combiner $(\rho_1, \eta_1, \boldsymbol{\alpha}_1, \boldsymbol{\beta}_1)$ are obtained from the multi-view varifocal calibration. Its extrinsics $\mathbf{T}_{\mathrm{E:S}_1}$ is determined by the display calibration. As the relative pose between the two combiner surfaces $\mathbf{T}_{\mathrm{S}_1:\mathrm{S}_2}$ is fixed, $\mathbf{T}_{\mathrm{E:S}_2}$ is also known. For intrinsics of the outer combiner $(\rho_2, \eta_2, \boldsymbol{\alpha}_2, \boldsymbol{\beta}_2)$, only the Zernike coefficients for distortion $(\beta_{2,2} \dots \beta_{2,Q})$ needs to be estimated in this step. The refractive indices γ_1, γ_2 are found from the material property. Therefore, unknown parameters in the optical model $\boldsymbol{\Theta}_j$ are $(\beta_{2,2} \dots \beta_{2,Q}, \mathbf{T}_{\mathrm{E:T}_j})$.

The pose of the target T_j in the eyeball frame $\mathbf{T}_{\mathrm{E:T}j}$ can be factored into two relative poses:

$$\mathbf{T}_{\mathrm{E:T}j} = \mathbf{T}_{\mathrm{E:C1}} \mathbf{T}_{\mathrm{C1:T}j}. \tag{23}$$

Using SLAM observations, we can separately estimate the target's pose relative to the SLAM system $\mathbf{T}_{\mathrm{C1:T}j}$. Thus see-through calibration contains many optical models $\boldsymbol{\Theta}_j$, the only unknown is the relative pose between the eyeball and the SLAM $\mathbf{T}_{\mathrm{E:C1}}$.

The set of parameters to be estimated is

$$\theta = (\boldsymbol{\beta}_{2,2}, \dots \boldsymbol{\beta}_{2,Q}, \mathbf{T}_{\mathrm{E:C1}}), \tag{24}$$

which has $Q + 5$ unknowns. As we will show in Sec. 5.1.3, experimentally it is realistic to assume the deformation on both optical surfaces is identical, i.e. $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2$. In this case we only need to estimate a single pose $\theta = (\mathbf{T}_{\mathrm{E:C1}})$ with 6 unknowns.

	Data acquisition	Optimization
Multiview varifocal	6min	8min
SLAM	5min	5min
Display	1.5min	20sec
See-through	2min	30sec

TABLE 2: Calibration time breakdown. In our experiments, calibrating the varifocal component of the same AR display using polynomial model takes several hours because data capture needs to densely scan the eyebox, and much more parameters are trained.

5 Results

In this section, we first present calibration results of the HMD prototype in Fig. 1a using raycast calibration, to demonstrate its effectiveness in dealing with realistic manufacture and mounting errors. Then, we show several simulations that quantitatively analyze our approach at each individual step.

We quantify calibration accuracy by reprojection errors in the camera space. Given a pair of correspondences $(\mathbf{c}, \mathbf{t})$ and calibrated parameters $\boldsymbol{\Theta}$, we can compute reprojected eyeball camera pixels as $\bar{\mathbf{c}} = \mathrm{raytrace}(\mathbf{t}, \boldsymbol{\Theta})$ and compute the reprojection error as $|\bar{\mathbf{c}} - \mathbf{c}|$. This metric depends on the intrinsic parameters of the eyeball camera. Thus, we also report the angular difference between the corresponding chief rays. Specifically we use $\mathrm{unproject}(\cdot)$ to map from the camera pixel $\mathbf{c}$ to the ray $\mathbf{r}_1$ into the camera. Let $\bar{\mathbf{v}}$ and $\mathbf{v}$ be the direction of the rays. The angular error is measured as $\arccos(\bar{\mathbf{v}} \cdot \mathbf{v})$. In this section we report calibration results of a single eye. Not surprisingly, in experiments we observe similar accuracy between the two eye cups due to symmetry of the prototype.

5.1 System calibration of an AR HMD

Following the pipeline in Fig. 4, we create an automatic process to systematically calibrate the prototype in Fig. 1a using raycast calibration. The whole process takes 30 minutes in total, including data acquisition and optimization. Fig. 8 shows the calibrated model and Table 2 shows the breakdown of calibration time.

