BundleFit: Display and See-Through Models for Augmented Reality Head-Mounted Displays

Yufeng Zhu
mike323zyf@gmail.com

Abstract

1. Introduction

In augmented reality (AR), computer-generated visual content is superimposed on the user's view of the real world through devices such as smartphones, tablets, and specialized glasses. An AR head-mounted display (HMD), typically consisting of a see-through display placed in front of the user's eyes, is a device worn on the head that allows the user to see both the real world and virtual content overlaid on it. AR HMDs can be used for a wide range of applications, including entertainment, education, and industrial training. They have the potential to revolutionize the way we interact with the world and each other by providing a more immersive and interactive experience. Some examples of AR HMDs include the Microsoft HoloLens and the Magic Leap One [2, 27].

To achieve a seamless and immersive AR experience, the virtual elements must be correctly registered with the physical objects. This is where world-locked rendering (WLR) plays a crucial role, as it is a technique used in AR systems to generate visual content that appears as if it is part of the real world, rather than being displayed on top of it. WLR systems rely on a combination of sensors and computer vision techniques to ensure that the virtual content is correctly aligned even if the users move or change their perspective. Despite the advances made in WLR, there are still a number of challenges that need to be addressed, which includes improving the accuracy and stability of various sub-modules, such as inside-out tracking (IOT), eye tracking, display and see-through modeling, etc [25]. In this work, we focus on addressing the topics related to the display and see-through modeling of AR HMDs with complex optical designs.

The optical design of a see-through display can present various challenges for the WLR system, including highly nonlinear geometric distortion, non-uniformity, and other issues. Additionally, changes in the user's pupil position can exacerbate these challenges due to the phenomenon known as "pupil swim" [9]. To address the problem of viewpoint-dependent geometric distortion, researchers have investigated various approaches to display and see-through modeling [12]. However, they have struggled with either accuracy issues or a lack of flexibility. In this paper, we propose new display and see-through models that are designed to overcome these difficulties. Briefly, we examine the fiber bundle structure [20] within the context of display and see-through problems and present black box fitting models that can be generalized to different optical designs of see-through displays and exhibit improved structure preservation properties, leading to enhanced accuracy. We will begin by discussing our proposed models, and then outline the process of calibration and WLR systems integration. Finally, we assess the performance of the models through both simulated experiments and real-world evaluations using AR HMDs. As we will demonstrate, our models are not dependent on the specific optical design and are able to achieve high levels of accuracy in modeling performance.

2. Related Work

Our research falls under the category of AR HMD display and see-through calibrations, a highly active area of study within the domains of computer vision and graphics [12]. Historically, this task has been challenging due to the complexity of the optics involved [1, 6, 7, 26, 32]. Traditional methods, such as the Single Point Active Alignment Method (SPAAM) and its variations [8, 13, 39] were used to estimate the projection from the real world to the display by requiring users to manually align highlighted pixels with pre-determined 3D points, which can be time-consuming and susceptible to user errors. To minimize human involvement in the calibration process, Owen et al. [35] proposed a two-phase method that greatly reduces manual alignment effort. More recently, fully automated approaches [10, 21] have been developed to eliminate the need for manual calibration altogether, making the process more efficient and less prone to errors caused by user variability. However, these methods have limitations in terms of accuracy [22], as they often rely on simplified models for the display and see-through optics. Additionally, the geometric distortion caused by the optics can vary significantly depending on the

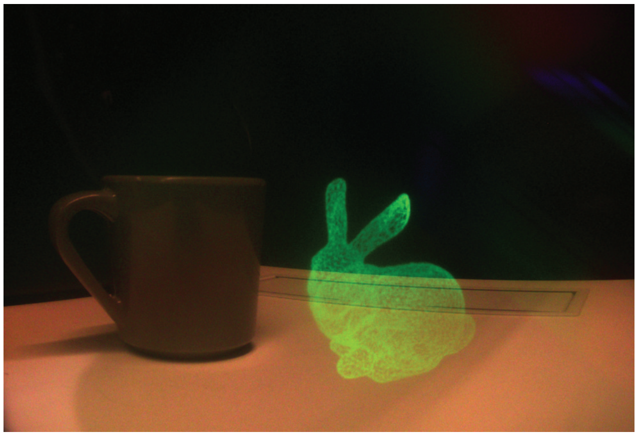


Figure 1. Real example figure for World Locked Rendering.

user's pupil position, known as pupil swim [3, 5, 9], which also needs to be considered in the calibration process.

To improve the accuracy of WLR systems, different methods have been examined. Recognizing that each display pixel is perceived as a point light source from the viewer's perspective, researchers have employed triangulation techniques to determine its 3D position for each pixel in order to model the display optics [28, 29]. However, this assumption is not sufficient to fully capture complex optical designs, as the rays corresponding to each pixel may not converge at a single point. As an alternative, using a light field representation is more appropriate for precise modeling [23, 24]. Another direction is to trace the rays through display and see-through optics for each pixel and estimate unknown parameters by minimizing reprojection errors [9, 15]. To reduce the computational cost of ray tracing, researchers have applied nonlinear dimension reduction techniques to pre-traced light rays from simulations, enabling real-time image distortion correction [14]. These methods, however, are sensitive to the accuracy of manufacture and assembly process, as they depend on a certain Computer-Aided Design (CAD) model. Additionally, they are inflexible as the ray tracing software implementation is specific to certain optical systems. With the growing trend of neural network development, this technique has also been adopted in the modeling of display optics [18, 37].

3. Display And See-Through Models

Display and see-through models are essential components for world-locked AR, which is to position virtual objects with reference to real objects as shown in Fig. 1. The display model is designed to correct for the complex geometric distortions that are often introduced by the display optics, such as a curved combiner. The see-through model is needed to align the display and IOT subsystem within a common coordinate frame. To align virtual and real objects accurately, the AR system must have a precise mapping from the physical world to the display panel. This is achieved by performing a calibration process for the IOT, eye tracking, display, and see-through models prior to using world-locked applications on AR devices. In this paper, we will focus on the discussion of the display and see-through models, which are the main technical contributions of our work. We will also briefly mention the IOT and eye tracking modules to show how our proposed models can be integrated into a world-locked AR system. We direct readers interested in these topics to further literature [16, 19, 38] for a more comprehensive review.

