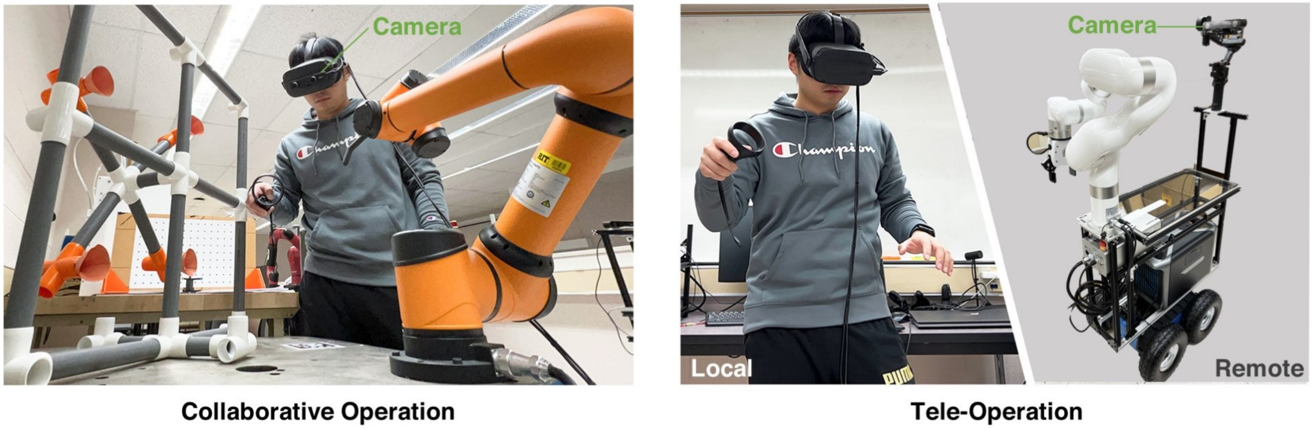
A Global Correction Framework for Camera Registration in Video See-Through Augmented Reality Systems

Wenhao Yang
Department of Industrial
and Systems Engineering
Rochester Institute of Technology
Rochester, New York 14623, USA
Email: wy7711@rit.edu

Yunbo Zhang[*]
Department of Industrial
and Systems Engineering
School of Information (affiliated)
Rochester Institute of Technology
Rochester, New York 14623, USA
Email: ywzeie@rit.edu



Collaborative Operation

Tele-Operation

Fig. 1. Two types of applications of Augmented Reality in robot operation tasks are illustrated: collaborative operations at the shared workspace and teleoperation from a distance.

Augmented Reality (AR) enhances the user's perception of the real environment by superimposing virtual images generated by computers. These virtual images provide additional visual information that complements the real-world view. AR systems are rapidly gaining popularity in various manufacturing fields such as training, maintenance, assembly, and robot programming. In some AR applications, it is crucial for the invisible virtual environment to be precisely aligned with the physical environment to ensure that human users can accurately perceive the virtual augmentation in conjunction with their real surroundings. The process of achieving this accurate alignment is known as calibration. During some robotics applications using AR, we observed instances of misalignment in the visual representation within the designated workspace. This misalignment can potentially impact the accuracy of the robot's operations during the task. Based on previous research on AR-assisted robot programming systems, this work investigates the sources of misalignment errors and presents a simple and efficient calibration procedure to reduce the misalignment accuracy in general video see-through AR systems. To accurately superimpose virtual information onto the real environment, it is necessary to identify the sources and propagation of errors. In this work, we outline the linear transformation and projection of each point from the virtual world space to the virtual screen coordinates. An offline calibration method is introduced to determine the offset matrix from the Head-Mounted Display (HMD) to the camera, and experiments are conducted to validate the improvement achieved through the calibration process.

[*] Corresponding author.

Nomenclature

In this paper, we use the following nomenclature. Capital letters denote vectors and Boldface capital letters denote matrices. It is common to use a specific letter, such as "T" or "M", to represent a general transformation matrix. Lowercase letters denote scalar values. Given coordinate systems A and B, the transformation from A to B is defined by ${}^{A}_{B}\mathbf{M}$, where ${}^{A}_{B}\mathbf{M}$ is the transformation. The unit of translation in the transformation matrix is the meter.

1 Introduction

The Augmented Reality (AR) technique superimposes computer-generated graphics onto the physical environment, such as texts, 3D graphics, and animations [1]. AR can be classified into three categories based on the hardware devices used [2]: hand-held device-based, head-mounted display (HMD)-based, and projector-based. The findings from [3, 4] indicated that wearable devices, particularly see-through HMDs, not only provide greater accuracy but also deliver an immersive user experience that significantly enhances human perception and interaction in both physical and virtual environments. The advantages of HMD AR can be attributed to the wearable system's ability to overlay virtual instructions directly into the operator's field of view (FOV), which provides a self-centered perspective and an egocentric viewpoint [5, 6]. Furthermore, according to [7], the HMD AR demonstrates mobility and hands-free capabilities, particularly in the context of industrial applications. HMD AR has been effectively deployed across diverse manufacturing domains, encompassing areas such as training [8], maintenance [9], design [10], assembly [11], human-robot interaction [12], and remote assistance [13].

AR techniques have been effectively employed in robot programming tasks, enabling both collaborative operations at the same location and teleoperation from a distance, as illustrated in Figure 1. A prevalent scenario involves the utilization of AR for robot programming in a shared workspace where collaborative tasks are carried out in close proximity. The integration of AR in robotics has demonstrated its potential to enhance the accuracy and efficiency of robot operations by providing supplementary information and visual aids [14]. AR merges the virtual world with the physical environment, allowing human operators to interact in the virtual realm by setting virtual waypoints for the Tool Center Point (TCP) and defining a Collision-Free Volume (CFV) [15, 12]. Subsequently, the robot perceives these inputs based on the augmented environment. However, the successful execution of robot actions relies on the accurate interaction and placement of virtual objects by humans with respect to the physical environment, as any misregistration in the AR scene can introduce significant errors caused by human factors [16]. In other words, the interaction of virtual objects is intricately interconnected with the process of perception [17]. Therefore, the effective implementation of AR in robotics necessitates precise alignment and human perception of computer-generated graphics within the physical environment.

For well-registered AR scenes, the spatial co-localization of virtual objects and the physical environment should be achieved, and these virtual objects should be interacted with as they are in the physical environment. Thus, there is a requirement for the virtual elements to register in the physical environment and remain stable from different points of view. The precise alignment of computer-generated objects and panels with the physical world in AR systems is referred to as *registration*. In other words, *registration* is to optimally align two or more rigid bodies by estimating the best transformation between them [18]. The *calibration* is the process that enables this goal and ensures precise virtual widgets and animations in AR operations. Nevertheless, misregistration can arise from disparate AR systems and a multitude of factors.

Presently, commercial AR HMD systems available on the market comprise Optical See-Through (OST) HMDs and Video See-Through (VST) [19]. While an OST HMD allows users to directly see the physical world through a transparent screen and the virtual content displayed on the screen simultaneously, the VST HMD displays the whole scenario with the physical world captured by cameras and integrated with virtual content [20]. The primary market-available AR HMDs generally fall into these two categories, such as Google Glass (OST), HoloLens (OST), Magic Leap (OST), Meta Quest Pro (VST), Varjo XR-3 (VST), and the recently released Apple Vision Pro (VST). OST HMDs usually provide a better depth perception of the real world, but they suffer from the limited FOVs and challenges of occlusion handling [21]. On the other hand, VST HMDs have outstanding rendering features to handle the occlusion and the consistency between the real and synthetic views [21]. Both OST and VST HMDs face challenges related to misregistration according to their different visualization methods. One essential factor contributing to misregistration is depth perception, which has been well-recognized based on the established optical models of different HMDs. There are many factors other than depth that may also affect the alignment of human perception and the virtual content, including camera pose, head pose, eyeball pose, eye focal length, and image plane position. Nevertheless, it is unclear how these factors affect the correct registration of virtual content in the physical world.