5.1.1 Multi-view varifocal calibration

We acquire correspondences at 25 pupil positions and 2 varifocal positions. The pupil positions are uniformly sampled in the range of $[-4, 4] \times [-4, 4]$ mm eye box, and the varifocal positions are sampled at 1mm and 7mm. We use the correspondences at $\mathbf{p} = (\pm 2, 0)$mm with both varifocal positions for calibration and the other positions for model selection and validation. To collect this data, we mount the eyeball camera on a two-axis translation stage to move it horizontally or vertically. The prototype is mounted statically on the fixture. When the stage is homed, the eyeball camera sits approximately at the center of the designed eyebox $\mathbf{p} = (0, 0)$ mm. Note that the use of high-precision two-axis translation stage is only to evaluate the accuracy of the calibration exhaustively at different pupil positions inside the eyebox. As described above, we only use two pupil positions for calibration.

Fig. 9 reports the reprojection error of the raycast calibration using different numbers Q of the Zernike sequence in Eq. 5

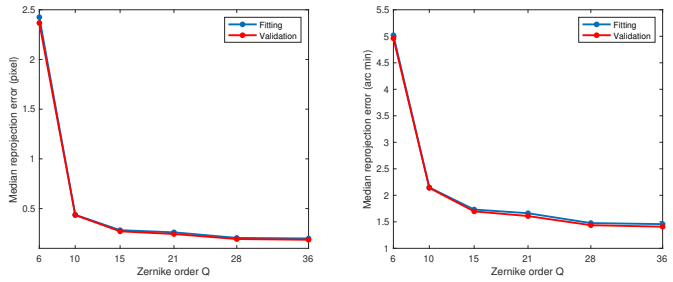


Fig. 9: Median reprojection error for multiview varifocal calibration with different Zernike orders Q. Both training and validating error decreases slowly after $Q = 28$. Note that we select $Q = N(N+1)/2, N = 1, 2, 3, ...$ based on the definition of the Zernike polynomials [28].

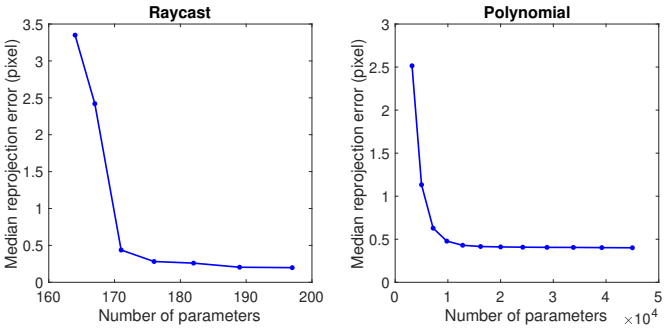


Fig. 10: Median reprojection error for raycast and polynomial models as a function of model parameters. The evaluation is performed on all $K = 25$ eyeball camera positions and $L = 2$ varifocal positions. The raycast model requires significantly fewer number of parameters than polynomial model but achieves smaller errors.

to model the combiner distortion. Both the training and validating RMSE of reprojection error decreases as Q increases. Although overfitting is elusive, the error decrease becomes negligible when the Zernike order Q is 28 or more, at about 0.25 pixels (1.5 arc minutes) for validating error.

Compared to the polynomial models used to calibrate many state-of-the-art AR HMD, e.g. [3], we observe the proposed raycast model requires many fewer parameters to achieve a similar accuracy on the correspondences of 25 pupil positions and 2 varifocal positions. As the polynomial model is not taking pupil and varifocal positions as input, a separate set of parameters needs to be estimated for every possible pupil and varifocal position. Therefore to account for distortions at K pupil positions and L varifocal settings, we need to create $K \times L$ different Qth-order polynomial models, each with $4(Q+1)^2$ parameters. In comparison, the raycast model with Zernike order Q requires $6(K+L)+Q-1$ parameters.