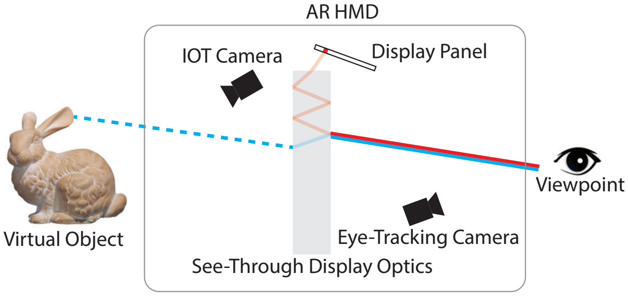


Figure 2. Explaination figure for World Locked Rendering.

3.1. Background

We begin with a brief introduction to the WLR system and emphasize the roles of the display and see-through models. For the sake of simplicity, we make the following assumptions: (1) camera's depth of field is infinite; (2) we are limiting our consideration to geometric optics; (3) undesired artifacts caused by the optics, such as ghosting, nonuniformity, and others, are not taken into account. By using an HMD with rigidly mounted IOT cameras and IMUs, we can estimate the HMD's pose through front-end visual inertial odometry (VIO) and the geometry of the surrounding physical objects through back-end simultaneously localization and mapping (SLAM), provided that the IOT system is calibrated. The IOT problem has been extensively researched in previous works [31, 33]. An HMD equipped with an IOT system will have both its pose and the geometry of the real world represented in its own coordinate frame (i.e. device frame). Without loss of generality, we will consider the monocular case only. As an illustration, we have selected a sculpture of a bunny, shown in Fig. 2. Our goal is to demonstrate how to render a virtual bunny on the display in such a manner that it appears aligned with the real sculpture from the perspective of a viewpoint provided by the eye tracking system. It is assumed that the eye tracking cameras are calibrated and connected to the IOT system, allowing both the geometry of the sculpture and the viewpoint to be represented in the device frame.

To determine the values of each pixel on the display panel, researchers have suggested using a white box model [9, 15], which involves explicitly tracing rays throughout the optical system. As depicted in Fig. 2, the path of the corresponding ray (red) is traced from each pixel through the specified viewpoint, which reveals how the pixel is perceived from that perspective. Next, a ray (blue) with the same viewing angle is traced through the see-through optics, emanating from the viewpoint and projecting into the real world. Eventually, the intersection of this ray path (blue) with the virtual objects to be displayed (in this case, the virtual geometry of the sculpture computed by the SLAM) determines the value of the corresponding pixel.

The white box model has the advantage of being physically interpretable, but suffers from the following drawbacks: (1) it relies on the accuracy of the hardware manufacturing and assembly process in order to achieve accurate performance; (2) it lacks flexibility and requires changes to the software implementation when the hardware design is altered. Alternatively, a black box model can avoid these issues by modeling the relationship between the display pixels and the rays (dashed blue) projected into the real world, which is ultimately what matters, directly as a function (see Fig. 2) and omit all the intermediate details of ray tracing. However, this approach is not generally effective as it disregards the role of the viewer's perspective. In general, the pixel-ray mapping is viewpoint dependent because the correspondence it describes varies significantly according to the viewer's positions due to the effect of pupil swim [9].

To address this issue, we utilize two black box models: the display model and the see-through model, eliminating the need for intricate ray tracing process within display and see-through optics (light red and light blue). The display model (in Sec. 3.3) explains the relationship between the display pixels and the rays (solid red) that are perceived from the viewer's perspective, while the see-through model (in Sec. 3.4) describes the correspondence between rays (solid blue) starting from the viewpoint and rays (dashed blue) that project into the real world. In order to provide a better understanding of the motivation and context for our new approach, we begin by discussing the rationale behind it in Sec. 3.2.

3.2. Fiber Bundle Fitting

Model fitting is a well-researched technique that has been widely applied to solve a diverse range of problems and challenges in various fields. It involves identifying the most suitable model to describe the relationships between a set of variables, estimating the model's parameters using available data, and evaluating the model's ability to represent the relationships and make predictions [34]. In this work, we are applying this technique to address display and see-through problems, attempting to find models that can accurately describe the correspondence relationships involving pixels and rays.

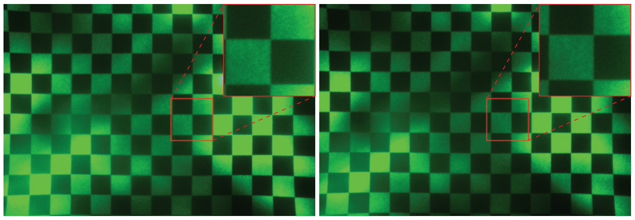


Figure 3. Pupil swim for display model.

One of the primary contributions of this work is the finding that, when the data space exhibits a fiber bundle structure and the projection operation is feasible, it is more effective to fit the subset of data on the base space rather than the entire data space in order to better preserve the fiber bundle structure. For example, let's assume that we have a set of fitting data, $(p_i, q_i), p_i \in P, q_i \in Q$, and want to fit the correspondence model, $f : P \mapsto Q$. If the data space P has a fiber bundle structure $\pi : P \mapsto B$, where P is the total space and B is the base space, we can fit intermediate model $\tilde{f} : B \mapsto Q$ instead. Then the desired models can be constructed as $f = \tilde{f} \circ \pi$ by combining with the projection operation π. Inspired by this idea, we will exploit the fiber bundle structure property in the development of the display and see-through models.