In our previous work [22, 23], registration errors were observed in the VST AR interface where 3D virtual objects appeared to float differently based on the observer's viewpoint, as shown in Figure 2. This issue caused significant errors in robot programming as user-defined virtual waypoints shifted, leading to confusion during task completion. An error propagation model was developed in our previous work [24], categorizing the error sources into seven categories, including camera extrinsic calibration, virtual object alignment, display, tracking, camera intrinsic calibration, rendering, and tracker alignment errors. However, a research question is still open: *what are the specific impacts of these error sources on misregistration in a VST AR system?* To answer this research question, we analyzed the error sources in VST AR systems in detail, established a mathematical model of errors, proposed a calibration method to reduce the errors, and conducted both qualitative and quantitative evaluations of the error model and the calibration method. The detailed contributions are listed as follows:

1. A detailed analysis of the error sources in the VST AR systems with insights into the factors affecting misregistration.
2. A mathematical model that effectively captures and explains the impact of errors on the overall system performance, specifically focusing on the HMD-to-Camera transformation.
3. A global calibration method to rectify the identified errors and improve the alignment accuracy between virtual objects and the physical environment.
4. Qualitative and quantitative evaluations to validate the effectiveness of the mathematical model and the calibration method, ensuring the reliability of the calibration process and its impact on misregistration reduction.

In our study, instead of utilizing commercially available VST AR HMDs, we developed a custom system by connecting an Oculus VR HMD with a Zed mini stereo camera via a computer. This system bears similarities to existing setups, as described in [25]. A key advantage of our approach lies in its cost-effectiveness and wider applicability compared to market-available devices like the Meta Quest Pro and Varjo XR-3. Our system framework is particularly versatile since it relies on computer connectivity to both the stereo camera and HMDs, making it suitable for various applications such as collaborative operations (depicted on the left side of Figure 1) and teleoperations (depicted on the right side of Figure 1). In teleoperations, the user wearing an HMD needs to access a robot from a distance and perceive the environmental information captured by the stereo camera placed in the remote environment. Thus, in this scenario, the stereo camera has to be separate from the HMD, and the market-available HMDs with built-in stereo cameras are not applicable. Actually, in teleoperations, the remote camera is controlled and moved on a gimbal to synchronize with the motion of the HMD, which indeed requires calibration between different devices. It is also worth mentioning that commercial HMDs also require pre-calibration, which is kept undisclosed as their intellectual property. Our research objective is not centered around comparing registration accuracy with commercial HMDs. Instead, we aim to investigate the influence of error sources on registration accuracy, focusing on understanding their impacts rather than evaluating the calibration algorithm itself.

The structure of the paper is as follows: In Section 2, the relevant literature on addressing the registration problem in general AR systems is reviewed. Section 3 provides an overview of the applied AR system and identifies the sources of error in this model based on a pilot study. In Section 4, a mathematical model is proposed to explain how errors affect the results, and Section 5 presents a global calibration method for correcting registration errors. The implementation and results of the calibration are described in Section 6, and the conclusion and potential future directions are outlined in Section 7.

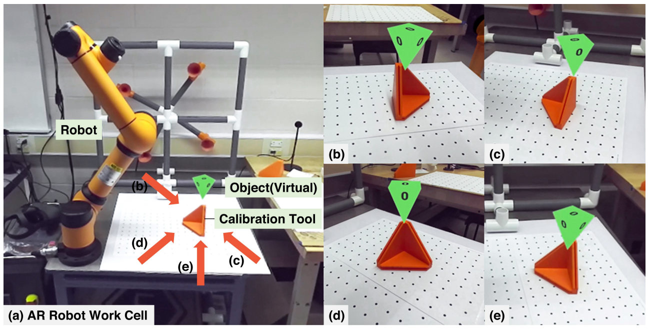


Fig. 2. (a) Collaborative AR robot work cell featuring a virtual flip pyramid as a visualized robot TCP waypoint. (b-d) demonstrate the observed misregistration between the virtual widget and the physical calibration tool from varying viewpoints.

2 Related Works

HMDs that provide immersive user experiences, also referred to as Mixed Reality (MR) displays, can be broadly classified into two categories based on their approaches, as outlined in the survey [2]. The first category is Video See-Through Augmented Reality (VST AR), where users indirectly perceive their physical surroundings through an augmented video feed from cameras. This AR system continuously captures real-world image frames through cameras and superimposes computer-generated graphics as virtual content onto these frames. Users view the augmented image frames on a screen or display. On the other hand, the second category is Optical See-Through Augmented Reality (OST AR), where users have a direct view of the physical world through semi-transparent displays or optical elements, while simultaneously, virtual content is overlaid onto the screen. OST AR devices typically utilize waveguides or holographic optical elements to project virtual content seamlessly into the user's perception, creating the illusion that it is an integral part of the real world. The primary distinction between VST AR and OST AR lies in their approaches to combining and presenting virtual information to users. The OST AR system exhibits certain limitations, including a restricted augmentable field of view, device obtrusiveness, the requirement for frequent HMD recalibrations, the low luminance of the microdisplays, and potential perceptual conflicts [6].

In the past decade, a significant number of researchers have devoted their efforts to tackling the perceptual issues inherent in wearable AR devices. Depth perception is a crucial issue in affecting interaction [26]. Diaz et. al. [27] reviewed design factors that influence depth perceptions in the context of see-through AR HMDs. The utilization of depth cues [28], which leads to accurate estimation in monoculars or binoculars, allows objects to appear in their intended spatial positions in augmented reality applications. Currently, AR HMDs offer the capability to harness binocular cues by rendering two distinct images, simulating the left and right eye perspectives of the virtual object within the real-world context [27]. This approach enables the creation of a visual experience that mimics the natural binocular vision of human observers. Therefore, extensive research is dedicated to addressing the issue of cue loss in AR content, including critical depth cues such as binocular disparity, binocular convergence, accommodative focus, relative size, and occlusions [29].

In the realm of stereoscopic VST AR HMDs, the user's perception of the 3D world is reliant upon the interplay between two distinct optical systems: the acquisition camera and visualization display [30]. To achieve orthostereoscopic functionality in binocular VST HMDs, researchers have put forth several crucial conditions [31, 32]. These include aligning the center of projection for both the cameras and displays with the centers of projection of the user's eyes, precise alignment of the left and right optical axes of the displays with the corresponding optical axes of the cameras, equating the distances between the left and right cameras as well as between the left and right displays to the observer's Inter-Pupillary Distance (IPD), and ensuring that the field of view (FOV) of the displays aligns with the FOV of the cameras. However, commercially available HMDs do not presently offer a means to achieve a genuine orthoscopic view devoid of geometric aberrations [17].

One limitation arises from the necessity for both the cameras and lenses to physically converge at the focal point, resulting in a toed-in HMD configuration [17]. However, commonly available VST HMDs adopt a parallel setup with fixed cameras and lenses, which introduces geometric aberrations, including diplomatic vision. Nevertheless, techniques have been developed to mitigate the impact of these geometric distortions [32, 33]. Therefore, numerous studies have employed custom-made VST AR devices within the research domain to achieve a stereoscopic and egocentric solution. For instance, the ZED Mini from Stereolabs [34], recognized as the world's first external camera designed specifically for AR applications [35], has found widespread use in various research areas [36, 37, 38, 39]. However, most studies directly attach the camera to the VR HMD using a manufacturer mount without calibration, which introduces an additional camera extrinsic error for different HMDs. Thus, a calibration or correction process becomes necessary for such a VST AR system.