5.1.2 Display calibration

In display calibration, the prototype is mounted onto the robot arm. We reuse the estimated combiner distortion $\boldsymbol{\beta}_1$ from the multi-view varifocal calibration and only optimize the extrinsic parameters in Eq. (17). We achieve a median fitting error of 0.28 pixels (equivalently, about 1.16 arc minutes) in display calibration[4]. More than 90% of the pixels are under a reprojection error of 0.64 pixels (1.68 arc minutes). As an ablation study, if we use the combiner sag based on the optical prescription (i.e. $\boldsymbol{\beta}_1 = \mathbf{0}$), the reprojection error increases to 1.48 pixels (equivalently, about 2.66 arc minutes), and the 90% percentile of reprojection error goes to 3.19 pixels.

5.1.3 See-through calibration

We run see-through using 12 discrete image frames, each with a different HMD-to-target pose. We have tried to run see-through calibration in two alternative modes: either to keep the lens deformation as a constant i.e. $\boldsymbol{\beta}_2 = \boldsymbol{\beta}_1$, or use $\boldsymbol{\beta}_1$ as an initialization to $\boldsymbol{\beta}_2$, and jointly optimize $\boldsymbol{\beta}_2$ with the extrinsic parameter $\mathbf{T}_{\text{E:C1}}$. Both experiments result in a median reprojection error of about 0.52 pixels (1.58 arc minutes). The difference of error between the two modes are in the scale of 0.01 pixels.

5.2 End-to-end verification

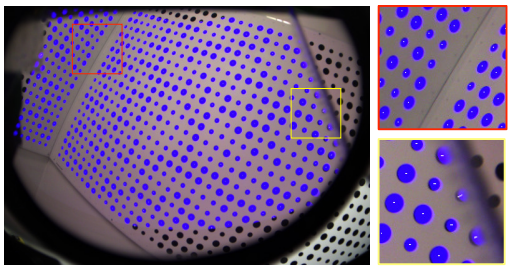
See-through calibration provides an anchor between the display and the inside-out tracking. In Fig. 11a, we show a joint verification of the display, SLAM and see-through calibration. In this test, the eyeball camera observes multiple Calibu targets through the combiner. Meanwhile, the targets are observed and localized by SLAM cameras. Using the optical model estimated from the display and see-through calibration, the display is able to show virtual Calibu patterns to overlay on the real ones. As shown in Fig. 11, the virtual and real images align precisely in majority of the field of view. Quantitatively, we observe more than 90% of the dots to have a misalignment error less than 1.0 pixel with median alignment error to be about 0.5 pixels. We observe small misalignment near the edge of the combiner due to abrupt deformations caused by mounting error which cannot be captured by the Zernike lens deformation model.

Fig. 11b shows a more interesting example where the display shows a virtual Ramesses statue (3D mesh simplified from that of [30]) on the desk. This is done by (1) using SLAM to localize the position of the desk relative to the HMD, (2) generating the virtual object in eyeball's local frame so that its relative pose to the desk is locked (3) projecting the virtual object to the display panels to determine the rendered image.

5.3 Simulated experiments

We individually evaluate the multi-view varifocal, display and see-through calibration using data generated by Monte Carlo simulation. For each calibration step, we generate 100 perturbed models of the HMD prototype $\boldsymbol{\Theta}_1, ...\boldsymbol{\Theta}_{100}$ by simulating form and mounting errors from the original optical prescription $\boldsymbol{\Theta}_o$. Specifically, we add a zero-mean Gaussian perturbation to each dimension of the translation and rotation. Standard deviation of the perturbation are 2 mm for translations and 2 degrees for rotations. In multi-view varifocal calibration, we also simulate combiner deformation. To do so, we perturb the combiner sag z_1 with zero-mean Gaussian with a standard deviation of 0.5 mm and fitting it to a Zernike model of 28 orders. We use this Zernike model as the ground-truth deformation.

4. The eyeball cameras on the robot arm uses a different lens. Therefore the same angular error corresponds to a larger pixel error



(a)

(b)

Fig. 11: Joint verification of display, SLAM and see-through calibration. (a) The HMD prototype observes the Calibu target, and uses the calibrated model to render a virtual pattern (blue) to overlay on top of see-through image (black). The rendering does not cover the whole field-of-view due to limited size of the display. (b) A virtual Ramesses statue [30] placed on top of a real desk, rendered using the calibrated HMD.