3.3. Display Model

The display model illustrates the connection between the display pixels and the rays perceived from the viewer's perspective. This correspondence may vary as the viewpoint shifts due to the effect of pupil swim as shown in Fig. 3. Thus the forward and backward display problems can be defined as

$$\begin{gathered} f_{display} : (\mathbf{p}_{pixel}, \mathbf{p}_{view}) \mapsto \mathbf{v}_{view}, \\ \mathbf{p}_{pixel} \in \mathbb{R}^2, \mathbf{p}_{view} \in \mathbb{R}^3, \mathbf{v}_{view} \in \mathbb{S}^2, \end{gathered} \tag{1}$$

and

$$f^{-1}_{display} : (\mathbf{p}_{view}, \mathbf{v}_{view}) \mapsto \mathbf{p}_{pixel}, \tag{2}$$

respectively. In Eq. (1) and Eq. (2), $\mathbf{p}_{pixel}$ denotes a display pixel's position, while $\mathbf{p}_{view}$ is the viewpoint position and $\mathbf{v}_{view}$ represents the direction of the corresponding perceived ray.

Volumetric Viewpoint Sampling It is straightforward to model such correspondence by following the problem definitions. To collect data for fitting the model, a calibrated eyeball camera is used to represent the viewer's perspective. This process involves sampling pixels on the 2D display panel and taking volumetric samples of viewpoints by placing the eyeball camera at various 3D locations to capture the

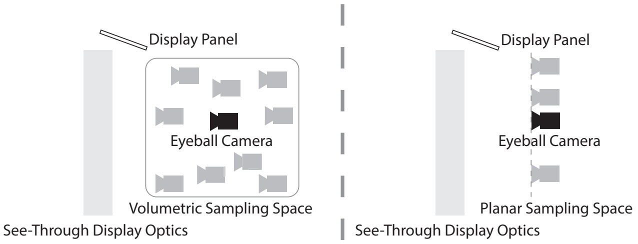

Display Panel
Eyeball Camera
Volumetric Sampling Space
See-Through Display Optics
Display Panel
Eyeball Camera
Planar Sampling Space
See-Through Display Optics

Figure 4. Volume sampling approach vs plane sampling.

corresponding rays from different perspectives (see Fig. 4). Parameters of the chosen model can then be estimated using the collected data. However, this volumetric viewpoint sampling approach is resource-intensive and prone to inaccurate modeling. Specifically, if the fitting step is not carefully constrained, the fitted model may incorrectly predict that two viewpoints with the same viewing angle for a pixel will have different perspectives of that pixel (see Fig. 5).

Planar Viewpoint Sampling Upon closer examination of the correspondence problems in Eq. (1) and Eq. (2), we observe that the data space $(\mathbf{p}_{view}, \mathbf{v}_{view})$ of each pixel, which consists of its corresponding rays emanating from the display optics, has a fiber bundle structure. As shown in Fig. 6, the base space for each pixel's fiber bundle is the surface of the display optics and the fibers are its corresponding rays. Motivated by the discussion in Sec. 3.2, we introduce two intermediate model problems in a similar way,

$$\begin{aligned}\tilde{f}_{display} : (\mathbf{p}_{pixel}, \tilde{\mathbf{p}}_{view}) &\mapsto \mathbf{v}_{view}, \\ \mathbf{p}_{pixel}, \tilde{\mathbf{p}}_{view} \in \mathbb{R}^2, &\mathbf{v}_{view} \in \mathbb{S}^2,\end{aligned} \tag{3}$$

and

$$\tilde{f}^{-1}_{display} : (\tilde{\mathbf{p}}_{view}, \mathbf{v}_{view}) \mapsto \mathbf{p}_{pixel}. \tag{4}$$

Our proposed method involves sampling the viewpoint position $\tilde{\mathbf{p}}_{view}$ on a two-dimensional plane, $\mathbb{P}_{display}$, as depicted in Fig. 4, rather than in three dimensions. We then use the projection operations to recover the desired forward and backward display models.

The projection operation $\pi^{\vee}_{display}$ for the backward model is simple and involves intersecting the 2D plane $\mathbb{P}_{display}$ with the ray emanating from the 3D viewpoint $\mathbf{p}_{view}$ and traveling in the opposite direction of $\mathbf{v}_{view}$ as in Eq. (5).

$$\begin{aligned}\pi^{\vee}_{display} : (\mathbf{p}_{view}, \mathbf{v}_{view}) &\mapsto \tilde{\mathbf{p}}_{view}, \\ \tilde{\mathbf{p}}_{view} = \mathbf{p}_{view} &- t^* \cdot \mathbf{v}_{view},\end{aligned} \tag{5}$$

where

$$t^* = \operatorname*{argmin}_{t \geq 0} dist(\mathbf{p}_{view} - t \cdot \mathbf{v}_{view}, \mathbb{P}_{display}), \tag{6}$$

and $dist(\cdot, \cdot)$ is the distance function. Therefore, the backward display model can be constructed as

$$f^{-1}_{display} = \tilde{f}^{-1}_{display}(\pi^{\vee}_{display}(\mathbf{p}_{view}, \mathbf{v}_{view}), \mathbf{v}_{view}). \tag{7}$$

While the projection operation for the backward model is relatively obvious, the forward model is more involved. In the forward case, the goal is to predict how a pixel $\mathbf{p}_{pixel}$ is perceived from a given viewpoint $\mathbf{p}_{view}$, but the intermediate forward model $\tilde{f}_{display}$ that we fit requires the viewpoint position to be located on the base plane $\mathbb{P}_{display}$. We formulate this projection operation, $\pi^{\wedge}_{display}$, as an optimization problem,

$$\begin{aligned}\pi^{\wedge}_{display} &: (\mathbf{p}_{pixel}, \mathbf{p}_{view}) \mapsto \tilde{\mathbf{p}}_{view}, \\ \tilde{\mathbf{p}}^*_{view}, t^* &= \operatorname*{argmin}_{\tilde{\mathbf{p}}_{view}, t \geq 0} dist(g(\tilde{\mathbf{p}}_{view}, \mathbf{p}_{pixel}, t), \mathbf{p}_{view}), \\ &\quad s.t. \quad \tilde{\mathbf{p}}_{view} \in \mathbb{P}_{display}, \\ g(\tilde{\mathbf{p}}_{view}, \mathbf{p}_{pixel}, t) &= \tilde{\mathbf{p}}_{view} + t \cdot \tilde{f}_{display}(\mathbf{p}_{pixel}, \tilde{\mathbf{p}}_{view}),\end{aligned} \tag{8}$$