In AR devices, calibration is required to mitigate geometrical distortions resulting from the HMDs. This process involves estimating the camera's parameters and aligning various coordinates to ensure accurate correspondence between virtual and physical objects [40]. Grubert [41] provides a comprehensive summary of calibration procedures in OST AR systems according to manual, semiautomatic, and automatic approaches. Manual calibration methods often involve aligning a target object or 2D marker, which introduces the possibility of input errors during subjective operations. In manual categories, alignment setups can be categorized as either environment-centric or user-centric. In the environment-centric setup, targets are positioned at predetermined locations [42]. Users are required to change the line-of-sight or on-screen reticle to align the target. Conversely, in the user-centric alignment, the user stays at the static location and line-of-sight, while the target is movable. However, the calibration process is susceptible to a significant number of errors due to human factors, necessitating the evaluation through user studies [43].

VST AR systems encounter a similar challenge. Interacting in AR environments through VST HMD devices poses an essential issue due to the discrepancy between the human eye and the camera's intrinsic parameters (e.g., resolution limitations and optical distortions), thereby impeding accurate estimation of egocentric distances [29, 33]. Since this problem is hard to solve, many researchers try to align the object on the image to approximately achieve registration. Given the complexity of this problem, numerous researchers strive to address it by attempting to align the object in the image to achieve approximate registration. The 3D registration problem is normally known as the Simultaneous Pose and Correspondence problem (or SPC problem). Many works are proposed to solve the SPC problem using EM-type algorithms, which apply an alternative minimization procedure. In this domain, some researchers focus on the Iterative Closest Point (ICP) algorithm [44], [45], [46], SoftAssign algorithm [47], and variants [48]. However, EM-type algorithms show shortcomings like local minima and rely on good initialization. Li and Hartley [18] propose a global search solution for 3D-3D registration problems. In fact, the broad applicability of these algorithms in AR robot applications is limited, mainly due to their reliance on environmental information. For example, these algorithms typically necessitate a known object with 3D model information acting as a fiducial marker, which demands precise data collection and pre-training. Moreover, The calibration methods discussed in this context rely on the presence of specific physical objects for real-time calibration. However, a notable challenge arises when the target object is no longer within the field of view, as it can lead to substantial errors in the calibration process.

Fuhrmann et al. [49] introduced a rapid calibration method suitable for optical see-through and video-based AR head-mounted displays (HMDs) with a straightforward implementation. Subsequently, researchers have explored automated vision-based verification and alignment methods, often employing fiducial markers during the calibration process [50]. Hu et al. [51] proposed a unique approach by calibrating misregistration using the bare hand as the alignment target. Nevertheless, various calibration methods have relied on user alignment of a target object or 2D marker, which introduces potential input errors due to the object's 6 degrees of freedom (6-DOF). As a result, the calibration process needs to account for human perception. Therefore, it is imperative to identify the factors that contribute to misregistration and thoroughly investigate their effects and consequences.

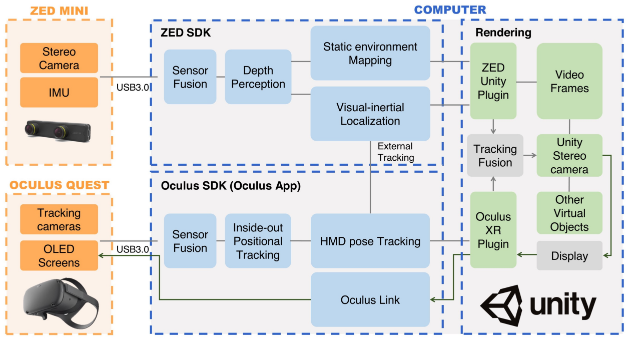


Fig. 3. Video See-Through Augmented Reality (VST-AR) systems components and framework.

3 System Implementation and Registration Framework

In our previous research work [24, 23], we identified registration errors occurring within the AR scene during the operation of a human-robot AR interface. The observed phenomenon involved the floating of 3D virtual objects based on different viewpoints, as depicted in Figure 2. Essentially, the expected stationary nature of the virtual content was compromised by shifting movements corresponding to the observer's motion. This issue can significantly impact robot programming, as the user-defined virtual waypoints undergo shifts that may cause confusion in the user's perception and hinder task completion.

We implemented a VST AR system comprising a stereo camera, the ZED mini, and a Virtual Reality (VR) headset, the Meta Quest 1st generation. The system framework is illustrated in Figure 3. The Meta Quest 1st generation headset offers a diagonal field of view (FoV) of 115° and a resolution of 2880 × 1600 pixels (1440 × 1600 per eye). The ZED mini, mounted on the front of the headset with manufacturer accessories, captures real image frames from the physical environment for AR video passthrough. The HMD is streamed using the ZED mini, which provides a 104° horizontal FoV and a resolution of 2560 × 720 pixels. The Oculus Quest headset utilizes the inside-out tracking system, Oculus Insight, to track its motion. The main system operates on the Unity 3D platform of a PC, serving as the graphic engine. A virtual camera is synchronized with the real camera's movement to capture virtual image frames, generated by the computer graphics system based on the 3D virtual environment. The real and virtual images are merged with occlusion properties using depth information in the Unity rendering pipeline, and the resulting AR images are visualized on the HMD. In this system, the stereo camera functions as two separate monocular systems, enabling the realization of binocular disparity and providing an immersive user experience. The system was tested on a laptop with the following specifications: Intel(R) Core(TM) i7-9850H @ 2.60GHz processor, 16GB of RAM, NVIDIA Quadro RTX 4000 GPU, and Windows 10 Enterprise operating system.

To investigate the registration error further, we implemented a small virtual ball at the tip of a calibration tool, as depicted in Figure 4 (a). The ball serves as a registration marker, aligning with the tip of the calibration tool in a hypothetical scenario where they remain stationary regardless of the observer's viewpoint. The observer remains still at a distance of 0.5 *m*, except for rotating their head to allow the registration marker to appear in different zones of the AR image. Subsequently, the virtual ball exhibits movement and fails to align with the calibration tool, as observed in Figure 4 (b-d). Based on our analysis of error propagation [24], we attribute the primary source of the misregistration error to the camera's extrinsic error. From the observation of misregistration, we hypothesize that *the camera's extrinsic error undermines the final AR synthesis*.

To gain a comprehensive understanding of and mitigate the registration errors, a registration framework for Video See-Through Augmented Reality is proposed. This framework encompasses various modules, as illustrated in Figure 5. In a typical VST AR system, all devices need to be registered within a common reference coordinate, denoted as the world coordinate in Figure 5 (a). While the camera intrinsic parameters play a role, the primary source of extrinsic errors lies in the relative transformation (***V*** in Figure 5 (a)), which affects the generation of accurate virtual content images. This relative transformation is influenced by the registration of each element in the world coordinate.

For general applications, a concise and effective approach is to employ the HMD tracking coordinate as the reference frame, as depicted in Figure 5 (b). Since the camera is fixed on the HMD, the Camera-to-Tracking transformation can be shared with the HMD tracking system, represented as the H transformation, accompanied by a fixed offset transformation denoted as D. While the virtual objects (such as waypoints) are defined by the transformation F based on the user's perspective, the accuracy of the relative transformation A may be compromised, resulting in misregistration and an inadequate representation of virtual objects.

This visualization holds significant importance in AR-based robot programming applications within the context of Digital Twin (DT) and intelligent manufacturing. In such applications, the robot's programmed motion is determined by the C translation between the virtual robot base and virtual waypoints, which are represented in the robot space. It is worth noting that the accuracy of this translation relies on the F and G transformations, with G being dependent on the I transformation and F being influenced by the camera's extrinsic calibration. Consequently, the precision of robot programming is directly impacted by the accuracy of the visualization.

Hence, achieving dynamic registration of Camera-to-Tracking is crucial, and this transformation can be propagated through two separate transformations, D and H. Due to the H transformation that can be obtained from the HMD tracking system, it is imperative to conduct a pre-calibration for obtaining the HMD-to-Camera transformation, which serves as a fixed offset transformation when the camera is mounted on the HMD. Although this problem could be solved by the manufacturer integrating the camera and HMD, it still remains a problem for applications if the camera cannot be integrated into the HMD. For example, as shown in the right of Figure.1, in the AR-based teleoperation system, the camera is near the robot, which is remote to the user who wears an HMD, and the camera extrinsic error still needs to be overcome. In the scenario of a remote AR system, the camera's registration with another tracking system is accomplished through the L transformation. By calibrating the two tracking systems with the J transformation, the camera's absolute extrinsic parameters can be determined. The key transformation and corresponding calibration processes are summarized in Table 1.