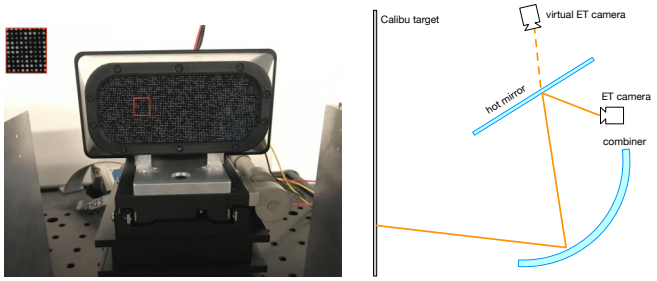
Using each model $\mathbf{\Theta}_i$, we simulate noisy correspondences between the camera pixels and the target points. The standard deviation of the noise is 0.5 pixels. For multi-view varifocal calibration, we generate correspondences for two camera positions and two targets positions. For the other two, we create correspondences for a single camera position and a single target position. This results in approximately 200000 pixel correspondences for multi-view varifocal across four image frames, 50000 for display calibration, and 10000 for see-through calibration.

First we evaluate the accuracy of the calibration algorithms. For each calibration step, we perform the corresponding Raycast calibration using the simulated correspondences generated by $\mathbf{\Theta}_1 \dots \mathbf{\Theta}_{100}$, and initialize the model using $\mathbf{\Theta}_o$. Table 3 reports the mean and standard deviation of the difference between the converged model and the true model $\mathbf{\Theta}_i$. The difference is very small.

To examine the robustness of the calibration steps to initialization, we use the optical prescription $\mathbf{\Theta}_o$ as the ground-truth to generate noisy correspondences. For each calibration step, raycast calibration is repeated 100 times starting from each $\mathbf{\Theta}_i$. We observe the standard deviation of the converged parameters to be below 1um in translation and 0.1 arc minutes in rotation.

		translation (mm)		rotation (arc min)	
		mean	stdv	mean	stdv
Multi-view varifocal	z_1	0.013	0.011	-	-
Display	$\mathbf{T}_{E:S_1}$	0.015	0.005	0.37	0.21
	$\mathbf{T}_{S_1:D}$	0.010	0.004	0.31	0.20
See-through	$\mathbf{T}_{E:C1}$	0.152	0.083	0.24	0.12

TABLE 3: Quantitative evaluation of each calibration step using simulated data. We report the mean and the standard deviation of calibration errors of 100 trials. For each trial, the correspondences are generated from a randomly perturbed model to simulate form and mounting noises. The sag error in multi-view varifocal is obtained by uniformly sampling 3D points on the sag, and computing their distance to the ground-truth sag in the z direction of the sag's local coordinate.



(a) (b)

Fig. 12: Fixture and model for calibrating eye-tracking camera. (a) The fixture is a static backlit target with a tiny Calibu pattern on it. The target is placed at the depth of nominal eye position. (b) By substituting the eye-tracking camera and the hot mirror with an equivalent virtual camera, eye-tracking calibration shares a common geometric model with display calibration.

6 Additional Example: Eye Tracking Calibration

As shown in Fig. 1b, the HMD prototype is also equipped with a glint-based eye tracker. The eye-tracker estimates the pupil positions of the user, so that the display properly show virtual objects according to the world-to-display projection at these pupil positions. The eye tracker uses an LED ring to shoot light onto the cornea, and an eye-tracking camera to see the reflection through the combiner and a hot mirror. The eye-tracking calibration both localizes the LED lights and estimates the mapping between pixels on the eye-tracking camera and 3D world points near the eyeball.

We use the translation stage in Fig. 3a to localize the LED lights right after multi-view varifocal calibration. By capturing images of the LED lights from different camera positions, we use bundle adjustment to estimate the position of the LED lights with respect to the eyeball camera. The reprojection error is just 0.17 pixels (Fig. 13a).