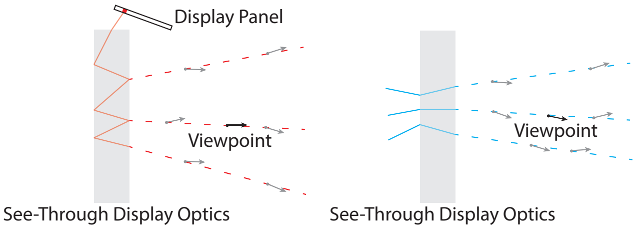
whose global optimal candidate $\tilde{\mathbf{p}}^*_{view}$ is the projection of $\mathbf{p}_{view}$. Or in other words, $\tilde{\mathbf{p}}^*_{view}$ perceives $\mathbf{p}_{pixel}$ in the same way as $\mathbf{p}_{view}$ perceives it. Then it is straightforward to construct the forward display model as follows:

$$f_{display} = \tilde{f}_{display}(\mathbf{p}_{pixel}, \pi^{\wedge}_{display}(\mathbf{p}_{pixel}, \mathbf{p}_{view})). \tag{9}$$

It is evident that our new approach is more resource-efficient compared to the volumetric viewpoint sampling approach. More importantly, it avoids the fitting artifacts, as illustrated in Fig. 5, by leveraging the fiber bundle structure present in the display problems.

3.4. See-Through Model

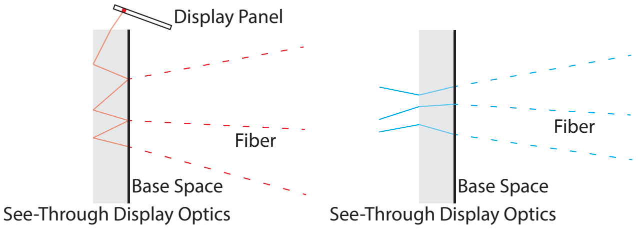
The see-through model describes the relationship between rays that originate from the viewpoint and rays that project into the real world. Similar to the display model,



(a) Volume sampling approach failure for display.

(b) Volume sampling approach failure for seethru.

Figure 5. Volume sampling approach failure.



(a) Display fiber bundle structure. (b) Seethru fiber bundle structure.

Figure 6. Fiber bundle structure.

it also depends on the viewpoint, as shown in Fig. 7. The forward and backward see-through problems can be defined as

$$\begin{aligned} f_{seethru} &: (\mathbf{p}_{view}, \mathbf{v}_{view}) \mapsto (\mathbf{p}^{+}_{real}, \mathbf{p}^{-}_{real}), \\ \mathbf{p}^{+}_{real}, \mathbf{p}^{-}_{real} &\in \mathbb{R}^2, \mathbf{p}_{view} \in \mathbb{R}^3, \mathbf{v}_{view} \in \mathbb{S}^2, \end{aligned} \tag{10}$$

and

$$\begin{aligned} f^{-1}_{seethru} :&(\mathbf{p}_{view}, \mathbf{p}_{real}) \mapsto \mathbf{v}_{view}, \\ &\mathbf{p}_{real} \in \mathbb{R}^3, \end{aligned} \tag{11}$$

respectively. In Eq. (10) and Eq. (11), $\mathbf{p}_{view}$ denotes the viewpoint position and $\mathbf{v}_{view}$ represents the ray direction starting from the viewpoint as in Sec. 3.3. Additionally, $\mathbf{p}_{real}$ is a 3D location in the real world, while $\mathbf{p}^{+}_{real}$ and $\mathbf{p}^{-}_{real}$ are the two-plane parameterization of the real world light field [11,30], which are two points located on two real world planes, $\mathbb{P}^{+}_{real}$ and $\mathbb{P}^{-}_{real}$. The forward model unravels how the see-through optics alters the path of a ray when it is projected from a viewpoint into the real world. And the backward model details how a real world point will be perceived from a viewer's perspective through the see-through optics. As discussed in Sec. 3.3, we should avoid volumetric viewpoint sampling approach and take advantage of the fiber bundle structure in the see-through problems. Accordingly, we also adopt a planar viewpoint sampling strategy and propose an intermediate model problem,

$$\begin{aligned} \tilde{f}_{seethru} &: (\tilde{\mathbf{p}}_{view}, \mathbf{v}_{view}) \mapsto (\mathbf{p}^{+}_{real}, \mathbf{p}^{-}_{real}), \\ \tilde{\mathbf{p}}_{view}, \mathbf{p}^{+}_{real}, \mathbf{p}^{-}_{real} &\in \mathbb{R}^2, \mathbf{v}_{view} \in \mathbb{S}^2. \end{aligned} \tag{12}$$

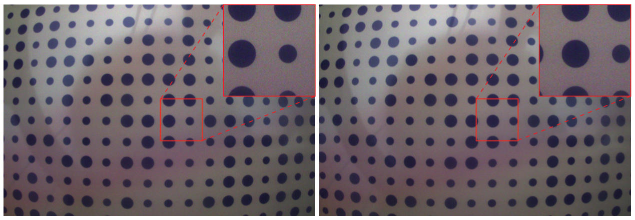


Figure 7. Pupil swim for see-through model.

The viewpoint $\tilde{\mathbf{p}}_{view}$ is sampled on a 2D plane, $\mathbb{P}_{seethru}$ and we will define the projection operations to restore the forward and backward see-through models.