In essence, the registration problem boils down to accurately propagating transformations. To simplify the aforementioned registration framework, a key aspect of achieving precise registration lies in the camera-to-tracking transformation, which is the synchronization of virtual cameras with their real counterparts. Figure 6 illustrates the transformation model of the virtual cameras. The base origin represents the virtual world origin, and the virtual objects are positioned in this world coordinate system through the *World-to-Object* transformation. The HMD tracking center is also tracked in the virtual world coordinates, enabling the acquisition of the *World-to-HMD* transformation via the tracking system. This transformation provides the relative pose of the HMD, including its position and orientation, in the virtual world system. The *HMD-to-Camera* transformation remains fixed, as the real camera is rigidly mounted on the HMD. Significantly, the pose of the virtual camera with respect to the virtual world coordinates is determined by two transformations: *World-to-HMD* and *HMD-to-Camera*. Each virtual camera then generates a perspective projection of the objects onto an image plane to produce a 2D virtual image. The *Camera-to-Image* transformation, which encompasses intrinsic parameters such as the optical center, focal length, field of view, and lens distortions, represents the non-linear mapping involved in this process. Drawing upon the preceding discussion, we have classified the sources of misregistration between virtual and real images into four distinct categories:

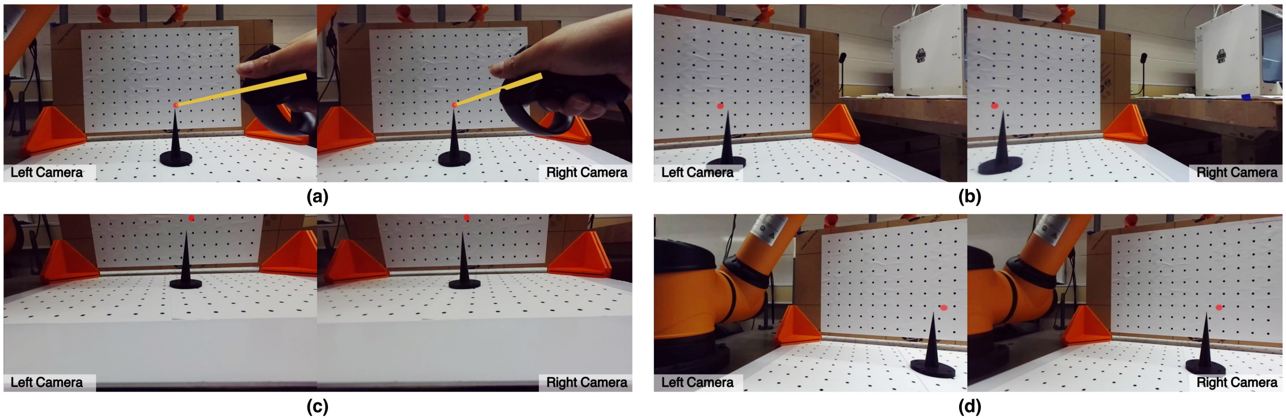


Fig. 4. A misregistration case in a stereo video see-through AR system. (a) Initially, a stationary red virtual bass is placed on the tip of a physical calibration tool. (b-d) Keep the viewer rotating in place, which leads to the ball and calibration tool moving to different positions on the image. The ball moves in different directions and doesn't align with the calibration tool.

Table 1. Key transformations in the VST AR Registration Frameworks

Transformation	**Description**	**Nature**	**Calculation Process**
Camera intrinsics	Camera optical properties	fixed	Camera calibration
A, **B**, **V**	Camera extrinsic	varying	Registrate to a common reference
C	Objects(e.g. waypoints) in the robot coordinate	varying	Registrate to a common reference
D	Camera-to-HMD	fixed	Manufacturer setting or calibrated
E	Camera-to-World	varying	Depend on **D** and **H**
F	Obects-to-World	varying	Define by users
G	Virtual-robot-to-World	fixed	Depend on **I**
H	HMD-to-World	varying	From tracking system
I	Robot-to-World	fixed	Manually measurement
J	Tracking-to-Tracking	varying	Need registration
K	Other-cameras-to-World	varying	Depend on **J** and **L**
L	Camera-to-other-tracking	varying	If the camera is movable, should be tracked. Otherwise, the static transformation should be calibrated.

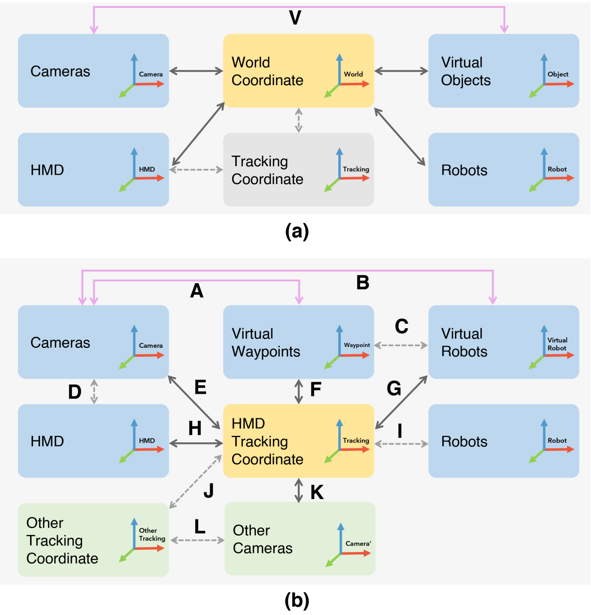


Fig. 5. Registration Frameworks in VST AR and AR-based Robot Applications. (a) In a typical VST AR registration framework, all devices are aligned and registered in a world coordinate, serving as a unified reference coordinate. (b) For AR-based robot applications, a general and efficient registration framework is proposed. In this framework, all devices are calibrated and synchronized to the HMD tracking coordinates, establishing a common reference frame.

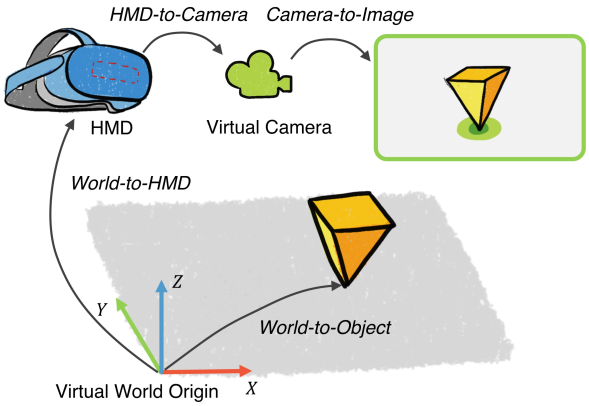


Fig. 6. The registration framework simplifies the process of mapping virtual objects in video-based augmented reality systems onto images using a virtual camera.

1. **Inaccurate placement of virtual objects in the virtual world**: In many AR-based robot programming applications, the entire robot work cell serves as the shared space. Without physical objects or features as registration targets, virtual objects rely on accurate *World-to-Object* transformations. If these transformations are imprecise, the corresponding objects will appear misaligned in the image.
2. **Inaccurate *World-to-HMD* transformation**: This error has two components. First, tracking errors in the tracking system used to determine the HMD's pose introduce imprecision. These errors depend on the tracking system and distance. Second, misalignment between the origins of the tracking system and the virtual world system further contributes to inaccuracies.
3. **Lack of synchronization between the virtual camera and the real camera's movement**: This error arises from two factors. Firstly, the previous imprecise *World-to-HMD* transformation leads to an inaccurate virtual camera position in the virtual world system. Secondly, the relative offset transformation from the HMD to the camera introduces inaccuracy.
4. **Errors in the *Camera-to-Image* mapping**: The perspective projection of real cameras is typically described using a pinhole camera model with distortion. Inaccurate intrinsic parameters representing this camera model result in the misregistration of pixel positions in the 2D image.