To estimate the mapping between eye-tracking camera and 3D world points we use the fixture of Fig. 12a. It is a static planar target with Calibu pattern that sits at the nominal eye positions. Interestingly, its raycast model is similar to that of the display calibration – a camera looks at a planar target via the reflection of the combiner (Fig. 12b). Here the

camera is the virtual image of the eye-tracking camera in the hot mirror. The eye-tracking camera is not preliminarily calibrated, thus we jointly estimate both the camera intrinsic parameters and relative pose between the mirror image of the eye-tracking camera, the combiner and the target. As Fig. 13b shows, raycast calibration achieves a reprojection error of 0.2 pixels.

7 Conclusion

We propose Raycast calibration, an efficient approach to calibrate AR HMDs with off-axis reflective combiners. The method accounts for highly non-linear geometric distortions caused by the combiner optics. Using a physics-based model, our approach only requires a small amount of data and can capture the full variation of display distortions under pupil swim and varifocal change.

The raycast calibration is a general framework that can be potentially extended beyond combiner-based AR HMDs. As we have shown in Section 4, with small modifications it can be used for calibrating other optical systems of a different architecture or using non-refractive/reflective optical elements.

Admittedly, raycast calibration may have inherent ambiguities that cannot be resolved. For example, if the curved combiner is perfectly spherical, its extrinsic parameters could be ambiguous for display calibration. In practice, we deal with such issue by carefully designing the calibration fixture to acquire data that further constrain the problems. Another approach is to predetermine certain parameters in the model as the prescription. Such choices are often specific to the optical design of the HMD.

Appendix

A.1 Collecting correspondences for display distortions

As Fig. 2 shows, the combiner significantly distorts the display image. Therefore, it is challenging to use conventional grid detection algorithms to establish correspondences.

In our implementation, we exploit the idea of Gray-code for efficiently acquiring display-eyeball correspondences. We render a sequence of checkerboard patterns from coarse to fine, and record the corresponding eyeball camera images. At each checkerboard resolution, we render a pattern of the resolution and its inverse pattern. Thus we can compare the intensity of each display and camera pixels between the two frames, to assign each a pixel a binary code. Putting binary codes across all resolution together, we get a Gray encoding for each display pixel and each camera pixel. Then for each gray code, we take the centroid of the corresponding display and camera pixels to generate a correspondence $(\mathbf{d}, \mathbf{x})$.

A.2 Combined Quadric-Zernike model for free-form optical surfaces

Both quadric and Zernike models in Eq. 5 are standard models in optics to represent lens surface [31].

The quadric base shape is formulated as

$$\mathrm{q}(x, y; \rho, \eta) = \frac{\rho(x^2 + y^2)}{(1 + \sqrt{1 - (1+\eta)\rho^2(x^2+y^2)})} \tag{25}$$

where (x, y) is a point on aperture, ρ controls the curvature of the shape and η controls is conic constant to control the eccentricity of the shape.

The Zernike offset $\sum_{p=1}^{n} \boldsymbol{\alpha}_p \mathrm{b}_p(x, y)$ sum a sequence of orthogonal basis b_p on a unit disk [28]. Similar to polynomial, the Zernike sequence provides an universal approximator to any 2.5-D continuous surface to represent a free-form surface.

A.3 Computing ray intersection with parametric surfaces

In this appendix we discuss how we compute the intersection between a ray with origin (x_o, y_o, z_o) and direction (u, v, w) with a parametric surface $z(x, y)$. We assume the ray and the surface are expressed in the same local coordinate frame. We also assume that we can compute partial derivatives $\partial z/\partial x, \partial z/\partial y$ at any point (x, y) within the surface aperture. This is true for the quadric-Zernike surface described in this paper but also applies to other common forms of optical surface prescriptions.

To start with, we parameterize all points on the ray as

$$(x_o + u \cdot t, y_o + v \cdot t, z_o + w \cdot t) \tag{26}$$

and define the distance between the point and the surface in the z-axis as

$$d_z(t) = z_o + w \cdot t - z(x_o + u \cdot t, y_o + v \cdot t) \tag{27}$$

Note that we can analytically $\partial d_z/\partial t$ by

$$\frac{\partial d_z(t)}{\partial t} = 1 - \frac{\partial z}{\partial x} \cdot u - \frac{\partial z}{\partial y} \cdot v \tag{28}$$

where the partial derivatives $\partial z/\partial x, \partial z/\partial y$ are evaluated at $(x_o + u \cdot t, y_o + v \cdot t)$.