This time, the forward see-through model shares a similar projection operation with the backward display model (see Eq. (5) and Eq. (6)), which is defined as

$$\begin{aligned} \pi^{\wedge}_{seethru} : (\mathbf{p}_{view}, \mathbf{v}_{view}) &\mapsto \tilde{\mathbf{p}}_{view}, \\ \tilde{\mathbf{p}}_{view} = \mathbf{p}_{view} &- t^{*} \cdot \mathbf{v}_{view}, \end{aligned} \tag{13}$$

where

$$t^{*} = \underset{t \geq 0}{\operatorname{argmin}}\, dist(\mathbf{p}_{view} - t \cdot \mathbf{v}_{view}, \mathbb{P}_{seethru}). \tag{14}$$

And the forward model can be easily constructed as

$$f_{seethru} = \tilde{f}_{seethru}(\pi^{\wedge}_{seethru}(\mathbf{p}_{view}, \mathbf{v}_{view}), \mathbf{v}_{view}). \tag{15}$$

On the other hand, the projection operation of the backward model requires a little bit more effort to build. In this case, we want to understand how a given viewpoint $\mathbf{p}_{view}$ perceives a 3D real world point $\mathbf{p}_{real}$. Since we've already fitted an intermediate model $\tilde{f}_{seethru}$ in Eq. (12), which explains how rays from the real world become rays that reach the viewer after passing through the see-through optics, the only remaining task is to determine which ray emanating from $\mathbf{p}_{real}$ will ultimately reach $\mathbf{p}_{view}$. We formulate this problem as an optimization process,

$$\begin{aligned} \pi^{\vee}_{seethru} &: (\mathbf{p}_{view}, \mathbf{p}_{real}) \mapsto \tilde{\mathbf{p}}_{view}, \\ \tilde{\mathbf{p}}^{*}_{view}, t^{*} &= \underset{\tilde{\mathbf{p}}_{view}, t}{\operatorname{argmin}}\, dist(h(\tilde{\mathbf{p}}_{view}, \mathbf{p}_{view}, \mathbf{p}_{real}, t), \mathbf{p}_{real}), \\ s.t. \;\; &\tilde{\mathbf{p}}_{view} \in \mathbb{P}_{seethru}, \\ h(\tilde{\mathbf{p}}_{view}, \mathbf{p}_{view}, \mathbf{p}_{real}, t) &= \mathbf{p}^{+}_{real} + t \cdot (\mathbf{p}^{+}_{real} - \mathbf{p}^{-}_{real}), \\ \mathbf{p}^{+}_{real}, \mathbf{p}^{-}_{real} &= \tilde{f}_{seethru}(\tilde{\mathbf{p}}_{view}, \frac{\mathbf{p}_{view} - \tilde{\mathbf{p}}_{view}}{\|\mathbf{p}_{view} - \tilde{\mathbf{p}}_{view}\|}), \end{aligned} \tag{16}$$

whose global optimal candidate $\tilde{\mathbf{p}}^{*}_{view}$ is the projection of $\mathbf{p}_{view}$. Thus $\mathbf{p}_{view}$ will perceive $\mathbf{p}_{real}$ in the same way that $\tilde{\mathbf{p}}^{*}_{view}$ perceives it through the see-through optics. And the backward see-through model can be defined as

$$\begin{aligned} f^{-1}_{seethru} &= \frac{\mathbf{p}_{view} - \tilde{\mathbf{p}}_{view}}{\|\mathbf{p}_{view} - \tilde{\mathbf{p}}_{view}\|}, \\ \tilde{\mathbf{p}}_{view} &= \pi^{\vee}_{seethru}(\mathbf{p}_{view}, \mathbf{p}_{real}), \end{aligned} \tag{17}$$

which is simply the direction vector of the ray that passes through both $\mathbf{p}_{view}$ and its projection on the base plane $\mathbb{P}_{seethru}$.

4. Results

In this section, we will discuss how to incorporate our proposed models into the WLR system, including model calibration, integration with rendering frameworks, and evaluation of the models through simulation and testing on a real AR device.

4.1. Model Calibration

In Sec. 3, we introduced our proposed display and see-through models under the assumption that the intermediate models, defined in Eq. (3), Eq. (4) and Eq. (12), have been selected and fitted. In this section, we will provide more details on the intermediate model calibrations. Theoretically, any reasonable choice should be acceptable in our case and we choose the polynomial model for its simplicity and differentiability. Moreover, the calibration data (i.e. model training data) will be collected and used to minimize the model fitting error in order to determine the model parameters, which are polynomial coefficients in our setting. We will move on to the introduction of calibration data collection and skip the explanation of model fitting process, as polynomial fitting is a standard technique [34].

Display Model In Eq. (3) and Eq. (4), we aim to model how a pixel $\mathbf{p}_{pixel}$ is perceived from a viewpoint $\tilde{\mathbf{p}}_{view}$ located on the base plane $\mathbb{P}_{display}$. As previously mentioned in Sec. 3.3, we use a calibrated eyeball camera to capture the direction of a pixel's corresponding ray, $\mathbf{v}_{view}$. The camera is rigidly mounted to a motion stage that is equipped with a micrometer controller as depicted in Fig. 8. We build an HMD calibration station and position the camera around the eye box, which is located approximately 7-12 mm away from the display optics.

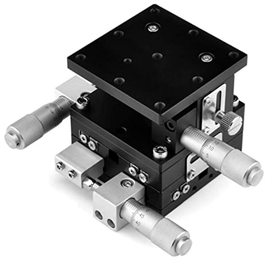


Figure 8. Motion stage with micrometer controller.

By adjusting the micrometer controller, we can move the camera to different locations within a single plane, which we refer to as the display base plane $\mathbb{P}_{display}$. At each camera position $\tilde{\mathbf{p}}_{view}$, we collect correspondence data between display pixels $\mathbf{p}_{pixel}$ and perceived rays $\mathbf{v}_{view}$ by displaying some calibration patterns, such as gray code [4], fringe patterns [40], etc. We can then fit the polynomial model for Eq. (3) and Eq. (4) using the collected data. To facilitate integration with the IOT subsystem, we convert the eyeball camera positions and perceived rays into the device frame before model fitting. We will wait until the next paragraph to explain how to perform the coordinate frame conversion.