As there are no physical objects available for aligning with virtual objects in our setup, users are granted the freedom to manipulate virtual objects in real 3D space. Consequently, the determination of the *World-to-Object* transformation for each individual and dependent object relies on user-defined parameters. Hence, Error Source 1 does not exist since the *World-to-Object* transformation is given by the users. Error Source 2 is currently being addressed through hardware development, assuming the reliability of the implemented tracking system. As for Error Source 3, camera intrinsic calibration methods have been extensively studied for resolving the Camera-to-image transformation. Based on observations of misregistration, we hypothesize that *the camera's extrinsic error undermines the final AR synthesis*. Consequently, this research primarily focuses on calibrating the *HMD-to-camera* transformation.

4 Mathematical Model

To examine the impact of the *HMD-to-camera* transformation on the resulting misregistered image, we propose a mathematical model to elucidate how registration errors arise based on the transformation model illustrated in Figure 7. According to the model, the primary source of error lies in the unsynchronized movement of the virtual camera, resulting in incorrect extrinsic parameters for generating the virtual images. Utilizing the pinhole camera model [52, 53], the conversion of the 3D world coordinate points $P(X_w, Y_w, Z_w)$ to 2D image pixel coordinates $P(U, V)$ can be expressed as in Eq. 1, where $\mathbf{K}$ represents the intrinsic parameters matrix, and $[\mathbf{R} \mid \mathbf{T}]$ denotes the extrinsic parameter matrix.

$$Z_c \begin{bmatrix} U \\ V \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{T}_{3\times1} \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{1}$$

$$\mathbf{K} = \begin{bmatrix} f_x & s & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

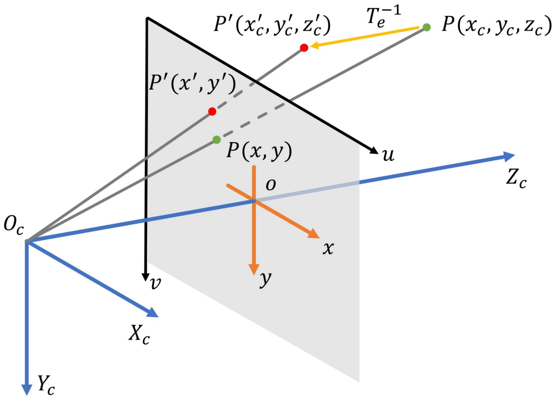


Fig. 7. The conversion from the camera coordinate system to the image coordinate system is performed based on the pinhole camera model [52, 53].

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{T}_{3\times1} \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{3}$$

In the camera intrinsic matrix (Eq.2), the focal lengths in pixel units are denoted as f_x and f_y, the skew coefficient is represented by s, and the coordinates of the principal point are given as u_0 and v_0. Utilizing the camera extrinsic matrix, the 3D world coordinate points $P(X_w, Y_w, Z_w)$ can be transformed into camera coordinates $P(X_c, Y_c, Z_c)$ as described in Eq.3.

However, if an incorrect camera extrinsic matrix is used with an error transformation $\mathbf{T}_e$, a different position p' in the camera coordinate system is obtained (Eq. 4), leading to a varied projection on the image coordinate system as shown in Eq. 5. Here, $\mathbf{T}_e^{-1}$ represents the inverse of the error transformation in the camera coordinate system.

$$\begin{bmatrix} X_c' \\ Y_c' \\ Z_c' \\ 1 \end{bmatrix} = \mathbf{T}_e^{-1} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{4}$$

$$Z_c' \begin{bmatrix} U' \\ V' \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X_c' \\ Y_c' \\ Z_c' \\ 1 \end{bmatrix} \tag{5}$$

The transformation $\mathbf{T}_e$ is a 4×4 homogeneous matrix comprising six parameters, specifically three for translation and three for rotation. To investigate the influence of registration, we narrow our focus to a single parameter and consider six different factors: translation ($\mathbf{T}_{TX}$, $\mathbf{T}_{TY}$, $\mathbf{T}_{TZ}$) and rotation ($\mathbf{T}_{RX}$, $\mathbf{T}_{RY}$, $\mathbf{T}_{RZ}$) along the XYZ axis of the camera, as summarized in Table 2. These factors individually and significantly contribute to the misregistration results within the implemented video-based Augmented Reality system.

4.1 Translation errors along the Z-X-Y axis

The assumption is made that the camera experiences translation errors, encompassing three distinct conditions denoted as $\mathbf{T}_{TZ}$, $\mathbf{T}_{TX}$, and $\mathbf{T}_{TY}$, which are detailed in Table 2. We first examine the scenario where there is a translation error along the Z-axis represented by $\mathbf{T}_{TZ}$. For simplicity, we make the assumption of no skewness, represented by $s = 0$ in Equation 2. Utilizing equations 1, 2, 4, and 5, the relative relationship between the actual pixel position (U_{TZ}', V_{TZ}') and the ideal pixel position (U, V) can be derived as Eq. 6.

$$\begin{cases} U_{TZ}' = \dfrac{Z_c}{Z_c - e_z} U - \dfrac{e_z}{Z_c - e_z} u_0 \\ V_{TZ}' = \dfrac{Z_c}{Z_c - e_z} V - \dfrac{e_z}{Z_c - e_z} v_0 \end{cases} \tag{6}$$

The ratio K_z is defined as the quotient of e_z (the signed magnitude of the translation) and Z_c (the magnitude of the depth projection along the Z-axis), as shown in eq. 7. The sign indicates the direction, and the unit of measurement is in meters. Consequently, the actual pixel position (U_{TZ}', V_{TZ}') is determined by the ideal pixel position (U, V) and the ratio K_z.

$$K_z = \frac{e_z}{Z_c} \tag{7}$$

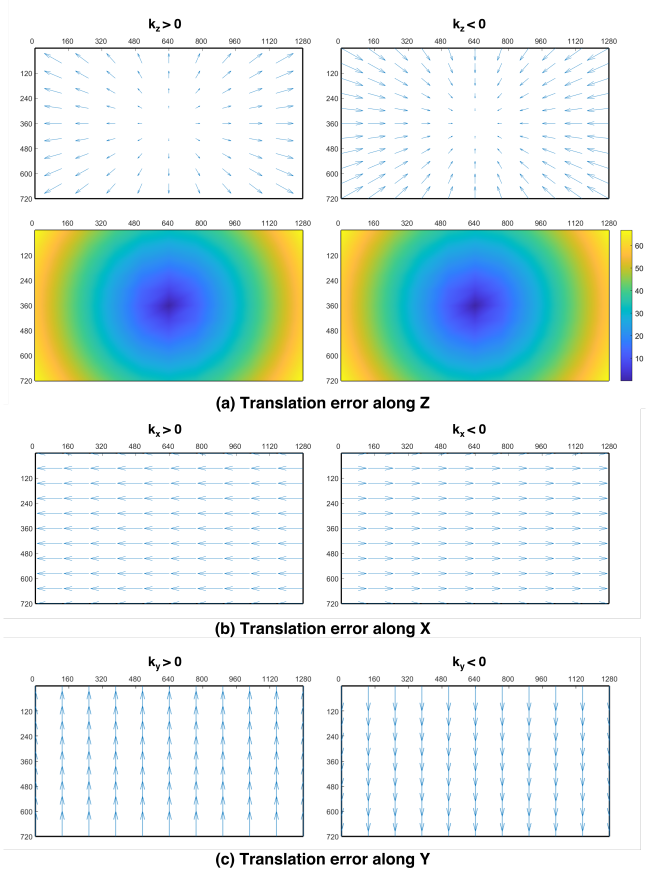
$$\begin{cases} U_{TZ}' = \dfrac{1}{1 - K_z} U - \dfrac{K_z}{1 - K_z} u_0 \\ V_{TZ}' = \dfrac{1}{1 - K_z} V - \dfrac{K_z}{1 - K_z} v_0 \end{cases} \tag{8}$$

$$\begin{cases} \Delta U_{TZ} = \dfrac{K_z}{1 - K_z} (U - u_0) \\ \Delta V_{TZ} = \dfrac{K_z}{1 - K_z} (V - v_0) \end{cases} \tag{9}$$

The simulated trend of the virtual content on the mixed image is depicted in Figure 8(a), utilizing Equation 8 with predefined parameters of $K_z = 0.1$ and $K_z = -0.1$. From the simulation result, translation errors along the Z-axis in the ideal virtual camera coordinates result in a zooming ef-

Table 2. 6 individual transformation errors.