We solve ray intersection by finding the distance t so that $d_z(t) = 0$. For special parametric forms such as plane and spheres, the problem has analytical solutions and are cheap to compute. For general surface, we use Newton's iteration to find the solution. In our implementation the iteration converges quickly in $2 - 3$ iterations and therefore it is practical to be embedded ray intersections in objective functions for optimization.

A.4 Light deflections at optical surfaces

We implement ray reflection and refraction following Snell's law. For completeness of the paper, the appendix reviews the method, which has been used in computer graphics for decades.

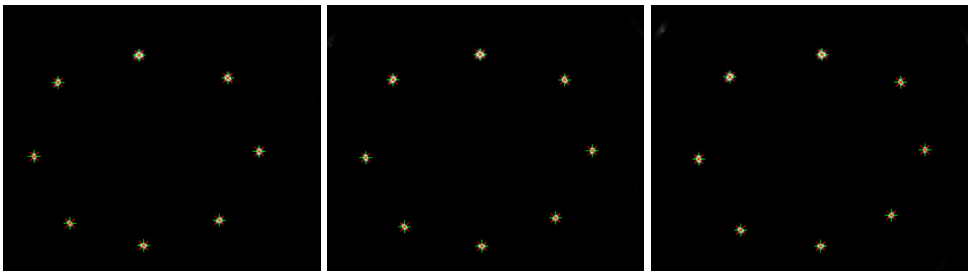
Given an incident ray and a surface, we can compute their intersection using the approach of A.3 and then compute the surface normal at the ray intersection. This defines the origin of the deflected ray.

As shown in Fig. 14 we can compute the direction of the deflected ray following the corresponding physical laws. Specifically, the incident angle is defined as

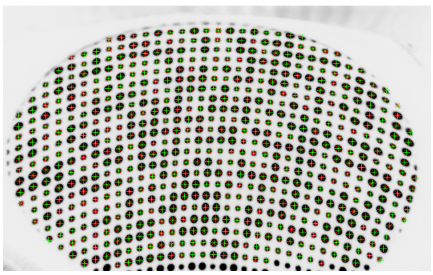
$$\theta_i = \arccos(\mathbf{n} \cdot \mathbf{r}) \tag{29}$$

By law of refraction, the exit angle θ_o satisfies

$$\sin(\theta_i)\bar{\gamma}_i = \sin(\theta_o)\bar{\gamma}_o \tag{30}$$

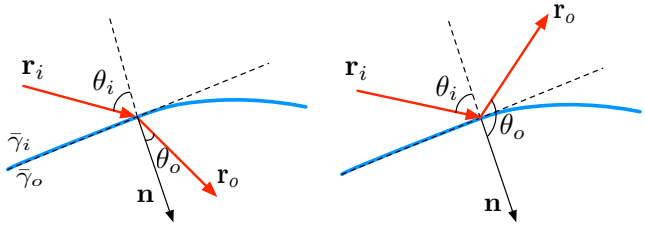


(a) reprojection error on LED calibration from three different views.



(b) reprojection error on ET calibration

Fig. 13: Reprojection errors on LED and eye-tracking camera calibration. Red crosses: image correspondences collected by LED and Calibu detectors. Green crosses: reprojected image pixels from the calibration model.



(a) refraction (b) reflection

Fig. 14: Light deflection at an optical surface.

Therefore by defining relative refractive index at the surface

$$\gamma = \bar{\gamma}_i / \bar{\gamma}_o \tag{31}$$

We can compute

$$\theta_o = \arcsin(\sin(\theta_i)\gamma) \tag{32}$$

And the refracted ray direction

$$\mathbf{r}_o = \gamma \mathbf{r}_i + (\cos(\theta_o) - \gamma \cos(\theta_i))\mathbf{n} \tag{33}$$

Note that we can consider light reflection as a special case of refraction by defining $\gamma = -1$ in the case of reflection:

$$\mathbf{r}_o = -\mathbf{r}_i + 2\cos(\theta_i)\mathbf{n} \tag{34}$$

Acknowledgments

The authors thank colleagues at FRL/Facebook who led the design, fabrication and bring-up of the prototype. We would also like to thank Zahid Hossain, Logan Wan, Jesus Briales, Yjing Fu, Jason Sensibaugh and Byron Taylor for their contribution to this research project.

