

Gaze-Vergence-Controlled See-Through Vision in Augmented Reality

Zhimin Wang*, Yuxin Zhao*, and Feng Lu[†], Senior Member, IEEE

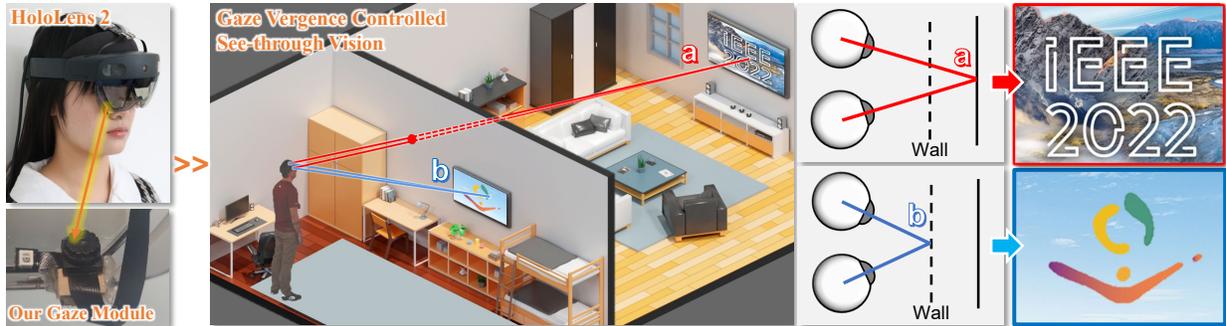


Fig. 1. We propose a gaze-vergence-controlled see-through vision in AR. We build a gaze tracking module with two infrared cameras and assemble it into the Microsoft HoloLens 2. With our gaze depth estimation algorithm, the user's gaze depth can be computed from gaze vergence and used to control see-through vision.

Abstract— Augmented Reality (AR) see-through vision is an interesting research topic since it enables users to see through a wall and see the occluded objects. Most existing research focuses on the visual effects of see-through vision, while the interaction method is less studied. However, we argue that using common interaction modalities, *e.g.*, midair click and speech, may not be the optimal way to control see-through vision. This is because when we want to see through something, it is physically related to our gaze depth/vergence and thus should be naturally controlled by the eyes. Following this idea, this paper proposes a novel gaze-vergence-controlled (GVC) see-through vision technique in AR. Since gaze depth is needed, we build a gaze tracking module with two infrared cameras and the corresponding algorithm and assemble it into the Microsoft HoloLens 2 to achieve gaze depth estimation. We then propose two different GVC modes for see-through vision to fit different scenarios. Extensive experimental results demonstrate that our gaze depth estimation is efficient and accurate. By comparing with conventional interaction modalities, our GVC techniques are also shown to be superior in terms of efficiency and more preferred by users. Finally, we present four example applications of gaze-vergence-controlled see-through vision.

Index Terms—Augmented Reality, See-through Vision, Gaze Vergence Control, Gaze Depth Estimation

1 INTRODUCTION

Virtual Reality and Augmented Reality (VR & AR) have attracted much attention from both academia and industry in the past five years. In particular, with the rise of the meta-verse in recent years, AR and VR are widely considered the keys to the next generation of the internet [3] [72]. The AR/VR industries continue to climb in market value [1]. These technologies are also utilized in a large number of applications from different fields, *e.g.*, games, education and health care [44].

While VR produces immersive virtual worlds generated by computer graphics, AR technology aims at enhancing the user experience by seamlessly integrating the virtual objects with the physical world [60]. The big tech giants, *e.g.*, Microsoft and Apple, are also shifting their focus to AR and trying to apply AR technology in different areas such as intelligent manufacturing and online retail [2].

Since AR is able to link the real and virtual worlds, one interesting application is to expand the user's vision, such as allowing the user to see the occluded objects behind a wall, namely see-through vision [52]. The see-through vision has been explored in recent years [46, 55, 56]. Researchers have made efforts to improve the visual effect of see-through vision in AR [9, 13]. For instance, Avery *et al.* designed the *Edge Overlay* technique to provide depth cues for see-through vision [10]. Erat *et al.* presented the user's view with photorealistic rendering from a three-dimensional reconstruction of hidden areas [22].

The above works make see-through vision more natural and realistic. However, the way to interact with see-through vision is less studied. In fact, see-through vision can significantly benefit from interaction control, so as to enrich the user experience when using AR Head Mounted Display (HMD) devices [14, 49]. By intention, the user can turn on/off see-through vision or show it at a different distance. However, we argue that using the common interaction modalities, *e.g.*, midair click and speech, may not be the optimal way to control see-through vision. This is because when the user wants to see through a wall, he needs to think about the corresponding click gesture or speech command and then execute it. It is not intuitive and requires extra effort to switch the thinking, which will distract the user's attention.

Intuitively, the human eye gaze can be a more natural input to control see-through vision. When we intend to see through something, we are actually fixating at a new distance, which is physically related to the gaze depth/vergence. For instance, the gaze depth increases when we fixate on the occluded objects behind the wall, while it decreases when we look at the target at a nearer distance.

*These two authors contributed equally to the paper.

[†]Corresponding author: Feng Lu.

ZhiminWang and Yuxin Zhao are with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: zm.wang@buaa.edu.cn; zyuxin@buaa.edu.cn).

Feng Lu is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, also with the Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: lufeng@buaa.edu.cn).

Inspired by this observation, a natural idea is to control see-through vision by gaze depth/vergence. However, it is not easy to use gaze vergence to control see-through vision in AR HMDs. The problems can be summarized in three aspects. 1) The mainstream AR devices do not support gaze depth estimation. For example, the Microsoft HoloLens 2 only offers single gaze ray but does not provide the gaze vergence or access to eye images [5]. 2) Recent studies using gaze vergence for interaction are mostly in desktop or VR scenarios [7, 35]. These methods rely on specific SDKs that cannot be easily adapted to the AR HMD. 3) There are few works in the literature that discuss how to flexibly control see-through vision by gaze depth.

To address these issues, our solution contains the following steps: 1) We build a gaze tracking module with two infrared cameras and assemble it into the Microsoft HoloLens 2, as shown in Fig. 1. 2) We design two gaze depth estimation methods, which can be easily adapted to different eye trackers. 3) With our gaze depth estimation algorithm, we propose two control modes of gaze vergence and apply them to see-through vision. We also investigate the efficiency of different modalities by quantitative performance measurements as well as subjective feedback. Finally, we demonstrate the gaze-vergence-controlled techniques with four example applications¹.

Overall, our contributions are as follows:

1. **Novelty:** We propose a Gaze-Vergence-Controlled (GVC) see-through vision technique in AR, offering new experiences.
2. **System Implementation:** We customize two eye cameras and design gaze depth estimation methods for HoloLens 2. We also show that these methods are accurate and effective for see-through vision control.
3. **Control Modes:** We propose two control modes of gaze vergence for see-through vision, which are called Stimulus-Guided (SG) see-through mode and Self-Control (SC) see-through mode.
4. **Evaluation:** We demonstrate the efficiency and usability of our method through comparison and analysis. Four example applications of gaze vergence control are presented.

2 RELATED WORK

In this section, we review see-through vision and gaze interaction in AR and discuss the estimation methods of gaze depth.

2.1 See-Through Vision

Occlusion visualization has been extensively explored in recent years. Elmqvist *et al.* reviewed fifty techniques of occlusion management and classified them into five patterns [21]. We mainly concentrate on two patterns that are related to our work.

See-through vision. The see-through vision can make the occluding surface partially transparent to turn objects visible [29, 46]. Researchers have made efforts to improve the visual effect of see-through vision in AR [9, 13, 25, 30]. For instance, Avery *et al.* provided see-through visualization with depth cues when users viewed hidden objects behind walls [10]. Erat *et al.* synthesized three-dimensional models of occluded areas for presenting the photorealistic see-through vision. They also controlled a camera drone to explore the real scene via hand gestures and gaze direction [22]. Bane *et al.* presented four interactive tools that allow users to explore see-through vision with different perspectives [11].

Multi-perspective visualization. The multi-perspective vision is characterized by transforming an alternative view into the main window [64, 71]. Prior studies captured occluded regions from the secondary perspective and integrated them seamlessly into the user's view [61, 70]. Lilija *et al.* compared four different views for occluded object manipulation [40]. They found see-through vision had the best performance.

To summarize, previous literature mainly focused on the overlay effect of hidden areas and occluding layers. However, the interaction

method is less studied. In fact, the see-through vision can significantly benefit from the interaction control. According to the intention, the user can turn on/off see-through vision or show it at a different distance. However, we argue that using the common interaction modalities, *e.g.*, midair click and speech, may not be the optimal way to control the see-through vision. This is because when we want to see through something, it is physically related to our gaze depth/vergence and thus should be naturally controlled by the eyes. Inspired by this fact, we propose a novel gaze-vergence-controlled see-through vision in AR.

2.2 Gaze Interaction in AR

Interaction techniques aim to improve the user experience, which is vital for AR HMD devices. With the rise of gaze estimation accuracy [17, 41, 67], different gaze-based techniques have been explored, such as gaze dwelling and vergence eye movement.

Gaze dwelling. Most existing works exploit gaze dwelling as the input technique [37, 62, 66]. For instance, Wang *et al.* used one second as the dwell time of selection for gaze-based interaction [65]. However, such gaze inputs often suffer from the Midas Touch problem [45], where users unintentionally trigger selections with natural eye movements.