Axis	Translation Error	Rotation Error
X	$\mathbf{T}_{TX} = \begin{bmatrix} 1 & 0 & 0 & e_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\mathbf{T}_{RX} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) & 0 \\ 0 & sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Y	$\mathbf{T}_{TY} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & e_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\mathbf{T}_{RY} = \begin{bmatrix} cos(\beta) & 0 & sin(\beta)) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\beta)) & 0 & cos(\beta)) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Z	$\mathbf{T}_{TZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & e_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\mathbf{T}_{RZ} = \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 & 0 \\ sin(\gamma) & cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$



(a) Translation error along Z

(b) Translation error along X

(c) Translation error along Y

Fig. 8. (a) Misregistration error caused by Z-axis translation $K_z = [0.1, 0.1, \cdots, 0.1]$ (left) and $K_z = [-0.1, -0.1, \cdots, -0.1]$ (right). Upper row: virtual content pixel movement direction. Lower row: simulated magnitude of movement. (b) Error resulting from X-axis translation with parameters $K_x = [0.01, 0.01, \cdots, 0.01]$ (left) and $K_x = [-0.01, -0.01, \cdots, -0.01]$ (right). Uniform movement magnitude for all pixels. (c) Error resulting from Y-axis translation with parameters $K_y = [0.01, 0.01, \cdots, 0.01]$ (left) and $K_y = [-0.01, -0.01, \cdots, -0.01]$ (right). Uniform movement magnitude for all pixels.

fect. When the real virtual camera is positioned in front of the ideal one, the virtual content appears magnified. Conversely, when the real virtual camera is placed behind the ideal one, the virtual content appears reduced in size. This zoom effect is centered around the image's focal point. The magnitude of misregistration is quantified by Eq.9 and visualized in the lower portion of Figure8(a) with given $K_z = [\pm 0.1, \pm 0.1, \cdots, \pm 0.1]$. As observed in the simulated movement shown in Figure 8(a), the virtual content's pixels cover a greater distance as they move away from the image's center. This phenomenon, as described by Eq. 9, demonstrates a positive correlation between the magnitude of misregistration and both the variable K_z and the distance of the pixels from the center of the image.

The same methodology is applied to analyze the translation errors caused by $\mathbf{T}_{TX}$ and $\mathbf{T}_{TY}$. We introduce K_x and K_y as the ratios of the signed translation magnitude to the depth projection magnitude along the Z-axis, as expressed in Eq.10. The resulting misregistered pixel positions are determined using Eq.11 and Eq. 13. Simulated results with fixed values of K_x and K_y are depicted in Figure 8(b) and (c) respectively. Eq. 12 and Eq. 14 ensure consistent movement magnitudes for all pixels.

$$K_x = \frac{e_x}{Z_c},\ K_y = \frac{e_y}{Z_c} \tag{10}$$

$$\begin{cases} U'_{TX} = U - f_x K_x \\ V'_{TX} = V \end{cases} \tag{11}$$

$$\begin{cases} \Delta U_{TX} = -f_x K_x \\ \Delta V_{TX} = 0 \end{cases} \tag{12}$$

$$\begin{cases} U'_{TY} = U \\ V'_{TY} = V - f_y K_y \end{cases} \tag{13}$$

$$\begin{cases} \Delta U_{TY} = 0 \\ \Delta V_{TY} = -f_y K_y \end{cases} \tag{14}$$

Based on the findings, it is evident that the magnitude of pixel misregistration is predominantly influenced by the ratio of the translation distance of the camera coordinate system to the depth of the observed points. Notably, when the observed points are positioned at infinity, any camera offset translation becomes inconsequential. Furthermore, the analysis reveals that pixels located at different positions on the image exhibit varying levels of misregistration in response to the translation along the Z-axis, with pixels farther away from the image center displaying greater deviations. Conversely, the translation of camera coordinates along the X and Y axes produces a consistent effect on pixels across different positions. In this case, an important assumption is that all elements of K_z and K_y are equal, implying that all observed points reside on the same plane perpendicular to the camera's Z-axis. The magnitude of the misregistration in the 2D image is inversely proportional to the projection of the distance from observed points along the camera's Z-axis. However, when the observed points are randomly distributed, the actual magnitudes of the misregistration vary.

4.2 Rotation errors along the Z-X-Y axis

In this particular scenario, we make the assumption that the camera undergoes independent rotations along the Z, X, and Y axes in the ideal camera coordinate system. These rotations are denoted as $\mathbf{T}_{RZ}$, $\mathbf{T}_{RX}$, and $\mathbf{T}_{RY}$ respectively, and their detailed descriptions can be found in Table 2. To begin, we examine the rotation error specifically along the Z-axis. By incorporating $\mathbf{T}_{RZ}$ into Eq.4, all the viewed points in the camera coordinate system undergo transformations, resulting in different positions. Utilizing Eq. 1, 2 and 5, we establish a set of relationships between the actual pixel position (U'_{RZ}, V'_{RZ}) and the ideal pixel position (U,V) on the image plane, considering the viewed point's position in the ideal camera coordinates. These relationships are expressed in Eq.15. It is worth noting that the assumption of zero skew confidence is made, implying the absence of skewness.

$$\begin{cases} Z_c u = f_x X_c + u_0 Z_c \\ Z_c U'_{RZ} = f_x[cos(\gamma)X_c + sin(\gamma)Y_c] + u_0 Z_c \\ Z_c v = f_y Y_c + v_0 Z_c \\ Z_c V'_{RZ} = f_y[cos(\gamma)Y_c - sin(\gamma)X_c] + v_0 Z_c \end{cases} \tag{15}$$

To streamline and reduce the complexity of equations, the final expression for the misregistered pixel position (U'_{RZ}, V'_{RZ}) resulting from rotation error along the Z-axis is given by Eq.16. Considering the common case, where $f_x = f_y$, this equation is further optimized as Eq.17.Then the distribution of misregistration is visualized with the given γ in the upper part of Figure 9(a).