References

[1] M. Klemm, F. Seebacher, and H. Hoppe, "Non-parametric camera-based calibration of optical see-through glasses for ar applications," in *2016 International Conference on Cyberworlds*, Sep. 2016, pp. 33–40. 1, 2

[2] Y. Itoh and G. Klinker, "Light-field correction for spatial calibration of optical see-through head-mounted displays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 4, pp. 471–480, April 2015. 1, 2

[3] J. Kim, Y. Jeong, M. Stengel, K. Akşit, R. Albert, B. Boudaoud, T. Greer, J. Kim, W. Lopes, Z. Majercik *et al.*, "Foveated ar: dynamically-foveated augmented reality display, supplementary material," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, p. 99, 2019. 1, 2, 8

[4] O. Bimber and R. Raskar, *Spatial Augmented Reality: Merging Real and Virtual Worlds*. Natick, MA, USA: A. K. Peters, Ltd., 2005. 1

[5] Z. Zheng, X. Liu, H. Li, and L. xu, "Design and fabrication of an off-axis see-through head-mounted display with an x-y polynomial surface," *Applied optics*, vol. 49, pp. 3661–8, 07 2010. 1

[6] L. Wei, Y. Li, J. Jing, L. Feng, and J. Zhou, "Design and fabrication of a compact off-axis see-through head-mounted display using a freeform surface," *Optics Express*, vol. 26, p. 8550, 04 2018. 1

[7] Meta View, Inc., "Meta: M vision," 2019. [Online]. Available: http://www.metavision.com 1

[8] Leap Motion, Inc., "Project north star: Mechanical," 2019. [Online]. Available: https://developer.leapmotion.com/northstar 1

[9] Mira Labs, Inc., "Mira augmented reality," 2019. [Online]. Available: https://www.mirareality.com/ 1

[10] Reality8, Inc., "Realmax 100 product information," 2019. [Online]. Available: http://realmaxinc.com/realmax-100-product-information/ 1

[11] DreamWorld, Inc., "Dreamworld ar," 2019. [Online]. Available: https://www.dreamworldvision.com/ 1

[12] "From the lab to the living room: The story behind facebook's oculus insight technology and a new era of consumer vr," 2019. [Online]. Available: https://tech.fb.com/the-story-behind-oculus-insight-technology/ 1

[13] D. Bohn, "Slamdance: inside the weird virtual reality of google's project tango," 2015. [Online]. Available: https://www.theverge.com/a/sundars-google/project-tango-google-io-2015 1

[14] K. G, "Resolving the vergence-accommodation conflict in head-mounted displays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 7, pp. 1912–1931, July 2016. 1

[15] G. Fry and W. W. Hill, "The center of rotation of the eye," *American Journal of Optometry and Archives of American Academy of Optometry*, 1962. 1

[16] J. M. Cobb, D. Kessler, and J. A. Agostinelli, "Optical design of a monocentric autostereoscopic immersive display," in *International Optical Design Conference*. Optical Society of America, 2002, p. IMB5. 1

[17] F. L. Kooi and A. Toet, "Visual comfort of binocular and 3-d displays," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 25, no. 2, pp. 99–108, 2004. 2

[18] M. Tuceryan and N. Navab, "Single point active alignment method (spaam) for optical see-through hmd calibration for ar," in *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR 2000)*, Oct 2000, pp. 149–158. 2

[19] Y. Genc, M. Tuceryan, and N. Navab, "Practical solutions for calibration of optical see-through devices," in *Proceedings. International Symposium on Mixed and Augmented Reality*, Oct 2002, pp. 169–175. 2

[20] J. Grubert, J. Tümler, R. Mecke, and M. Schenk, "Comparative user study of two see-through calibration methods." in *IEEE Virtual Reality (VR)*, 01 2010, pp. 269–270. 2

[21] Y. Itoh and G. Klinker, "Interaction-free calibration for optical see-through head-mounted displays based on 3d eye localization," in *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, March 2014, pp. 75–82. 2