Vergence eye movement. Recent research tried to achieve Midas-touch-free interaction with vergence eye movement [7, 34, 35, 59, 63, 68]. For instance, Hirzle *et al.* controlled the presentation of hidden virtual content triggered by gaze vergence [27]. Compared with gaze dwelling, confirming selections via gaze vergence can be clearly distinguished from random visual skimming of the interface. Therefore, vergence eye movement has the inherent advantage of addressing the Midas Touch problem. However, there are few works in the literature that discuss how to flexibly control see-through vision by gaze depth. To this end, we propose two control modes of gaze vergence for see-through vision, which are called Stimulus-Guided (SG) see-through mode and Self-Control (SC) see-through mode.

2.3 Gaze Depth Estimation

Many studies have investigated how to compute the gaze depth, which can be broadly classified into two categories: 1) gaze ray-casting methods and 2) vergence-based methods.

Gaze ray-casting methods. In these methods, the single gaze ray intersects the first object in the scene, and the intersection is taken as the 3D Point of Regard (PoR) [42, 69]. The distance between the PoR and the center of both eyes is defined as the gaze depth. However, these methods do not deal with the occlusion ambiguity where multiple objects interact with the gaze ray, as they do not estimate the gaze depth directly. Therefore, the gaze ray-casting methods are not suitable for the gaze-vergence-controlled technique.

Vergence-based methods. The gaze vergence will change quickly when both eyes simultaneously move in opposite directions to fixate on objects at different depths. The vergence-based methods generally include indirect and direct methods. These indirect techniques first compute vergence-related features from near-eye images, *e.g.*, Inter-pupillary Distance (IPD), and then use them to regress the gaze depth [43, 48, 50]. For instance, Alt *et al.* detected the pupil diameter and IPD to estimate gaze depth and hence enabled gaze-based interaction with 3D virtual objects [8]. These direct methods obtain the gaze depth by computing the intersection of the gaze rays from both eyes [34, 47]. However, it is yet unclear as to which method could achieve better performance in AR HMD. In this work, we implemented and compared two widely used methods in HoloLens 2, *i.e.*, 3D line-of-sight intersection [34] and IPD-based regression [35, 36].

3 SYSTEM DESIGN

3.1 Overview

We propose a novel gaze-vergence-controlled see-through vision in AR. An overview of our work is shown in Fig. 2. To compute the gaze depth/vergence, we first design the gaze depth computation module. This module utilizes two methods to compute gaze depth, which are the 3D Line-of-sight Intersection (3D LoSI) and the Inter-pupillary Distance (IPD) based regression. Based on the predicted depth, we further propose two gaze-vergence-controlled modes of see-through

¹Project page: https://zhimin-wang.github.io/GVC_See_Through_Vision.html

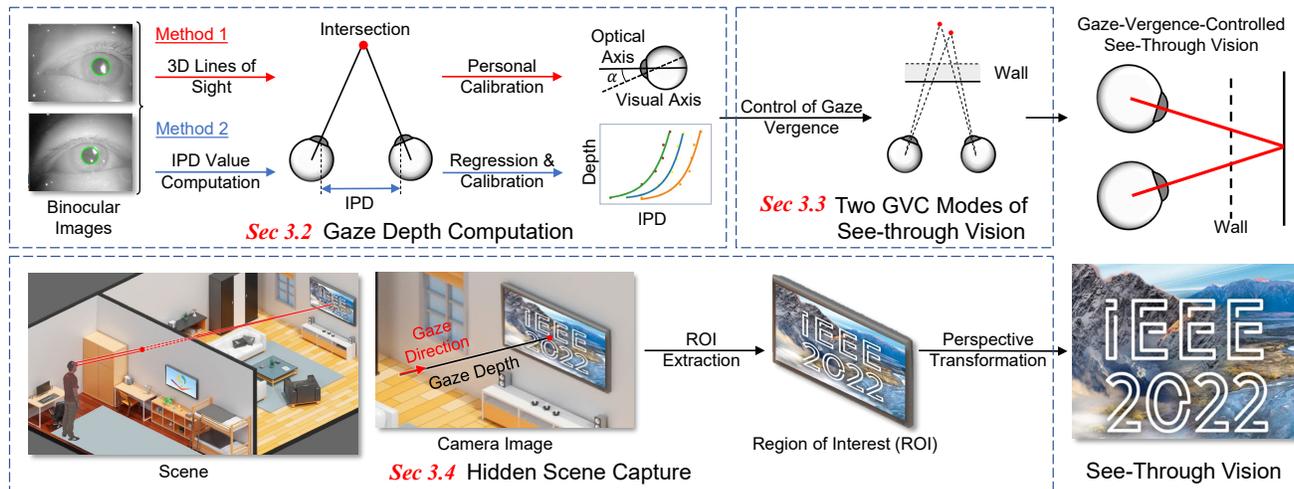


Fig. 2. Overview of our system. We propose the gaze-vergence-controlled see-through vision technique in AR. We first get gaze depth from the proposed gaze depth estimation algorithm. The red and blue arrows indicate our 3D Line-of-sight Intersection and IPD-based Regression methods, respectively. With our algorithm, we then design two gaze-vergence-controlled modes of see-through vision. Finally, we capture the hidden scene using a camera behind the wall. The camera’s view is seamlessly transformed into the user’s view.

vision. One is the Stimulus-Guided (SG) see-through mode and the other is the Self-Control (SC) see-through mode. Besides the interaction techniques, we also introduce how to present a natural visual effect of see-through vision, which reveals the hidden scene.

The rest of this section is organized as follows. 1) We introduce the gaze depth computation in Section 3.2 including 3D line-of-sight intersection and IPD-based regression. 2) The two gaze-vergence-controlled modes of see-through vision are introduced in Section 3.3. 3) We describe the presentation of see-through vision from the hidden scene in Section 3.4. 4) We finally provide the implementation details of this system at the end.

3.2 Gaze Depth Computation

The gaze depth is defined as the distance between the user’s PoR and the center of both eyes. We can compute the PoR using the intersections of the lines of sight from the left and right eyes. We build a gaze tracking module with two Near-Infrared (NIR) cameras and assemble it into the Microsoft HoloLens 2. Here we utilize two gaze depth computation methods. 1) The first way is to directly compute the 3D intersections of the lines of sight, as described in Section 3.2.1. 2) The other way is an implicit model, which takes the IPD as input and regresses the gaze depth, as presented in Section 3.2.2. The gaze vergence control combines the two methods, which will be described later in Section 4.

3.2.1 Method 1: 3D Line-of-sight Intersection

The 3D Line-of-sight Intersection (3D LosI) method uses the intersections of gaze rays from the left and right eyes. Because HoloLens 2 only provides a single line of sight, we need to modify it to support binocular gaze estimation. The mainstream strategy is to integrate HoloLens with the Pupil Labs’ eye tracker and use its software [20, 47]. However, this method does not calibrate the combined hardware beforehand. Instead, it merges the transformation between the scene camera and the eye camera with the kappa angle as a matrix to optimize. The kappa angle is the angle offset between the optical and visual axes [54]. This way causes an increase in systematic error [47].

3D Lines of Sight. To improve computation accuracy, our method is modified from Pupil Labs’ method in two ways: 1) employ the pupil detection method PuReST [53], which has robust performance to reflections or partial occlusion; 2) calibrate the hardware in advance and model the kappa angle. The goal of hardware calibration is to register the scene camera and eye cameras to a common coordinate system. A more detailed description of our calibration procedure follows in Section 3.5.1. The kappa angle is calculated by modeling the angle offset α between the visual and optical axes. The explicit definition

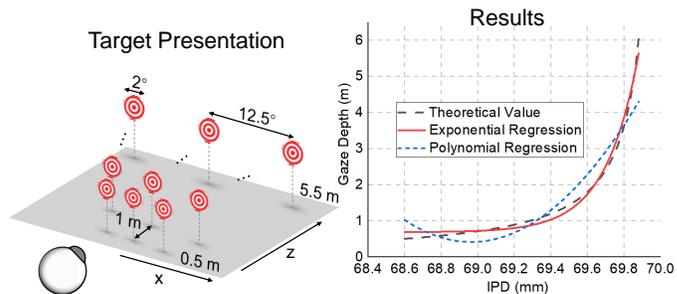


Fig. 3. Depth calibration and fitting. Subjects view the calibration scene consisting of gaze targets that are distributed in depth (left). The comparisons among the simulation (dashed black line), the exponential regression (solid red line), and the polynomial regression (dashed blue line) of the theoretical relationship between gaze depth and IPD (right).

of the kappa angle helps to compensate for the estimation error. We finally obtain two lines of sight from the left and right eyes.

Personal Calibration. We design a calibration scene to compute person-specific kappa angle $\hat{\alpha}$, as shown in the left part of Fig. 3. The gaze targets are displayed at depths between 0.5 m and 5.5 m and the distance interval in z axis direction is 1 m. The duration of each point is 2 seconds and we only record data during the last second. They are also scaled to subtend 2° of visual angle at all distances. The movement directions of peripheral targets at x - z plane keep 12.5° with the z axis. The y coordinates are set to the height of the user’s head. We collect some amounts of pupil data and gaze targets. Finally, we apply a least squares algorithm to optimize the kappa angle as Chen *et al.* did [16].

3D Gaze Intersection. After the above two procedures, we obtain accurate binocular gaze rays. Then we can calculate the intersections of two gaze rays as the 3D gaze points, using the function denoted as equation (7.14) in [57]. The gaze depth is the distance from the center of both eyes to the 3D intersection point.

3.2.2 Method 2: IPD-based Regression

The 3D LosI highly relies on the accuracy of binocular 3D gaze estimation. So it is also important to design a method that is insensitive to the line of sight. In this section, we introduce a technique that takes the IPD as input to regress gaze depth. Specifically, we implement two IPD-based methods: one utilizes the physical-based IPD in Millimeters (MIPD) to fit gaze depth, and the other uses the image-based IPD in

Pixels (PIPD) to regress the depth.