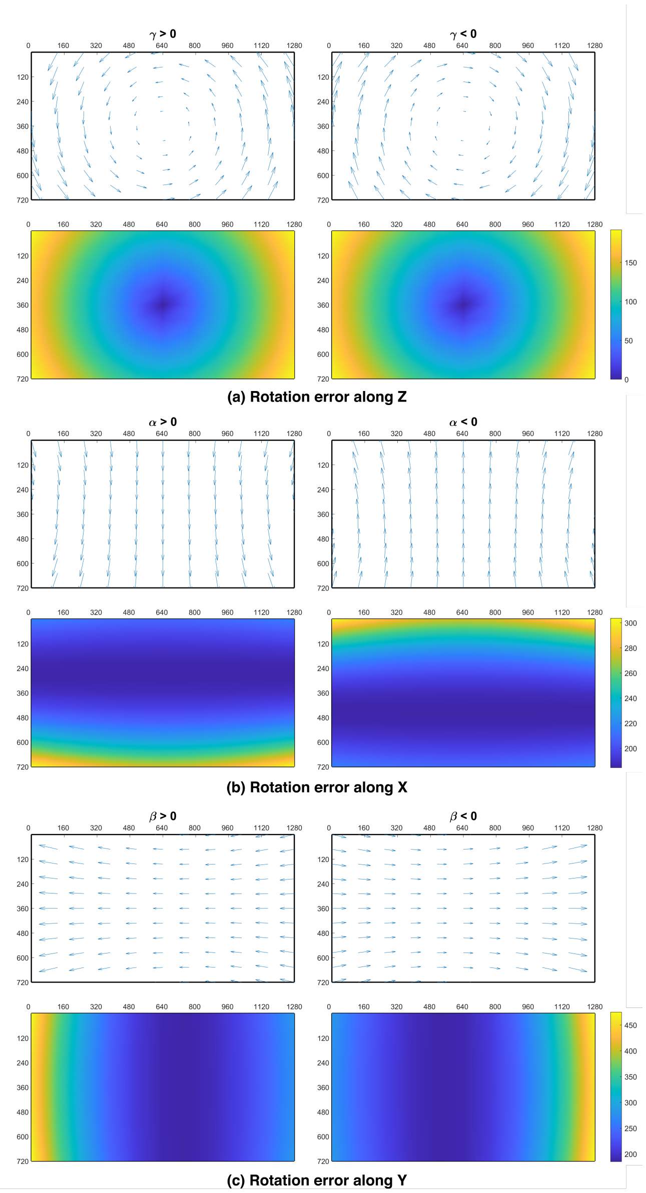
$$\begin{cases} U'_{RZ} = cos(\gamma)(U - u_0) + sin(\gamma)\frac{f_x}{f_y}(V - v_0) + u_0 \\ V'_{RZ} = cos(\gamma)(V - v_0) - sin(\gamma)\frac{f_y}{f_x}(U - u_0) + v_0 \end{cases} \tag{16}$$

$$\begin{cases} U'_{RZ} = cos(\gamma)(U - u_0) + sin(\gamma)(V - v_0) + u_0 \\ V'_{RZ} = cos(\gamma)(V - v_0) - sin(\gamma)(U - u_0) + v_0 \end{cases} \tag{17}$$

$$\begin{aligned} \Delta D^2_{RZ} &= (U'_{RZ} - U)^2 + (V'_{RZ} - V)^2 \\ &= (2 - 2cos(\gamma))[(U - u_0)^2 + (V - v_0)^2] \end{aligned} \tag{18}$$

The magnitude of misregistration, computed using Eq. 18, is visualized as a heatmap in the lower section of Figure 9 (a). Analysis of the simulation results reveals that when a camera error is introduced, specifically rotation around the Z-axis, all pixels within the 2D image rotate in the opposite direction relative to the image center. The magnitude of pixel displacement is directly proportional to the distance from the image center, while pixels located precisely at the image center remain unaffected and stationary.

By applying the same mathematical approach, the respective rotation matrices $\mathbf{T}_{RX}$ for rotation along the X-axis and $\mathbf{T}_{RY}$ for rotation along the Y-axis are utilized in Eq.4 and Eq.5. This enables the calculation of new coordinates for misregistration, as expressed by Eq.19 and Eq.20. To further investigate the impact of misregistration, analysis is conducted on the directional distribution and magnitude distribution. By considering different given rotation angles, variations in these distributions can be observed, as visually illustrated in Figure 9(b) and 9(c) respectively.



(a) Rotation error along Z

(b) Rotation error along X

(c) Rotation error along Y

Fig. 9. The distribution of misregistration directions and the corresponding heat map illustrating the magnitude of misregistration caused by individual factors are presented. (a) Misregistration error resulting from rotation along the Z-axis with angles of $\gamma = \pi/12$ (left) and $\gamma = -\pi/12$ (right). (b) Misregistration error arising from rotation along the X-axis with angles of $\alpha = \pi/12$ (left) and $\alpha = -\pi/12$ (right). (c) Misregistration error due to rotation along the Y-axis with angles of $\beta = \pi/12$ (left) and $\beta = -\pi/12$ (right).

$$\begin{cases} U'_{RX} = \dfrac{f_y(U - u_0)}{cos(\alpha)f_y - sin(\alpha)(V - v_0)} + u_0 \\ V'_{RX} = \dfrac{cos(\alpha)f_y(V - v_0) + sin(\alpha)f_y^2}{cos(\alpha)f_y - sin(\alpha)(V - v_0)} + v_0 \end{cases} \tag{19}$$

$$\begin{cases} U'_{RY} = \dfrac{cos(\beta) f_x (U-u_0) - sin(\beta) f_x^2}{cos(\beta) f_x + sin(\beta)(U-u_0)} + u_0 \\ V'_{RY} = \dfrac{f_x (V-v_0)}{cos(\beta) f_x + sin(\beta)(U-u_0)} + v_0 \end{cases} \tag{20}$$

The simulations were conducted to investigate the effects of positive and negative 15° rotation errors around the X-axis and Y-axis, respectively. The results showed that the majority of pixels experienced a consistent directional shift. However, it was observed that these distributions were non-uniform. Specifically, rotation errors around the X-axis resulted in horizontal variations, while rotation errors around the Y-axis caused vertical variations.

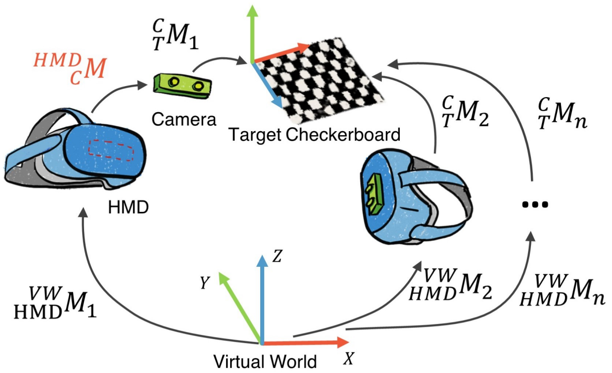


Fig. 10. Transformations involved in the calibration process, where the camera is attached to the HMD to capture images of a stationary checkerboard from various viewpoints.

5 Calibration of HMD-to-Camera Registration

The primary objective of this work is to achieve global misregistration reduction in video-based augmented reality systems without the use of detectable patterns. Accurate visualization of virtual content relies on the relative transformation between the camera and virtual objects, which are both registered in the same reference coordinate system, as explained in Section 3. In essence, the position of virtual content within the reference coordinate system is defined by the user based on the visualization outcome, forming an interconnected process. Figure 5 illustrates the registration framework, while Figure 6 presents a simplified transformation. It is crucial to ensure precise transformations at each step to achieve the desired level of accuracy in the final registration process.

Among the four error categories mentioned earlier, the errors associated with World-to-Object and Camera-to-Image can be effectively eliminated. This is because the virtual objects maintain static positions in the virtual world which is defined through user perception, and the camera is calibrated with accurate intrinsic parameters through Zhang's calibration method [54]. However, errors arise in the World-to-HMD transformation due to the employed tracking techniques, which are not the primary focus of this study. For our investigation, we utilize two commercial HMD devices, namely the Oculus Rift S and the first-generation Oculus Quest, both of which exhibit negligible inaccuracies that fall within acceptable tolerance levels. The positional accuracy of the Oculus Rift S HMD, as reported in a previous study, averages at 1.66 *mm* [55], while the first-generation Oculus Quest demonstrates an average positional accuracy of 6.86 *mm* [56]. Consequently, based on the derived error impacts from the mathematical model, our main objective is to calibrate the HMD-to-Camera transformation, aiming to minimize global misregistration while considering the existing errors associated with the World-to-HMD transformation.