[22] A. Plopski, Y. Itoh, C. Nitschke, K. Kiyokawa, G. Klinker, and H. Takemura, "Corneal-imaging calibration for optical see-through head-mounted displays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 4, pp. 481–490, April 2015. 2

[23] S. J. Gilson, A. W. Fitzgibbon, and A. Glennerster, "Spatial calibration of an optical see-through head-mounted display," *Journal of Neuroscience Methods*, vol. 173, no. 1, pp. 140 – 146, 2008. 2

[24] Hong Hua, Chunyu Gao, and N. Ahuja, "Calibration of a head-mounted projective display for augmented reality systems," in *Proceedings. International Symposium on Mixed and Augmented Reality*, Oct 2002, pp. 176–185. 2

[25] Leap Motion, Inc., "Bending reality: North star's calibration system," 2019. [Online]. Available: http://blog.leapmotion.com/bending-reality-north-stars-calibration-system/ 2

[26] "Calibu." [Online]. Available: https://github.com/arpg/Calibu 4

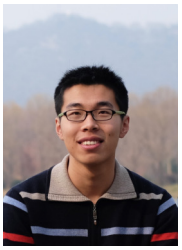
[27] S. Lovegrove, A. Patron-Perez, and G. Sibley, "Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras," in *British Machine Vision Conference, BMVC 2013, Bristol, UK, September 9-13, 2013*, 2013. [Online]. Available: https://doi.org/10.5244/C.27.93 4

[28] F. Zernike, "Beugungstheorie des schneidenverfahrens und seiner verbesserten form, der phasenkontrastmethode," *Physica*, vol. 1, no. 8, p. 689–70, 1934. 5, 8, 10

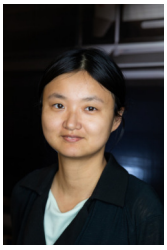
[29] S. Agarwal and K. Mierle, *Ceres Solver: Tutorial & Reference*, Google Inc. 6

[30] T. Flynn, "Colossal bust of Ramesses II," creative commons, Attribution-NonCommercial 4.0 International (CC BY-NC 4.0). [Online]. Available: https://sketchfab.com/3d-models/colossal-bust-of-ramesses-ii-v20-71355c314b2740e38f4329f658a509178, 9

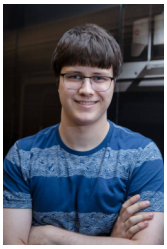
[31] M. Born and E. Wolf, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light (7th ed.)*, 1999. 10



Qi Guo is a PhD student at Harvard University advised by Todd Zickler. He is interested in combining optics and computer vision algorithms to create computational sensors. He received his bachelors degree in automation from Tsinghua University. He has interned at Facebook Reality Labs, Nvidia and Baidu, and was awarded ECCV Best Student Paper in 2016 and ICCP Best Demo in 2018.



Huixuan Tang is a research scientist at Facebook Reality Labs. She received PhD degree in Computer Science from University of Toronto in 2017. Her research interests are in augmented reality, computational photography and computer vision.



Aaron Schmitz is a mechanical engineer at Facebook AR Hardware. He received his master's degree in electrical engineering from the University of Washington in 2017 and his bachelor's degree in mechanical engineering from the University of Minnesota in 2013.

Wenqi Zhang is a software engineer at Facebook Reality Labs. He received his Master of Science degree from University of Southern California in 2013 and Bachelor of Engineering degree from Tsinghua University in 2011.

Yang Lou is a research scientist at Facebook Reality Labs. He received PhD degree in Biomedical Engineering from Washington University in St. Louis in 2018. His research interest includes computational photography, computer vision, and medical imaging.

Alexander Fix is a research scientist at Facebook Reality Labs. He received his PhD in Computer Science from Cornell University in 2016. His research interests include optimization and 3D computer vision.

Steven Lovegrove is a research scientist at Facebook Reality Labs. He received his PhD degree in Computer Vision from Imperial College London in 2012. His research interests are in augmented reality, computer vision, and robotics.

Hauke Malte Strasdat is a research scientist at Facebook Reality Labs. He received his PhD degree in Computing from Imperial College London in 2012. His research interests are in augmented reality, computer vision, and robotics.