See-Through Model Eq. (12) describes the correspondence between the light fields on the real-world side and the viewer side. We adopt a two-plane parameterization [11,30] for the real world light field and a base-direction parameterization for the other one. To collect the light field correspondence data, we follow these steps:

1. Place a calibration target in front of the HMD station and use the eyeball camera to take one capture without the HMD mounted (see Fig. 9), which gives us the relative pose between the eyeball camera and the target plane $\mathbb{P}^{+}_{real}$.

2. Put the HMD on the calibration station and use the IOT camera to take one capture of the calibration target (see Fig. 9), which gives us the pose of the eyeball camera and the target plane $\mathbb{P}^{+}_{real}$, both represented in the device frame.

3. Use the micrometer controller to move the eyeball camera to a set of capture locations $\tilde{\mathbf{p}}_{view}$ within the see-through base plane $\mathbb{P}_{seethru}$ (see Fig. 9). By using a pre-calibrated eyeball camera and matching the captured images with the calibration target pattern, we are able to obtain a collection of correspondences between the camera viewing directions $\mathbf{v}_{view}$ and locations $\mathbf{p}^{+}_{real}$ on the target plane for each camera position $\tilde{\mathbf{p}}_{view}$, all represented in the device frame.

4. Move the calibration target to a different location and repeat step 2 and 3 (see Fig. 9). From step 2, we have another target plane $\mathbb{P}^{-}_{real}$ represented in the device frame. From step 3, we collect the correspondences between $\mathbf{v}_{view}$ and $\mathbf{p}^{-}_{real}$ for the same set of $\tilde{\mathbf{p}}_{view}$ (all in device frame) by repeating the camera capture locations.

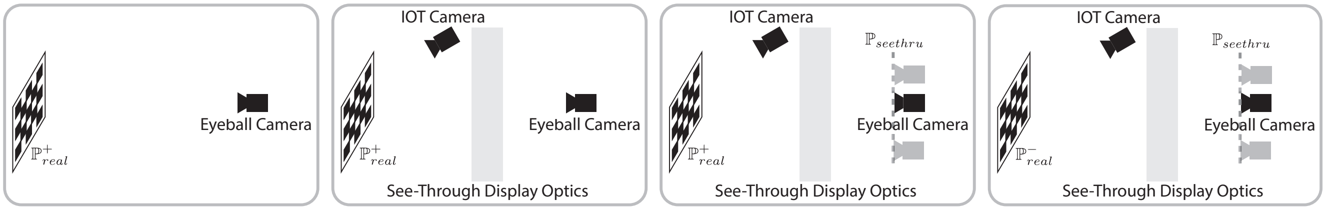
With the assumption that the micrometer controller is precise and the HMD calibration station remains stable when the HMD is mounted, we can collect the necessary fitting data for Eq. (12) using the steps described above.

4.2. Rendering Framework Integration

Our calibrated display and see-through models can be integrated with the WLR system described in Sec. 3.1. Now we outline the process of incorporating these models into two popular rendering frameworks, ray tracing and rasterization, used by such a system.

Ray Tracing In this framework, each pixel's value will be determined by tracing the path of its corresponding ray through a 3D scene [36]. Given a viewpoint $\mathbf{p}_{view}$ from the eye tracking system, our forward display and see-through models (Eq. (1) and Eq. (10)) can provide the necessary correspondence between display pixels $\mathbf{p}_{pixel}$ and rays projected into the real world $(\mathbf{p}^{+}_{real}, \mathbf{p}^{-}_{real})$. Therefore, it is straightforward to use our models in combination with a ray tracing rendering engine to generate the display image.

Rasterization This rendering technique works in a way that is different from ray tracing by projecting a 3D scene



(a) Two plane calibration.

(b) Two plane calibration.

(c) Two plane calibration.

(d) Two plane calibration.

Figure 9. Two plane calibration.

onto the image plane [17]. We use a two-phase approach to integrate it with our models. The first step involves rendering an intermediate image by positioning the rasterization camera at a given viewpoint. It is then warped onto the display image space to generate the final result. To compute the mapping between display image space and intermediate image space, we use our forward display and see-through models to find the corresponding ray, projected into the real world, for each display pixel and intersect it with the 3D scene. The intersection is then mapped onto the intermediate image space, using the rasterization camera's projection, to obtain the corresponding 2D position for each display pixel. It is notable that using rasterization rendering engines will introduce errors to the WLR system because their camera projections do not take into account changes in the ray path caused by the see-through optics.

4.3. Simulation Experiments

We perform Monte Carlo simulations to assess the individual performance of the display and see-through models. The CAD descriptions of a see-through display optical system serve as input for generating ground truth data through ray tracing. To obtain information for fitting and testing the models, we use a pinhole camera positioned at multiple sample locations. We sample the camera's image space to gather rays corresponding to camera pixels, and trace these rays through the optical system to find the matching display pixels for the display problem or the rays projected into the real world for the see-through problem. To mimic the system noise that occurs during actual data capture, we introduce zero-mean Gaussian perturbations to the camera pixel samples [15]. The model accuracy is measured by quantifying the pixel offset for the backward display problem and the viewing angle discrepancy for the others [24].