Compared with previous IPD-based studies, our methods differ in some aspects. First, prior research was mainly explored in the desktop environment or virtual reality settings [8, 35], which cannot be easily adapted to the AR HMD. Our module can be smoothly assembled with HoloLens 2. Second, we employ the robust pupil detection and accurate eye model fitting method, which have been demonstrated with superiority to previous methods [19, 53]. Another difference is the regression method. Previous research used the support vector regression or the neural network to learn the mapping from IPD value to gaze depth [38, 69] while we theoretically verify that exponential regression is enough for the task.

IPD Value Computation. As indicated above, there are two ways described as follows. We first perform the following procedure to obtain the MIPD: 1) Building the physical models of both eyes. We use the latest proposed 3D eye model fitting method [19], which can mitigate the effects of corneal refraction and apply the two-sphere eye model. 2) Both eye models are registered to a common coordinate system according to the calibration parameters of hardware, as described later in Section 3.5. We assume p_l and p_r are the 3D pupil centers of the left and right eyes. The MIPD is estimated as $\theta_1 = \|p_l - p_r\|_1$. We then compute the PIPD from each pair of eye images. Let x_l and x_r be the horizontal coordinates of the left and right pupil centers in images. The resolution of each image is 320×240 pixels. Therefore, the PIPD can be delivered as $\theta_2 = 320 - x_l + x_r$.

Regression for Depth. IPD-based regression needs to build a mapping from IPD value to gaze depth. To find an optimal mapping function, we simulate the relationship between gaze depth and IPD value theoretically, as the dashed black line shows in the right part of Fig. 3. We set the distance between both eyeball centers as 70 mm and the radius of the eyeball as 10.39 mm, which are provided by Pupil Labs [31]. According to the upward trend, we try to fit the simulation using polynomial and exponential regression. The results demonstrate that the exponential fitting approximates the theoretical value. It is similar to the finding by Kwon *et al.* [36], which used a logarithmic function to do that. Our regression function can be written as

$$\hat{d} = k_1 \cdot \exp(k_2 \cdot (\theta - \bar{\theta})) + k_3, \quad \bar{\theta} = \frac{1}{n} \sum_{i=1}^n \theta_i, \quad (1)$$

$$\hat{K} = \arg \min_K \left\{ \sum_{i=1}^n (d_i - \hat{d}_i)^2 \right\}, \quad K = \{k_1, k_2, k_3\}, \quad (2)$$

where \hat{d} is the estimated depth value while d is the truth value. θ is the IPD value, and its units can be millimeters or pixels. The $\bar{\theta}$ subtracts the average value $\bar{\theta}$ for accelerating the parameter fitting. n is the number of gaze targets collected in the calibration procedure. \hat{K} is the optimal parameter set. We also combine the regression with the Random Sample Consensus (RANSAC) [23] to discard outliers.

Personal Calibration. We utilize the same calibration scene as in Section 3.2.1 to compute the parameters \hat{K} . We split gaze targets into three sets according to horizontal FOV distributions and fit three exponential functions, respectively. An example of exponential fitting is shown in the top center part of Fig. 2. In the prediction period, we divide horizontal FOV into three sections, which are consistent with three functions. The gaze depth of each test datum is computed by the exponential function from the corresponding section.

3.3 Two Control Modes of See-through Vision

Based on the two carefully designed gaze depth computation methods above, we can successfully obtain the gaze depth. But there is another equally important matter that is how to control see-through vision by gaze depth. In this section, we design two different gaze-vergence-controlled modes. One is the Stimulus-Guided (SG) see-through mode, and the other is the Self-Control (SC) see-through mode. The character of the first mode is simple and easy-to-use for users, while the second mode is more novel and attractive. We can thus choose a more suitable mode according to the specific application scenarios in AR. To the best

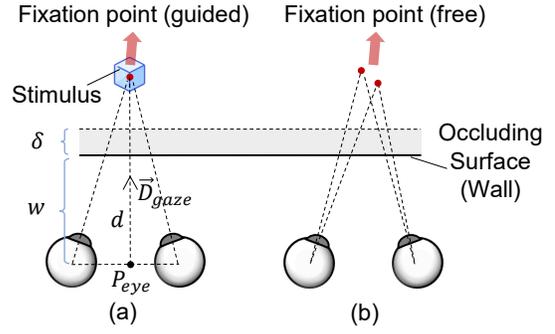


Fig. 4. Two control modes of gaze vergence when a user looks towards the occluded surface. (a) Stimulus-guided See-through mode. (b) Self-control See-through mode.

of our knowledge, there are few works in the literature that discuss how to flexibly control see-through vision by gaze depth.

Stimulus-Guided (SG) see-through mode. This mode allows users to trigger see-through vision by looking at a semi-transparent virtual stimulus behind the wall, as shown in Fig. 4a, which is similar to our viewing habit. We attempt some variations of the representation with the purpose of the stimulus being as noninvasive as possible for the user, *e.g.*, the size and transparency. We finally choose a purple cube with a length of 10 cm and a transparency of 50%. This cube is attached to the user's gaze ray, which is located 6 meters away from the eyes. The user first stands facing the wall. Then the participant employs the stimulus as visual guidance, and thus the fixation depth increases for exceeding the threshold of activation. Finally, the window of see-through vision is presented at a fixed distance, which helps to keep the PoRs fixated at a certain distance. More formally, the window position P_{window} of see-through vision in \mathbb{R}^3 is calculated as

$$\gamma = \begin{cases} w + j \cdot \delta, & \Phi(d) > w + \delta; \\ -\infty, & \text{otherwise,} \end{cases} \quad (3)$$

$$P_{window} = P_{eye} + \gamma \cdot \vec{D}_{gaze}, \quad (4)$$

where γ is the window depth of see-through vision. w is the distance from the user to the wall, while δ is the distance threshold, as shown in Fig. 4a. j is a scale factor greater than 1. d is the estimated depth value, and $\Phi(\cdot)$ is the filter function for data smoothing. P_{eye} is the center of both eyes, and \vec{D}_{gaze} is the normal vector of the gaze ray.

In the above-proposed model, some parameters need to be determined. A natural question arises: how to set reasonable parameters in practice. We set the range of w as (0.5, 3] m. This is because this distance range is the most commonly used range for daily indoor interaction. We call this distance range the middle distance in the context of our paper, referring to Bardins *et al.* [12]. To increase the robustness of control, the δ is determined by the mean and standard deviation of estimation error (see Section 4 for a more detailed description). We use the mean filter as the $\Phi(\cdot)$ and the time window is empirically set as 1 second [15]. To stabilize the PoRs at a certain distance, we set the scale factor j as 2. In fact, users do not need to know the depth of hidden objects. The user only needs to try to focus further, and as long as the gaze depth exceeds $w + \delta$, the hidden object is shown. Then the hidden object will guide the users' gaze to be stabilized at its depth.

Self-Control (SC) see-through mode. The SC see-through mode enables the user to freely control vergence eye movement without the need for a stimulus, as shown in Fig. 4b, which is novel and appealing. The user first stands w meters away from the wall. Then the user needs to perform a voluntary divergence eye movement to trigger see-through vision. This action is completed by contracting the extraocular muscles of the eyes [18]. The range of w , the distance threshold δ , the time window of $\Phi(\cdot)$, the scale factor j , and the computation of window position P_{window} are the same as in the first mode.

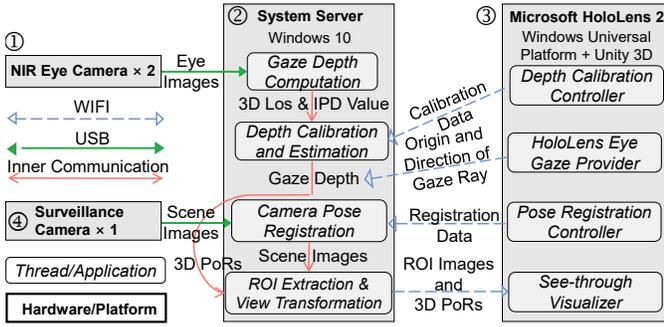


Fig. 5. The system architecture of our experimental setup. The system includes four main hardware components: 1) the customized NIR eye cameras, 2) the system server, 3) the HoloLens, 4) the surveillance camera. The software components run on the system server and HoloLens.

3.4 Hidden Scene Capture

The flexible control of see-through vision by gaze depth is elaborated on the last section. We further introduce how to acquire the content of see-through vision from the hidden scene. Here, we expect see-through vision to be natural and realistic. For example, the user’s view is consistent with physical laws, *i.e.*, the presented content consistent with that the user directly sees the scene without the occluding wall. Besides, the window of see-through vision naturally follows the gaze direction. To address these requirements, our solution consists of three steps: 1) To capture hidden scene, we embed a surveillance camera behind the occluding wall. The camera is first registered to the HoloLens coordinates. 2) We further compute the Region of Interest (ROI) of users in the HoloLens space and map the ROI into the camera space. 3) We finally perform a perspective transformation to transform the image of ROI into the user’s view in HoloLens. We illustrate these steps as follows.

Camera Pose Registration. The goal of this step is to register the camera coordinates to the HoloLens coordinates. We first manually align a virtual cuboid with a chessboard in AR and then register the camera to the HoloLens space H by detecting the chessboard. The width and length of this cuboid are equal to the size of the chessboard. We collect 2D pixel coordinates of the chessboard in C and 3D coordinates of the cuboid corners in H . Finally, we use the Efficient Perspective-n-Point (EPnP) algorithm [39] to compute the transformation T from H to C .