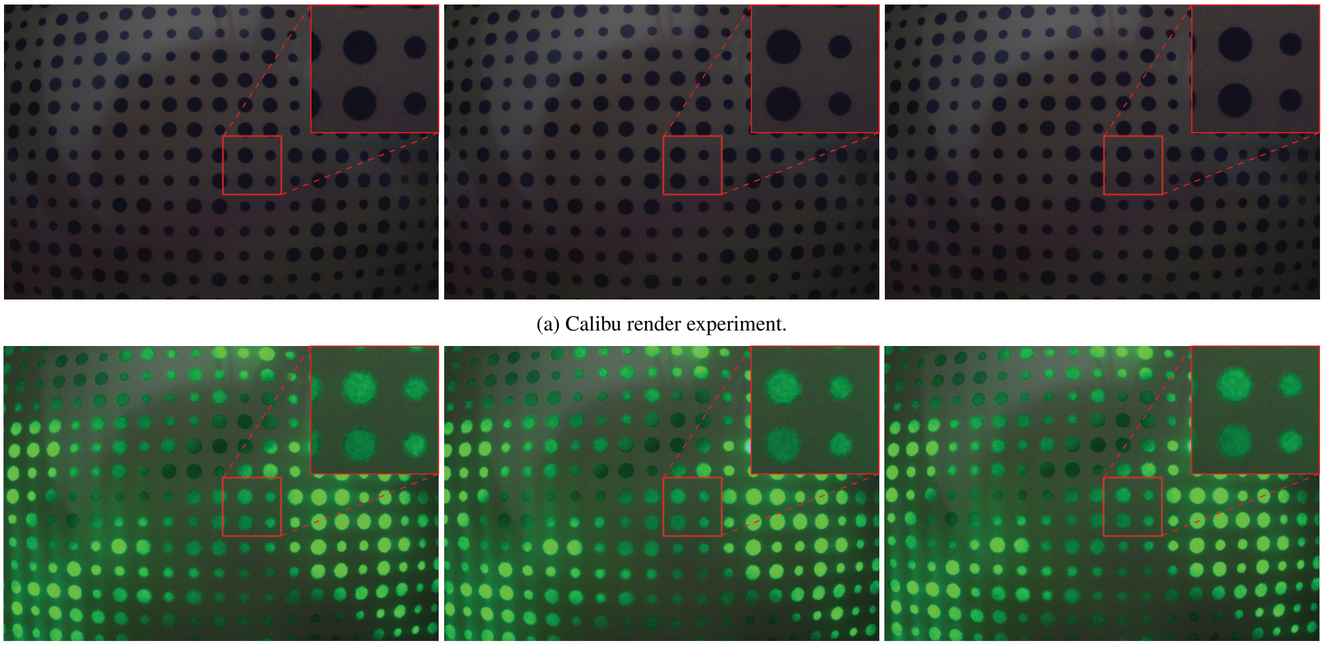
4.4. On-Device Experiments

We employ an HMD with see-through display optics and IOT system to evaluate the performance of display and see-through models. As described in Sec. 4.1, we use a pair of pre-calibrated eyeball cameras, mounted on a motion stage with micrometer controller, to capture display and see-through calibration patterns. During the calibration phase for the display model, we will secure the HMD onto the calibration station and use the eyeball cameras to record the displayed gray code patterns [4] at distinct locations within the 3D eye box region for volumetric sampling approach and different positions on a 2D plane in front of the display optics for our planar sampling method. In order to calibrate the see-through models, we use the same sampling positions for the eyeball cameras, which were utilized during the display calibration procedure. We use Calibu patterns [15], which are composed of a matrix of large and small dots arranged on a flat surface, to calibrate the see-through models. These dots, differentiated by the size of their adjacent dots, facilitate the efficient establishment of correspondences.

To evaluate the precision of the calibrated models, we carry out an end-to-end verification procedure by integrating them with a ray tracing rendering engine and our IOT system. In a manner similar to the see-through calibration setup, we will position a Calibu pattern in front of the HMD calibration station and adjust the position of the eyeball cameras while the HMD is mounted. The IOT system will first calculate the pose of the Calibu pattern and transforms it into the device frame. Then the display and see-through models will work together to determine the image to be displayed as discussed in Sec. 4.2, ensuring that the rendered Calibu pattern precisely aligns with the actual pattern from the perspective of each eyeball camera's position (see Fig. 10). In our experiment, we position the Calibu pattern in distinct locations in front of the calibration station and capture images using the eyeball cameras from different viewpoints within the eye box region. For each static configuration of the Calibu pattern and eyeball cameras, we will take two pictures using the eyeball cameras. As depicted in Fig. 10, one image will be captured with the display turned off, while the other will be captured with the display turned on. In this way, we can easily match the corresponding dots in each image pair and calculate the misalignment to evaluate the end-to-end model performance.

References

[1] Ronald Azuma and Gary Bishop. Improving static and dynamic registration in an optical see-through hmd. In *Pro-*



(a) Calibu render experiment.

(b) Calibu render experiment.

Figure 10. Calibu render experiment.

ceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, pages 197–204, 1994. 1

[2] Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4):355–385, 1997. 1

[3] Joshua M Cobb, David Kessler, and John A Agostinelli. Optical design of a monocentric autostereoscopic immersive display. In *International Optical Design Conference 2002*, volume 4832, pages 80–90. SPIE, 2002. 2

[4] Robert W Doran. The gray code. Technical report, Department of Computer Science, The University of Auckland, New Zealand, 2007. 6, 7

[5] Glenn A Fry. The center of rotation of the eye. *American Journal of Optometry*, 39:581–595, 1962. 2

[6] Anton Fuhrmann, Dieter Schmalstieg, and Werner Purgathofer. Fast calibration for augmented reality. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 166–167, 1999. 1

[7] Yakup Genc, Frank Sauer, Fabian Wenzel, Mihran Tuceryan, and Nassir Navab. Optical see-through hmd calibration: A stereo method validated with a video see-through system. In *Proceedings IEEE and ACM International Symposium on Augmented Reality*, pages 165–174. IEEE, 2000. 1

[8] Yakup Genc, Mihran Tuceryan, and Nassir Navab. Practical solutions for calibration of optical see-through devices. In *Proceedings. International Symposium on Mixed and Augmented Reality*, pages 169–175, 2002. 1