ROI Extraction of Hidden Scene. We naturally control the content of our see-through vision with eye movement. In short, we compute the ROI in the camera space C and clip the image of the ROI. Specifically, we first define the 3D PoR as the center of the user’s view (ROI) in H , which is a rectangle plane and perpendicular to the gaze ray. Then we compute the ROI in the camera space C by using the transformation T . Finally, we clip the image of ROI from the 2D image space of C .

Perspective Transformation. The user’s pose is different from the camera pose. In practice, it causes the viewpoint difference between them. Therefore, to make the user’s view consistent with physical laws, we map the image of ROI into the HoloLens space H . We apply the perspective transformation method [24] to transform them. For the final visual effect, please refer to Section 6.

3.5 System Implementation

A detailed overview of our system architecture, including hardware and software implementation and data flows between them, is shown in Fig. 5. We describe the implementation details as follows.

Hardware Setup. Our main hardware components are as follows: 1) The customized NIR eye cameras for gaze depth estimation are shown in Fig. 6. The eye cameras capture near-infrared images at 30Hz with 320×240 resolution. 2) The server uses an Intel Core i5-8500 with a 3.00Ghz CPU. 3) We use the Microsoft HoloLens 2 as the AR HMD. 4) One Logitech C9320e camera is used at 30Hz with 800×600 resolution.

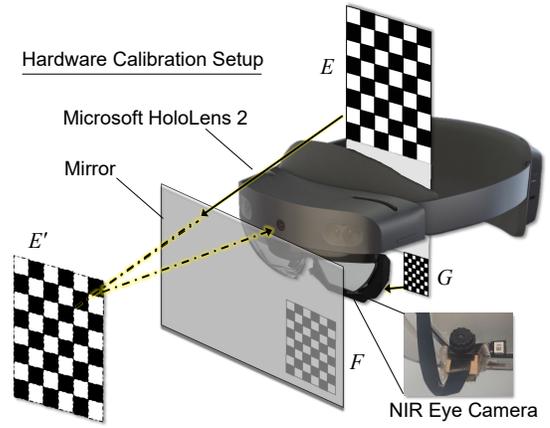


Fig. 6. A hardware calibration setup is used for calibrating the transformation between the scene camera and the customized eye cameras, including the two-chessboard pattern E - G , and another chessboard F attached to a mirror.

Hardware calibration. This module is used for calibrating the transformation between the scene camera and the customized NIR eye cameras. The hardware calibration is explored by Itoh *et al.* [28], who built a five-marker setup and registered these markers to a common coordinate system. This differs from our approach in that we employ a two-chessboard pattern and another chessboard attached to a mirror, as shown in Fig. 6, which is inspired by the mirror-based extrinsic calibration [58]. We minimize the number of markers, which can reduce the error caused by unifying different coordinate systems. The following describes the calibration procedure. 1) The scene camera detects the virtual image E' of the chessboard E , and we can compute the pose of E' in the scene camera coordinate system S . 2) The scene camera captures the chessboard F , and we can obtain the pose of the mirror in S . 3) Through the mirror symmetry, we can compute the pose of E in S . 4) The eye camera captures the chessboard G and the pose of G in the eye camera coordinate system N is obtained. 5) The E and G are coplane, and therefore they can be easily registered to a same coordinate system. Therefore, the S and N can share the common coordinate system.

Software Architecture. The software components and data streams between them are shown in Fig. 5. We use the MessagePack to un/pack the data [4] and the NetMQ for network communication [6]. We use Unity 3D for visualization on the HoloLens. The rendering rate on HoloLens is 60 fps while the frame rate on the system server is 30 fps.

4 QUANTITATIVE EVALUATION OF GAZE DEPTH ESTIMATION

Gaze depth estimation is one of the most important parts of our method, and therefore we first evaluated the depth accuracy of our proposed methods, namely 3D LosI, MIPD, and PIPD, described in Section 3.2, with the Pupil Labs 3D tracker [32]. We recruited 12 subjects from the campus (9 males and 3 females). The average age of participants is 23.9 (SD = 1.55). Three users had normal vision and nine users wore glasses. The experiments were conducted in an AR environment.

Design and Procedure. We designed a test scenario for evaluating these methods, which is similar to the calibration scenario in Section 3.2.2. We first introduced the experimental procedure to the participants. Then they performed gaze depth calibration as shown in the left part of Fig. 3. After that, participants began the test phase. In this phase, the gaze targets will appear 18 times in a random order within the range of 0.5 to 6 m. The size and duration of targets are the same as the calibration scenario. We collected pupil data and gaze targets to compare these methods simultaneously.

Results. We used the absolute difference between the estimated depth and the truth as an error evaluation metric. Quantitative comparison results are shown in Fig. 7 and Table 1, from which we make the following observations: 1) Overall, the 3D LosI achieves the best

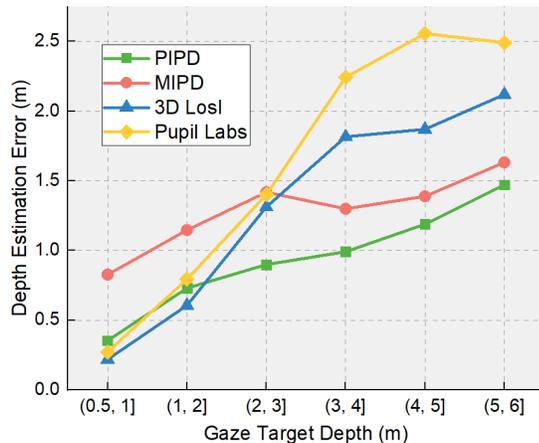


Fig. 7. Mean error comparison of our gaze depth estimation methods (PIPD, MIPD and 3D LosI) with the Pupil Labs 3D tracker. The standard deviation is not annotated in this figure for clear comparison.

Table 1. The error of depth estimation (m). The first row represents the distance range. The second to fifth rows include the mean of the error and its standard deviation.

Distance	(0.5, 1]	(1, 2]	(2, 3]	(3, 4]	(4, 5]	(5, 6]
PIPD	0.3±0.3	0.7±0.5	0.9±0.4	1.0±0.5	1.2±0.3	1.5±0.5
MIPD	0.8±1.2	1.1±0.7	1.4±0.8	1.3±0.5	1.4±0.7	1.6±0.6
3D LosI	0.2±0.1	0.6±0.4	1.3±0.9	1.8±0.7	1.9±0.4	2.1±0.4
Pupil Labs	0.3±0.2	0.8±0.5	1.4±0.7	2.2±0.6	2.6±0.5	2.5±0.4

performance in the range of (0.5, 2] m (error = 0.41 ± 0.34 m), while the PIPD outperforms the other methods at the (2, 6] m (1.14 ± 0.49 m). 2) Our 3D LosI (1.32 ± 0.88 m) surpasses the Pupil Labs 3D Tracker (1.63 ± 1.01 m) in all range of (0.5, 6] m. 3) We found the MIPD method has the highest error in the range of (0.5, 2] m (0.94 ± 1.03 m) due to two outliers.

Discussion. We discuss our results in three aspects. 1) The 3D LosI method slightly outperforms the PIPD at (0.5, 2] m. However, the error tends to increase abruptly with a slope of 0.6 at (2, 4] m. We argue that this is because after gaze depth exceeds 2 m, the accuracy of 3D LosI cannot meet the requirement that this method needs to correctly discriminate 1° vergence difference between 2 m and 4 m distance. 2) To overcome the above limitation, we build an optimal piecewise function for the gaze vergence control. Specifically, if the result of PIPD is in the range of (0.5, 2] m, we use the output of 3D LosI; otherwise, we still utilize the PIPD to estimate depth. We demonstrate that this piecewise function works efficiently for GVC techniques in the following section. 3) Our primary goal is to use the gaze vergence to perform daily indoor interaction within the middle distance, *i.e.*, (0.5, 3] m, as illustrated in Section 3.3. The gaze depth estimation error is 0.57 ± 0.44 m as predicted by the piecewise function. To increase the error-tolerant rate of the GVC techniques, we use the distance threshold δ as described in Section 3.3, which is set as the sum of mean error and standard deviation at each distance.

5 COMPARISONS OF INTERACTION MODALITIES FOR SEE-THROUGH VISION CONTROL

The primary goal of this section is to evaluate and compare the Gaze-Vergence-Controlled (GVC) techniques with two common modalities in AR see-through vision. There are few works in the literature that discuss how to control see-through vision by gaze depth in a flexible manner. We also want to know whether the GVC techniques have advantages over other modalities. To this end, we implement Stimulus-Guided Gaze (SGGaze) and Self-Control Gaze (SCGaze) and two conventional interactions, *i.e.*, midair click technique (Click) and

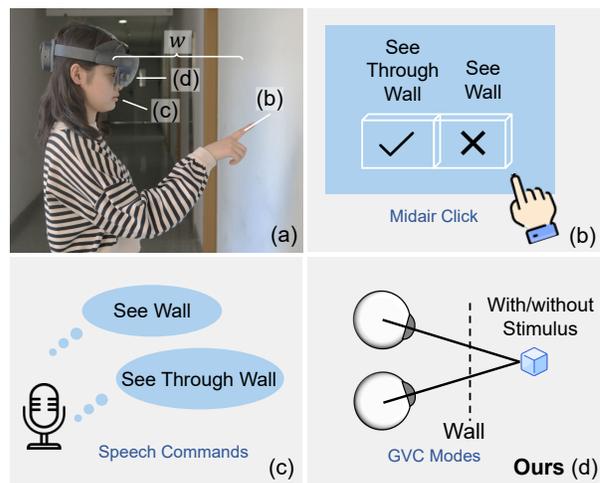


Fig. 8. The setting of see-through vision and the illustrations of four techniques. (a) The user employs four interaction modalities to control the see-through vision. w represents the distance between the user and the wall. (b) Midair click technique (Click). (c) Speech-based technique (Speech). (d) Stimulus-guided Gaze (SGGaze) and Self-control Gaze (SCGaze).

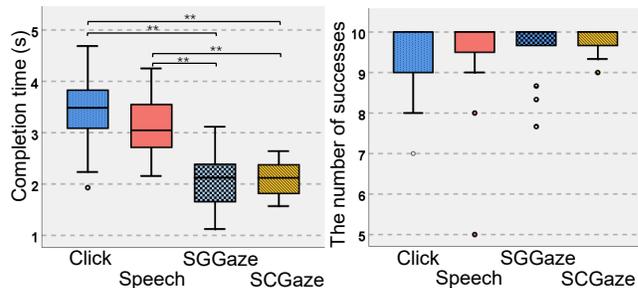


Fig. 9. Left: Boxplots of completion time of four modalities. Right: Boxplots of the number of successes of four modalities. The statistical significance is labeled with ** ($p < 0.05$). Error bars mean standard deviations. The little colored circles indicate the outliers. There is no statistically significant difference in the number of successes.

speech-based technique (Speech). We propose two hypotheses: H_1 : Controlling see-through vision has higher efficiency and usability with the GVC techniques than using Click and Speech within the middle distance. H_2 : Controlling see-through vision is more intuitive and attractive with the GVC techniques than using Click and Speech within the middle distance.

5.1 Participants and Task

We recruited 20 subjects from the campus (12 males and 8 females). The average age of participants is 24 (SD = 1.6). The pre-study questionnaire with 5-point Likert scales shows the participants have low prior familiarity with AR (Mean = 2.9), the eye tracker (Mean = 2.9), medium familiarity with speech-based inputs (Mean = 3.4), and high familiarity with button-based interaction (Mean = 4.2). All users can read and speak English fluently.

The task requires participants to control the visualization of occluded areas according to operating commands. In each trial, the distances between users and the wall are randomly chosen as 1, 2, or 3 m. Specifically, the user first views the objects attached to a wall. Then the user should do the following steps: 1) When the AR HMD displays the “See Through Wall”, the user tries to see through the wall using different modalities: clicking the “Check” button, saying “See Through Wall” or increasing the gaze depth. 2) Once the system shows the “See Wall”,

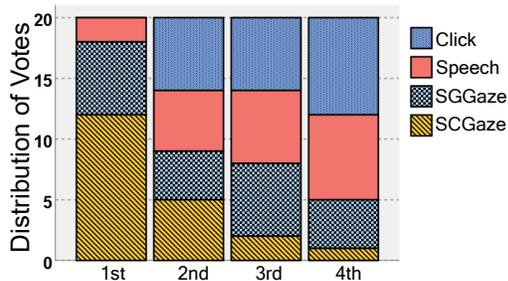


Fig. 10. The user preference ranking of four interaction modalities. The *SCGaze* is the most preferred by the users.

the participant does the opposite operations to close the see-through vision using four techniques. After each command is successfully performed, the user needs to keep the same state for 5 seconds and wait for the next command (called the waiting state later in Section 5.5.1). The user is required to repeat the above two steps five times.

5.2 Interaction Modalities

Midair Click Technique. The user employs index finger to touch the “Check” or “Uncheck” button for controlling see-through vision, see Fig. 8b. This technique is independent of the distance between users and the wall. Therefore, we evaluate it at a distance of 1 m.

Speech-based Technique. The participant uses the verbal command “See Through Wall” to see the occluded regions and says “See Wall” to turn off the see-through vision, see Fig. 8c. This modality is also unrelated to the distance and thus tested at a distance of 1 m.

The *SGGaze* and *SCGaze* techniques are implemented as described in Section 3.3, which is shown in Fig. 8d. Our goal is to use the GVC techniques in daily indoor interaction within the middle distance, *i.e.*, (0.5, 3] m. Therefore, we set the distance as 1, 2, and 3 m. We aim to explore whether different distances have an influence on performance.

5.3 Evaluation Metrics

Performance Measures. We employ three objective metrics to capture user performance: completion time, the number of successes, and the number of mistakes. Completion time is the time elapsed between when the operating command is displayed and when see-through vision is triggered or closed correctly. The number of successes is the number of times that the user triggers the corresponding operations successfully in 10 seconds. We count the number of mistakes as the number of times that the participant unintentionally triggers false commands in the waiting state.

Subjective Measures. Our subjective metrics describe the usability of four modalities. After finishing the task with one technique, users fill in the NASA’s Task Load Index [26] with 7-point Likert scales. Then they answer six free-response questions to report the naturalness and frustration of each modality. Upon the completion of all trials, they fill out a preference ranking questionnaire to rank all the modalities according to overall preference.

5.4 Experimental Procedure

The participants first filled in a pre-study questionnaire. Then they began a training phase where they were given visual and auditory instructions and practiced using different modalities. After training, they performed the experiments, including one task using four techniques and four questionnaires. The four interaction modalities were presented in random order. Each common modality was tested at a distance of 1 m. Each GVC technique ran a complete process for three distances. Users were required to rest for 30 seconds after each process to counteract the effects of fatigue. Finally, the participants filled out the preference ranking questionnaire. Prior to each section associated with gaze vergence, the users conducted gaze depth calibration as described in Section 3.2.1. For a fair comparison, the “See-through Wall” command was shown at the location of the wall, while the “See Wall” was displayed 0.5 m

in front of the see-through vision window. Such a setting ensures that the change of gaze vergence will not happen ahead of time. Overall, per subject performed 80 (= (2 techniques × 1 distance + 2 techniques × 3 distances) × 2 steps × 5 repetitions) trials. Each experiment took around 70 min.

5.5 Results

5.5.1 Objective Evaluation Results

Completion Time. We conducted repeated-measures ANOVAs ($\alpha = 0.05$) and post hoc pairwise t-tests to judge whether the average completion time is significantly different across modalities. For each GVC technique, we computed the average completion time of three distances. The results are shown in the left part of Fig. 9. The statistical analysis indicated that the effect of four modalities on completion time was statistically significant ($F(3, 57) = 37.662, p < 0.001, \eta^2 = 0.665$). We found that *SGGaze* was significantly faster than *Click* and *Speech* ($p < 0.001, 0.001$). Besides, *SCGaze* was also significantly faster than the two common modalities ($p < 0.001, 0.001$). In order to get convincing results, we also compared the GVC techniques at distances where users spent the longest completion time with *Click* and *Speech*. We found that the aforementioned significant difference still existed.

The left part of Fig. 12 shows the completion time of each GVC modality at 1, 2, and 3 m distances. There is no significant difference between *SGGaze* and *SCGaze*. We saw that the completion time of both GVC techniques gradually decreased as the distances increased. This was expected because the change of gaze depth at the near range requires a larger rotation amplitude of the eyeballs, which results in taking longer time, while the far range did the opposite.

The number of successes. We performed a repeated-measures ANOVA ($\alpha = 0.05$) to identify whether the number of successes is significantly different across modalities. The result is shown in the right part of Fig. 9. We found it failed to reject the equality of the levels of modalities on the number of successes ($F(3, 57) = 1.243, p = 0.301, \eta^2 = 0.061$). The middle part of Fig. 12 plotted the average number of successes of two GVC modalities averaged across subjects. We found that this metric was invariant to the distance. Overall, these results indicated that users can almost finish the correct operations at the assigned time.

The number of mistakes. We define the false triggering in the waiting state as the mistake mentioned before. It did not occur to the *Click* and *Speech* in the waiting state. We counted the average number of mistakes made by GVC techniques across users at three distances, as shown in the right part of Fig. 12. As expected, the number of mistakes increased with increasing distances. This is because when gaze depth increases exponentially, the accuracy decreases accordingly. We also found no significant effect of the two GVC techniques on the number of mistakes ($F(2, 76) = 0.710, p = 0.495, \eta^2 = 0.018$) but the average values of *SGGaze* are higher than those of *SCGaze* at all three distances. The reason for this difference could be that the stimulus of *SGGaze* distracted the users’ attention, which was reported by some users. We plan to minimize the transparency of the stimulus to reduce the distraction in the future.

5.5.2 Subjective Evaluation Results

Task Load. Repeated-measures ANOVA on the NASA TLX questionnaire demonstrated that four modalities had a significant difference in *mental/physical demand*. The post hoc pairwise t-tests between the modalities were shown in Fig. 11. In general, the *Click* achieved the highest *mental/physical demand* than all the other techniques. Users generally placed the buttons next to their hands. According to the free response, some participants found that clicking buttons required them to look down frequently, which distracted them and degraded the user experience. We also observed that the *Speech* had lower *mental demand* than the *SCGaze*. A few participants reported that they are more familiar with speech-based input than with gaze vergence control. They felt a little nervous when using GVC techniques at the beginning. There is no significant difference in terms of other task loads.

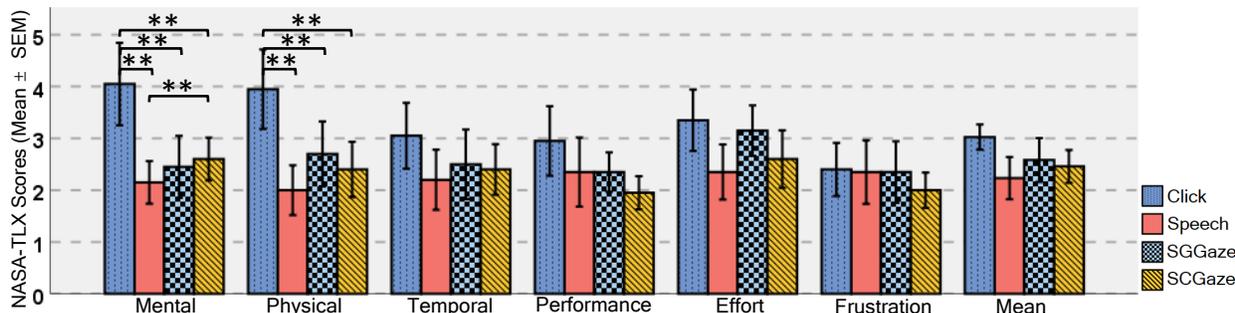


Fig. 11. Bar charts of scores on the NASA-TLX questionnaire for comparing four modalities. The statistical significances are labeled with ** ($p < 0.05$). Error bars mean standard deviations.