[9] Ying Geng, Jacques Gollier, Brian Wheelwright, Fenglin Peng, Yusufu Sulai, Brant Lewis, Ning Chan, Wai Sze Tiffany Lam, Alexander Fix, Douglas Lanman, and Others. Viewing optics for immersive near-eye displays: Pupil swim/size and weight/stray light. In *Digital Optics for Immersive Displays*, volume 10676, pages 19–35. SPIE, 2018. 1, 2, 3

[10] Stuart J Gilson, Andrew W Fitzgibbon, and Andrew Glennerster. Spatial calibration of an optical see-through head-mounted display. *Journal of Neuroscience Methods*, 173(1):140–146, 2008. 1

[11] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of The 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 43–54, 1996. 5, 6

[12] Jens Grubert, Yuta Itoh, Kenneth Moser, and J. Edward Swan. A survey of calibration methods for optical see-through head-mounted displays. *IEEE Transactions on Visualization and Computer Graphics*, 24(9):2649–2662, 2018. 1

[13] Jens Grubert, Johannes Tuemle, Ruediger Mecke, and Michael Schenk. Comparative user study of two see-through calibration methods. *IEEE Virtual Reality (VR)*, 10(269-270):16, 2010. 1

[14] Phillip Guan, Olivier Mercier, Michael Shvartsman, and Douglas Lanman. Perceptual requirements for eye-tracked distortion correction in vr. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–8, 2022. 2

[15] Qi Guo, Huixuan Tang, Aaron Schmitz, Wenqi Zhang, Yang Lou, Alexander Fix, Steven Lovegrove, and Hauke Malte Strasdat. Raycast calibration for augmented reality hmds with off-axis reflective combiners. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2020. 2, 3, 7

[16] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 2

[17] Donald Hearn, M Pauline Baker, and M Pauline Baker. *Computer Graphics with OpenGL*, volume 3. Pearson Prentice Hall Upper Saddle River, NJ:, 2004. 7

[18] Yuichi Hiroi, Kiyosato Someya, and Yuta Itoh. Neural distortion fields for spatial calibration of wide field-of-view near-eye displays. *Optics Express*, 30(22):40628–40644, 2022. 2

[19] Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost Van de Weijer. *Eye Tracking: A Comprehensive Guide to Methods and Measures*. OUP Oxford, 2011. 2

[20] Dale Husemoller. *Fibre bundles*, volume 5. Springer, 1966. 1

[21] Yuta Itoh and Gudrun 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)*, pages 75–82. IEEE, 2014. 1

[22] Yuta Itoh and Gudrun Klinker. Performance and sensitivity analysis of indica: Interaction-free display calibration for optical see-through head-mounted displays. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 171–176. IEEE, 2014. 1

[23] Yuta Itoh and Gudrun Klinker. Light-field correction for spatial calibration of optical see-through head-mounted displays. *IEEE Transactions on Visualization and Computer Graphics*, 21(4):471–480, 2015. 2

[24] Yuta Itoh and Gudrun Klinker. Simultaneous direct and augmented view distortion calibration of optical see-through head-mounted displays. In *2015 IEEE International Symposium on Mixed and Augmented Reality*, pages 43–48. IEEE, 2015. 2, 7

[25] Yuta Itoh, Tobias Langlotz, Jonathan Sutton, and Alexander Plopski. Towards indistinguishable augmented reality: A survey on optical see-through head-mounted displays. *ACM Computing Surveys (CSUR)*, 54(6):1–36, 2021. 1

[26] Falko Kellner, Benjamin Bolte, Gerd Bruder, Ulrich Rautenberg, Frank Steinicke, Markus Lappe, and Reinhard Koch. Geometric calibration of head-mounted displays and its effects on distance estimation. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):589–596, 2012. 1

[27] Kangsoo Kim, Mark Billinghurst, Gerd Bruder, Henry Been-Lirn Duh, and Gregory F Welch. Revisiting trends in augmented reality research: A review of the 2nd decade of ismar (2008–2017). *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2947–2962, 2018. 1

[28] Martin Klemm, Fabian Seebacher, and Harald Hoppe. Non-parametric camera-based calibration of optical see-through glasses for ar applications. In *2016 International Conference on Cyberworlds (CW)*, pages 33–40. IEEE, 2016. 2

[29] Martin Klemm, Fabian Seebacher, and Harald Hoppe. High accuracy pixel-wise spatial calibration of optical see-through glasses. *Computers & Graphics*, 64:51–61, 2017. 2

[30] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of The 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 31–42, 1996. 5, 6

[31] Mingyang Li and Anastasios I Mourikis. High-precision, consistent ekf-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013. 2

[32] Erin McGarrity and Mihran Tuceryan. A method for calibrating see-through head-mounted displays for ar. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality*, pages 75–84. IEEE, 1999. 1

[33] Anastasios I Mourikis, Stergios I Roumeliotis, et al. A multi-state constraint kalman filter for vision-aided inertial navigation. In *ICRA*, volume 2, page 6, 2007. 2

[34] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012. 3, 6

[35] Charles B Owen, Ji Zhou, Arthur Tang, and Fan Xiao. Display-relative calibration for optical see-through head-mounted displays. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 70–78. IEEE, 2004. 1

[36] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2016. 6

[37] Kiyosato Someya, Yuichi Hiroi, Makoto Yamada, and Yuta Itoh. Ostnet: Calibration method for optical see-through head-mounted displays via non-parametric distortion map generation. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 259–260. IEEE, 2019. 2

[38] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002. 2

[39] Mihran Tuceryan and Nassir Navab. Single-point active alignment method (spaam) for optical see-through hmd calibration for augmented reality. In *Proceedings IEEE and ACM International Symposium on Augmented Reality*, volume 11, pages 149–158, 2000. 1

[40] Chao Zuo, Shijie Feng, Lei Huang, Tianyang Tao, Wei Yin, and Qian Chen. Phase shifting algorithms for fringe projection profilometry: A review. *Optics and Lasers in Engineering*, 109:23–59, 2018. 6