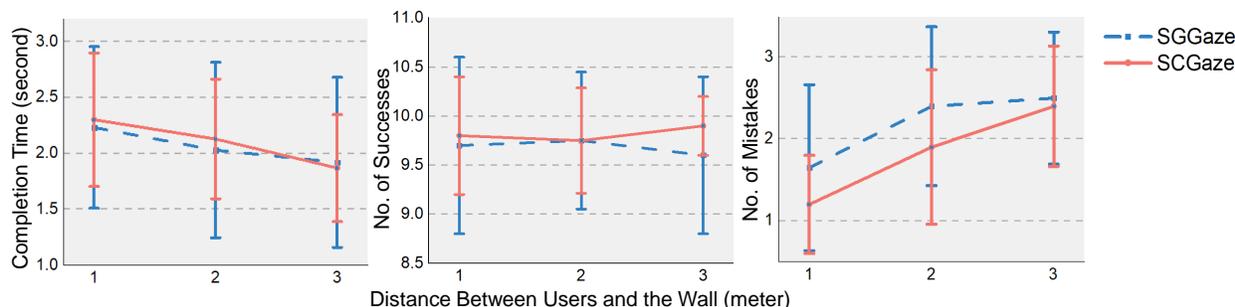


Fig. 12. Line charts of GVC techniques' performance under different measurements. The comparison of completion time (left), the number of successes (center), and the number of mistakes (right). Error bars mean standard deviations.

User preference. According to the results of the preference ranking questionnaire, the *SCGaze* is the most preferred by the users, as shown in Fig. 10. 60% of participants believed *SCGaze* ranked first in terms of preference, 30% of users preferred *SGGaze* the most, and two participants liked the speech-based technique the most.

5.6 Discussion

In this section, we discuss and summarize the results for validating the hypotheses.

H_1 : *Controlling see-through vision has higher efficiency and usability with the GVC techniques than using Click and Speech within the middle distance.*

Our results supported this hypothesis. In terms of speed, *SGGaze* outperformed *Click* and *Speech*; *SCGaze* was also superior to both common techniques. Besides, user's feedback also supposed that "Controlling see-through vision by looking closer or far away rarely requires response time" (P11). For accuracy, although the GVC techniques occasionally occurred with false triggering caused by the accuracy of depth estimation, the number of successes was still not affected.

In terms of usability, we thought that using gaze vergence to control see-through vision was convenient and easy to use. "After simple training, it is relatively simple and has no mental fatigue." (P2) "I feel relaxed using it." (P7) Users reported that "I feel arm fatigue after *Click*" (P3). Some participants claimed that "the speech command needs to speak aloud to trigger the switch, which is not convenient in a quiet space" (P4). We believed that the GVC techniques tackled the limitations of *Click* and *Speech* and improved the user experience. It freed both hands and users did not need to look away. It can also be done without making sounds. P6 and P10 had similar feelings. The above analysis accounts for the superior performance of GVC techniques.

H_2 : *Controlling see-through vision is more intuitive and attractive with the GVC techniques than using Click and Speech within the middle distance.*

Our results supported this hypothesis. We validated it in two aspects. 1) With regard to user preference, 60% of users preferred *SCGaze* and 30% of participants ranked *SGGaze* first. Most of the participants found

SGGaze and *SCGaze* to be enjoyable, e.g., "It is amazing. I have been looking in the same direction, but the change of vergence can convey a signal of seeing through the wall, which is a novel experience for me" (P16). 2) In terms of naturalness, the GVC techniques take advantage of our viewing habit, as when we want to see through something, it is physically associated with our gaze depth/vergence, and therefore should be naturally controlled by the eyes. In contrast, *Click* needs to interrupt the user experience and ask them to look down to press a button. *Speech* requires the participants to repeat boring commands. P4 and P7 also expressed similar opinions. To summarize, we believed that the gaze-vergence-controlled see-through vision is more appealing and intuitive than the common interaction modalities.

6 EXAMPLES FOR GAZE VERGENCE CONTROL

In this section, we demonstrate the GVC technique with four example applications, which can give insights and implications for designers. For more details, please refer to the supplemental video.

See Through a Wall via Gaze Vergence Control. We show how to see through an office wall using the proposed *SCGaze* technique, as shown in Fig. 13a. The user is immersed in the occluded environment with a first-person view and naturally controls the see-through vision with his eyes. In this example, one surveillance camera is attached to the inner wall of an office. The user stands 1 meter away from the wall. When the user fixates on the wall, no see-through vision is activated. When gaze vergence reaches the target depth, see-through vision is triggered. The user sees the television and the moving scene through see-through vision. The user's view is consistent with physical laws.

See Through an Object via Gaze Vergence Control. The second example enables the user to see through daily indoor objects, e.g., a television, with the *SCGaze* modality, as shown in Fig. 13b. This indicates that we can naturally make daily life objects invisible and thus expand our vision. In this example, the participant can see the person through the television. The user is 2 m away from the television and 4.5 m away from the person. When the gaze vergence reaches the depth of the person, the see-through vision is activated. The vision window also helps to keep the eyes focused at a certain distance. Then the user's gaze is put on the television and there is no see-through effect.

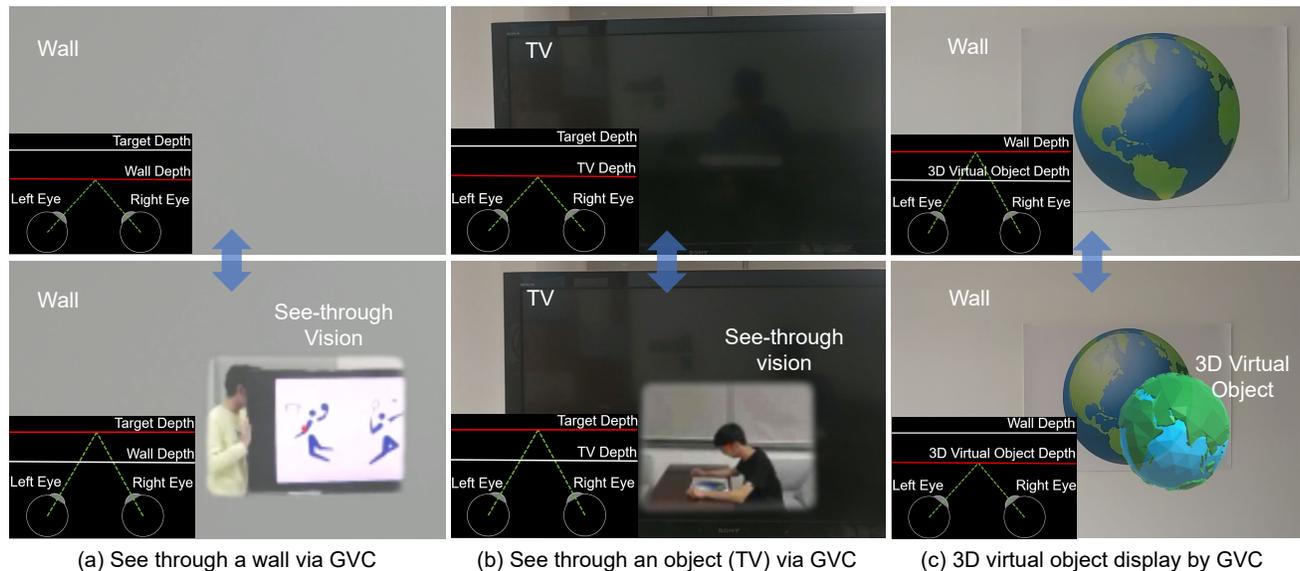


Fig. 13. We present three examples to provide the implications for researchers. (a) See through a wall via gaze vergence control. (b) See through an object via gaze vergence control. (c) 3D virtual object display triggered by gaze vergence.

3D Virtual Object Display Triggered by Gaze Vergence. Our third example shows that the GVC technique can control the display of additional information about a real object of interest. This is potentially an effective channel in maintaining or learning scenarios, *e.g.*, opening the menu when both hands are occupied [33, 51]. We implement a 3D virtual earth display triggered by gaze vergence, as shown in Fig. 13c. In this example, we do not see through but rather see in front. When the gaze target is on the wall, the user sees a real poster, which is located 2 m away from him. Then the user controls the gaze vergence to fixate 0.5 m in front of the wall. As a result, a 3D virtual earth is shown at the fixation position.

See Through Multiple Occluding Objects via Gaze Vergence Control. Our final example shows that the user can see through multiple occluding objects with the vergence control and switch between four layers of different depths. We set 4 layers in this demo: (0, 0.6] m centered at 0.3 m, (0.6, 1.4] m @ 1.0 m, (1.4, 3.0] m @ 2.2 m and (3.0 m, +∞) @ 4.5 m. When the gaze vergence reaches the depth of each layer, the image of each layer is shown in turn. When the user's gaze depth decreases, the see-through vision switches to the front layers and eventually comes back to the first layer. Please see our supplemental video for more details.

7 DESIGN IMPLICATIONS

Based on the results and observations of our study, we derive a set of guidelines and implications for the design of gaze-vergence controlled techniques in AR.

- Our results demonstrate that the proposed 3D LosI and PIPD methods perform differently in the range of (0.5, 6] m. We suggest that for the gaze depth estimation in the range of (0.5, 2] m, the Gaze-Vergence-Controlled (GVC) techniques can utilize the 3D LosI method. For a depth beyond 2 m, the PIPD method can be used.
- Providing an error-tolerant design for the GVC see-through vision can increase its robustness. We suggest setting a distance threshold for the control. Considering the depth estimation error, we use the sum of mean error and standard deviation as the threshold at each distance. Besides, we recommend using a filter function for smoothing depth values.
- The window of see-through vision should be fixed at a certain distance, which helps prevent the window drifting due to gaze error and stabilizes user's gaze depth to avoid frequent gaze adjustment. Meanwhile, the fixed window depth also avoids causing visual fatigue.

- During our experiments, we found that divergence movement can in fact be successfully performed but cannot last long. This is because the fixation points fall behind a wall in an instant, but they cannot be fixed without stimulus and thus come back to the wall plane quickly. Fortunately, the mechanism of our SCGaze can help avoid this problem. Our system can rapidly capture the instant change of vergence and activate see-through vision. The window of see-through vision can serve as a stimulus to help stabilize the user's vergence.

8 SYSTEM LIMITATIONS AND FUTURE WORK

The GVC techniques have higher efficiency and usability than using *Click* and *Speech* for controlling the see-through vision within the middle distance. However, when the distance exceeds 3 m, it is difficult to discriminate the vergence difference as described in Section 4. Therefore, for long-distance interaction (>3 m), we can use the modality independent of the distance, *e.g.*, speech-based technique.

In the future, we will design and implement a shared see-through vision between multiple users controlled by gaze vergence. In the current setting, we embedded a surveillance camera behind the wall to achieve see-through vision. For different users staying in adjacent rooms, we plan to enable them to wear a HoloLens 2. We can thus obtain images from the scene camera of each HoloLens. Each user can use gaze depth to trigger the see-through vision. The images are captured by different devices in adjacent rooms.

9 CONCLUSION

In this work, we proposed using the gaze-vergence-controlled see-through vision in AR. We first built a gaze tracking module with two infrared cameras and assembled it into the Microsoft HoloLens 2. With our gaze depth estimation algorithm, the user's gaze depth can be computed from gaze vergence and used to control the see-through vision. We evaluated the efficiency and usability of four interaction techniques. Experimental results demonstrated that gaze depth estimation is efficient and accurate. It showed that the GVC techniques are superior in terms of efficiency and more preferred by users. We also showed four example applications of GVC see-through vision.

REFERENCES

- [1] 10 Augmented Reality Trends of 2022: A Vision of Immersion. [Online]. <https://mobidev.biz/blog/augmented-reality-trends-future-ar-technologies>, Accessed March 1, 2022.

- [2] Apple vs. Microsoft: Who Will Augment Reality? [Online]. <https://blog.relaycars.com/apple-vs-microsoft-ar>, Accessed March 1, 2022.
- [3] Introducing the next generation of the internet: the metaverse. [Online]. <https://www.trtworld.com/life/introducing-the-next-generation-of-the-internet-the-metaverse-51117>, Accessed March 1, 2022.
- [4] Messagepack. [Online]. <https://github.com/neuecc/MessagePack-CSharp>, Accessed March 1, 2022.
- [5] Microsoft HoloLens 2. [Online]. <https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/input/eye-tracking/eye-tracking-eye-gaze-provider?view=mrtkunity-2021-05>, Accessed March 1, 2022.
- [6] Netmq. [Online]. <https://github.com/zeromq/netmq>, Accessed March 1, 2022.
- [7] S. Ahn, J. Son, S. Lee, and G. Lee. Verge-it: Gaze interaction for a binocular head-worn display using modulated disparity vergence eye movement. In *Conference on Human Factors in Computing Systems*, p. 1–7, 2020.
- [8] F. Alt, S. Schneegass, J. Auda, R. Rzayev, and N. Broy. Using eye-tracking to support interaction with layered 3d interfaces on stereoscopic displays. In *19th International Conference on Intelligent User Interfaces, IUI 2014, Haifa, Israel, February 24-27, 2014*, pp. 267–272. ACM, 2014.
- [9] B. Avery, W. Piekarski, and B. H. Thomas. Visualizing occluded physical objects in unfamiliar outdoor augmented reality environments. In *Sixth IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR 2007, 13-16 November 2007, Nara, Japan*, pp. 285–286. IEEE Computer Society, 2007.
- [10] B. Avery, C. Sandor, and B. H. Thomas. Improving spatial perception for augmented reality x-ray vision. In *IEEE Virtual Reality Conference 2009 (VR 2009), 14-18 March 2009, Lafayette, Louisiana, USA, Proceedings*, pp. 79–82. IEEE Computer Society, 2009.
- [11] R. Bane and T. Höllerer. Interactive tools for virtual x-ray vision in mobile augmented reality. In *3rd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2004), 2-5 November 2004, Arlington, VA, USA*, pp. 231–239. IEEE Computer Society, 2004.
- [12] S. Bardins, T. Poitschke, and S. Kohlbecher. Gaze-based interaction in various environments. In *Proceedings of the 1st ACM Workshop on Vision Networks for Behavior Analysis, VNBA '08*, p. 47–54. Association for Computing Machinery, New York, NY, USA, 2008. doi: 10.1145/1461893.1461903
- [13] P. C. Barnum, Y. Sheikh, A. Datta, and T. Kanade. Dynamic see-throughs: Synthesizing hidden views of moving objects. In G. Klinker, H. Saito, and T. Höllerer, eds., *Science & Technology Proceedings, 8th IEEE International Symposium on Mixed and Augmented Reality 2009, ISMAR 2009, Orlando, Florida, USA, October 19-22, 2009*, pp. 111–114. IEEE Computer Society, 2009.
- [14] M. Billinghurst, H. Kato, and S. Myojin. Advanced interaction techniques for augmented reality applications. In *International Conference on Virtual and Mixed Reality*, pp. 13–22. Springer, 2009.
- [15] G. Casiez, N. Roussel, and D. Vogel. 1 € filter: A simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 2527–2530. New York, NY, USA, 2012.
- [16] J. Chen, Y. Tong, W. D. Gray, and Q. Ji. A robust 3d eye gaze tracking system using noise reduction. In *Proceedings of the Eye Tracking Research & Application Symposium, ETRA 2008, Savannah, Georgia, USA, March 26-28, 2008*, pp. 189–196, 2008.
- [17] Y. Cheng, X. Zhang, F. Lu, and Y. Sato. Gaze estimation by exploring two-eye asymmetry. *IEEE Transactions on Image Processing*, 29:5259–5272, 2020.
- [18] J. L. Demer and R. A. Clark. Functional anatomy of human extraocular muscles during fusional divergence. *Journal of Neurophysiology*, 120(5):2571–2582, 2018.
- [19] K. Dierkes, M. Kassner, and A. Bulling. A fast approach to refraction-aware eye-model fitting and gaze prediction. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research Applications*, pp. 1–9, 2019.
- [20] C. Elmadjian, P. Shukla, A. D. Tula, and C. H. Morimoto. 3d gaze estimation in the scene volume with a head-mounted eye tracker. COGAIN '18, 2018.
- [21] N. Elmquist and P. Tsigas. A taxonomy of 3d occlusion management for visualization. *IEEE transactions on visualization and computer graphics*, 14(5):1095–1109, 2008.
- [22] O. Erat, W. A. Isop, D. Kalkofen, and D. Schmalstieg. Drone-augmented human vision: Exocentric control for drones exploring hidden areas. *IEEE Trans. Vis. Comput. Graph.*, 24(4):1437–1446, 2018.
- [23] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [24] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Mach. Vis. Appl.*, 12(1):16–22, 2000.
- [25] U. Gruenefeld, Y. Brück, and S. Boll. Behind the scenes: Comparing x-ray visualization techniques in head-mounted optical see-through augmented reality. In J. R. Cauchard and M. Löchtefeld, eds., *MUM 2020: 19th International Conference on Mobile and Ubiquitous Multimedia, Essen, Germany, November 22-25, 2020*, pp. 179–185. ACM, 2020.
- [26] S. G. Hart. Nasa-Task Load Index (NASA-TLX); 20 years later. In *Proc. Hum. Factors Ergon. Soc.*, vol. 50, pp. 904–908. California, USA, Oct. 2006.
- [27] T. Hirzle, J. Gugenheimer, F. Geiselhart, A. Bulling, and E. Rukzio. A design space for gaze interaction on head-mounted displays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*, p. 625, 2019.
- [28] Y. Itoh and G. Klinker. Interaction-free calibration for optical see-through head-mounted displays based on 3d eye localization. In *IEEE Symposium on 3D User Interfaces, 3DUI 2014, Minneapolis, MN, USA, March 29-30, 2014*, pp. 75–82. IEEE Computer Society, 2014.
- [29] S. Julier, Y. Baillot, D. G. Brown, and M. Lanzagorta. Information filtering for mobile augmented reality. *IEEE Computer Graphics and Applications*, 22(5):12–15, 2002.
- [30] Y. Kameda, T. Takemasa, and Y. Ohta. Outdoor see-through vision utilizing surveillance cameras. In *3rd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2004), 2-5 November 2004, Arlington, VA, USA*, pp. 151–160. IEEE Computer Society, 2004.
- [31] M. Kassner, W. Patera, and A. Bulling. Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Adjunct Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '14 Adjunct*, pp. 1151–1160, 2014.
- [32] M. Kassner, W. Patera, and A. Bulling. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *The 2014 ACM Conference on Ubiquitous Computing, UbiComp '14 Adjunct, Seattle, WA, USA - September 13 - 17, 2014*, pp. 1151–1160. ACM, 2014.
- [33] K. Kim, M. Billinghurst, G. Bruder, H. B. Duh, and G. F. Welch. Revisiting trends in augmented reality research: A review of the 2nd decade of ISMAR (2008-2017). *IEEE Trans. Vis. Comput. Graph.*, 24(11):2947–2962, 2018.
- [34] D. Kirst and A. Bulling. On the verge: Voluntary convergences for accurate and precise timing of gaze input. In *Conference on Human Factors in Computing Systems*, pp. 1519–1525, 2016.
- [35] S. Kudo, H. Okabe, T. Hachisu, M. Sato, S. Fukushima, and H. Kajimoto. Input method using divergence eye movement. In *Conference on Human Factors in Computing Systems*, pp. 1335–1340, 2013.
- [36] Y.-M. Kwon, K.-W. Jeon, J. Ki, Q. M. Shahab, S. Jo, and S.-K. Kim. 3d gaze estimation and interaction to stereo display. *International Journal of Virtual Reality*, 5(3):41–45, 2006.
- [37] M. Kytö, B. Ens, T. Piumsomboon, G. A. Lee, and M. Billinghurst. Pinpointing: Precise head-and eye-based target selection for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2018.
- [38] Y. Lee, T. Piumsomboon, B. Ens, G. A. Lee, A. Dey, and M. Billinghurst. A gaze-depth estimation technique with an implicit and continuous data acquisition for ost-hmds. In *International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments, ICAT-EGVE 2017, Posters and Demos, Adelaide, Australia, November 22-24, 2017*, pp. 1–2.
- [39] V. Lepetit, F. Moreno-Noguer, and P. Fua. Eppn: An accurate o(n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009.
- [40] K. Liliija, H. Pohl, S. Boring, and K. Hornbæk. Augmented reality views for occluded interaction. In S. A. Brewster, G. Fitzpatrick, A. L. Cox, and V. Kostakov, eds., *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*, p. 446. ACM, 2019.
- [41] Y. Liu, R. Liu, H. Wang, and F. Lu. Generalizing gaze estimation with outlier-guided collaborative adaptation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 3815–3824. IEEE, 2021.

- [42] R. Mantiuk, B. Bazyluk, and A. M. Tomaszewska. Gaze-dependent depth-of-field effect rendering in virtual environments. In M. Ma, M. Fradinho, and J. M. Pereira, eds., *Serious Games Development and Applications - Second International Conference, SGDA 2011, Lisbon, Portugal, September 19-20, 2011. Proceedings*, vol. 6944 of *Lecture Notes in Computer Science*, pp. 1–12. Springer, 2011.
- [43] D. Mardanbegi, C. Clarke, and H. Gellersen. Monocular gaze depth estimation using the vestibulo-ocular reflex. In K. Krejtz and B. Sharif, eds., *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, ETRA 2019, Denver, CO, USA, June 25-28, 2019*, pp. 20:1–20:9. ACM, 2019.
- [44] H. Martínez, D. Skourmetou, J. Hyppölä, S. Laukkanen, and A. Heikkilä. Drivers and bottlenecks in the adoption of augmented reality applications. *Journal of Multimedia Theory and Application*, 2(1), 2014.
- [45] P. Mohan, W. B. Goh, C. Fu, and S. Yeung. DualGaze: Addressing the midas touch problem in gaze mediated vr interaction. In *Proc. IEEE Int. Symp. Mix. Augmented Real.*, pp. 79–84. Munich, Germany, Oct. 2018.
- [46] S. Mori, S. Ikeda, and H. Saito. A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects. *IPSJ Transactions on Computer Vision and Applications*, 9(1):1–14, 2017.
- [47] S. Oney, N. Rodrigues, M. Becher, T. Ertl, G. Reina, M. Sedlmair, and D. Weiskopf. Evaluation of gaze depth estimation from eye tracking in augmented reality. *ETRA '20 Short Papers*, 2020.
- [48] J. Orlosky, T. Toyama, D. Sonntag, and K. Kiyokawa. The role of focus in advanced visual interfaces. *Künstliche Intell.*, 30(3-4):301–310, 2016.
- [49] H. M. Park, S. H. Lee, and J. S. Choi. Wearable augmented reality system using gaze interaction. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 175–176. IEEE, 2008.
- [50] T. Pfeiffer, M. E. Latoschik, and I. Wachsmuth. Evaluation of binocular eye trackers and algorithms for 3d gaze interaction in virtual reality environments. *JVRB-Journal of Virtual Reality and Broadcasting*, 5(16), 2008.
- [51] T. Piumsomboon, A. Day, B. Ens, Y. Lee, G. A. Lee, and M. Billingham. Exploring enhancements for remote mixed reality collaboration. In *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications, Bangkok, Thailand*, pp. 16:1–16:5. ACM, 2017.
- [52] C. Sandor, A. Cunningham, A. Dey, and V.-V. Mattila. An augmented reality x-ray system based on visual saliency. In *2010 IEEE International Symposium on Mixed and Augmented Reality*, pp. 27–36. IEEE, 2010.
- [53] T. Santini, W. Fuhl, and E. Kasneci. Purest: Robust pupil tracking for real-time pervasive eye tracking. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research Applications, ETRA '18*, 2018.
- [54] D. Su, Y.-F. Li, and H. Chen. Toward precise gaze estimation for mobile head-mounted gaze tracking systems. *IEEE Transactions on Industrial Informatics*, 15(5):2660–2672, 2019. doi: 10.1109/TII.2018.2867952
- [55] J. E. Swan, A. Jones, E. Kolstad, M. A. Livingston, and H. S. Smallman. Egocentric depth judgments in optical, see-through augmented reality. *IEEE transactions on visualization and computer graphics*, 13(3):429–442, 2007.
- [56] J. E. Swan, M. A. Livingston, H. S. Smallman, D. Brown, Y. Baillot, J. L. Gabbard, and D. Hix. A perceptual matching technique for depth judgments in optical, see-through augmented reality. In *IEEE Virtual Reality Conference (VR 2006)*, pp. 19–26. IEEE, 2006.
- [57] L. Swirski. *Gaze estimation on glasses-based stereoscopic displays*. PhD thesis, 08 2015.
- [58] K. Takahashi, S. Nobuhara, and T. Matsuyama. A new mirror-based extrinsic camera calibration using an orthogonality constraint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1051–1058, 2012. doi: 10.1109/CVPR.2012.6247783
- [59] T. Toyama, D. Sonntag, J. Orlosky, and K. Kiyokawa. Attention engagement and cognitive state analysis for augmented reality text display functions. *IUI '15*, p. 322–332. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2678025.2701384
- [60] D. W. F. van Krevelen and R. Poelman. A survey of augmented reality technologies, applications and limitations. *Int. J. Virtual Real.*, 9(2):1–20, 2010.
- [61] E. E. Veas, R. Grasset, E. Kruijff, and D. Schmalstieg. Extended overview techniques for outdoor augmented reality. *IEEE Trans. Vis. Comput. Graph.*, 18(4):565–572, 2012.
- [62] M. Vidal, A. Bulling, and H. Gellersen. Pursuits: spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In *The 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 439–448. ACM, 2013.
- [63] M. Vidal, D. H. Nguyen, and K. Lyons. Looking at or through? using eye tracking to infer attention location for wearable transparent displays. In *Proceedings of the 2014 ACM International Symposium on Wearable Computers, ISWC '14*, p. 87–90. Association for Computing Machinery, New York, NY, USA, 2014. doi: 10.1145/2634317.2634344
- [64] L. Wang, J. Wu, X. Yang, and V. Popescu. VR exploration assistance through automatic occlusion removal. *IEEE Trans. Vis. Comput. Graph.*, 25(5):2083–2092, 2019.
- [65] Z. Wang, H. Wang, H. Yu, and F. Lu. Interaction with gaze, gesture, and speech in a flexibly configurable augmented reality system. *IEEE Transactions on Human-Machine Systems*, 51(5):524–534, 2021.
- [66] Z. Wang, H. Yu, H. Wang, Z. Wang, and F. Lu. Comparing single-modal and multimodal interaction in an augmented reality system. In *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 165–166. IEEE, 2020.
- [67] Z. Wang, Y. Zhao, Y. Liu, and F. Lu. Edge-guided near-eye image analysis for head mounted displays. In *IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2021, Bari, Italy, October 4-8, 2021*, pp. 11–20. IEEE, 2021.
- [68] Z. Wang, Y. Zhao, and F. Lu. Control with vergence eye movement in augmented reality see-through vision. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops*, pp. 548–549. IEEE, 2022.
- [69] M. Weier, T. Roth, A. Hinkenjann, and P. Slusallek. Predicting the gaze depth in head-mounted displays using multiple feature regression. In B. Sharif and K. Krejtz, eds., *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, ETRA 2018, Warsaw, Poland, June 14-17, 2018*, pp. 19:1–19:9. ACM, 2018.
- [70] M. Wu and V. Popescu. Efficient VR and AR navigation through multi-perspective occlusion management. *IEEE Trans. Vis. Comput. Graph.*, 24(12):3069–3080, 2018.
- [71] D. Yu, Q. Zhou, J. Newn, T. Dingler, E. Velloso, and J. Gonçalves. Fully-occluded target selection in virtual reality. *IEEE Trans. Vis. Comput. Graph.*, 26(12):3402–3413, 2020.
- [72] H. Zhou, Z. Ren, and K. Zhou. Adaptive geometric sound propagation based on a-weighting variance measure. *Graphical Models*, 116:101109, 2021. doi: 10.1016/j.gmod.2021.101109