5.1 Initial estimation of HMD-to-Camera

External sensors are utilized to track the camera's movement and obtain extrinsic parameters. However, direct tracking of the camera using the Oculus IR tracking system is not feasible as it is primarily designed to track the HMD and controllers. Therefore, the camera's movement is indirectly inferred through the movement of the HMD. To establish the relationship between the HMD and the camera, an offset matrix $^{C}_{HMD}\mathbf{M}$ is employed as introduced in Table 1. The default calibration for the HMD-to-Camera transformation is documented by StereoLabs and represented by Eq. 21, where the unit of translation is in meters.

$$^{HMD}_{C}\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & -0.0315 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.115 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{21}$$

The HMD-to-Camera matrix manufacturer provided represents the spatial offset between the ZED camera and the HMD tracking frame reference. The ZED camera's reference frame is defined by its left sensor, while the HMD's reference frame is typically centered around the head. The translation vector's unit is in meters, and the value of 0.0315 *m* corresponds to half of the baseline distance of 63 *mm*. Consequently, the assumption is made that the camera undergoes a translation of 0.115 *m* along the positive Z-axis. It is important to note that this transformation is based on an average offset and may not accurately reflect real conditions. Because, the size of different HMDs and the angle at which they are attached the camera can introduce variations in the HMD-to-Camera matrix, leading to registration errors in display.

To determine the HMD-to-Camera transformation matrix, the calibration process involves setting up a controlled environment with a stationary checkerboard pattern, as depicted in Figure 10. The coordinate system used for this transformation is based on the virtual world, where the HMD is tracked using specific sensors. Within the HMD's reference frame, the HMD-to-Camera matrix represents the transformation from the HMD to the camera. In the camera frame, the transformation matrix $^{C}_{T}\mathbf{M}$ describes the transformation from the camera to the origin of the target checkerboard located at the top left corner. This transformation relates the target checkerboard origin (T) to the virtual coordinate system (VW) and involves both the camera and the HMD, as expressed in Eq. 22.

$$^{VW}_{T}\mathbf{M} = {}^{VW}_{HMD}\mathbf{M} \cdot {}^{HMD}_{C}\mathbf{M} \cdot {}^{C}_{T}\mathbf{M} \tag{22}$$

By utilizing the aforementioned equation, the transformation matrices $^{VW}_{T}\mathbf{M}$ and $^{HMD}_{C}\mathbf{M}$ are currently unknown. The $^{VW}_{HMD}\mathbf{M}$ matrix can be obtained through the tracking system, while $^{C}_{T}\mathbf{M}$ can be estimated using computer vision techniques applied to the captured images. To simplify and eliminate variables, the target checkerboard is kept stationary while moving the HMD and the attached camera together to capture images of the checkerboard from different viewpoints. This process allows for the removal of $^{VW}_{T}\mathbf{M}$ using Eq. 23.

$$^{VW}_{HMD}\mathbf{M}_i \cdot {}^{HMD}_{C}\mathbf{M}_i \cdot {}^{C}_{T}\mathbf{M}_i = {}^{VW}_{HMD}\mathbf{M}_j \cdot {}^{HMD}_{C}\mathbf{M}_j \cdot {}^{C}_{T}\mathbf{M}_j \tag{23}$$

In the above equation, the $^{HMD}_{C}\mathbf{M}$ matrix is the only unknown, with 6 variables. In order to perform the computation, a minimum of 3 distinct captured images are required. However, to improve accuracy, a total of 23 sets of effective images capturing the checkerboard from various viewpoints were collected for optimization. Any two of these sets can be used to form an equation using Eq.23. As a result, we obtained 253 equations (C^2_{23}) for further optimization. By rearranging the terms, Eq.23 can be expressed as Eq. 24.

$$^{vw}_{HMD}\mathbf{M}_j^{-1} \cdot {}^{vw}_{HMD}\mathbf{M}_i \cdot {}^{HMD}_{C}\mathbf{M}_i = {}^{HMD}_{C}\mathbf{M}_j \cdot {}^{C}_{T}\mathbf{M}_j \cdot {}^{C}_{T}\mathbf{M}_i^{-1} \tag{24}$$

This equation takes the form of $\mathbf{AX} = \mathbf{XB}$. To solve these equations, the Tsai-Lens method [57] can be applied. By utilizing the least squares method for nonlinear optimization, highly accurate and dependable results for $^{HMD}_{C}\mathbf{M}$ can be obtained, as depicted in Eq. 25, with translations measured in meters.

$$^{HMD}_{C}\mathbf{M} = \begin{bmatrix} 0.999754 & -0.006011 & 0.021326 & -0.039472 \\ 0.006008 & 0.999982 & 0.000198 & 0.001074 \\ -0.021327 & -0.000070 & 0.999773 & 0.130205 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{25}$$

It is important to emphasize that the calibration process described does not rely on user perception or the use of complex 3D calibration objects. Instead, any detectable 2D barcode can serve as a suitable calibration target. This calibration step is designed to address the primary errors present in general or DIY (Do-It-Yourself) VST AR systems used in both industry and research, and it can also be extended to calibrate the Device-to-Camera alignment, which directly impacts the display outcomes.

However, it is crucial to acknowledge that this calibration step cannot entirely eliminate all system errors. One key assumption made during this calibration process is the absence of tracking errors in the HMD, but it can be challenging to completely eliminate such errors due to variations in tracking systems and distances involved. Furthermore, in practical applications, virtual objects lack physical reference points. The determination of the World-to-Object transformation relies on subjective judgments made by users based on visual results. Inaccuracies in the World-to-Object transformations can adversely affect the Robot-to-Object matrix, consequently impacting the execution of robot programming, as detailed in Figure 5. Therefore, additional efforts are required to mitigate the propagation of global transformations and achieve more precise World-to-Object transformations.

5.2 Error Correction of HMD-to-Camera Registration

An additional error correction is performed on the HMD-to-Camera transformation matrix. The six parameters, encompassing translations and orientations, can be adjusted to minimize the initially estimated result obtained from the uncorrected input propagated from the World-to-HMD transformation.

Based on the simulation results presented in Section 4, it is observed that certain individual adjustments may yield similar corrections. For instance, translating along the X-axis and rotating along the Y-axis can produce comparable effects. Consequently, the selection of adjusted parameters depends on their uncertainty and sensitivity. Bajura and Neumann [58] discussed the uncertainty and sensitivity of image-space errors, highlighting that the camera's position has a greater impact on the projection of an image when the viewed point is relatively close to the camera, whereas the camera's orientation plays a more significant role when the viewed point is further away. This analysis aligns with the findings reported in Section 4.

1. To accommodate a zoom-in or zoom-out effect on the virtual content, it becomes necessary to adjust the translation along the Z-axis. Increasing the Z-axis translation will result in a zoom-out effect while decreasing it will yield a zoom-in effect.
2. In the case of rotation of the virtual content along the normal passing through the image center, adjustment of the rotation along the Z-axis becomes essential to ensure proper alignment of the axis orientation.
3. When there is a translation of the virtual content and the observed virtual points are relatively close to the camera, it is recommended to modify the translation along both the X and Y axes of the camera.
4. In scenarios where the virtual content undergoes translation and the observed virtual points are at a considerable distance from the camera, it is advisable to modify the rotation along both the X and Y axes of the camera.

6 Experimental Evaluation and Result

This section presents a comprehensive evaluation of the mathematical model, considering both qualitative and quantitative aspects. The qualitative evaluation involves the introduction of pre-set individual errors to the calibrated HMD-to-Camera transformation. By examining the resulting display output, we can verify the consistency between the mathematical model and the simulation results. The primary ob-

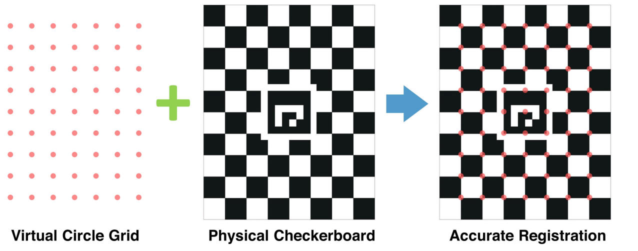


Fig. 11. The registration target in the AR scene, where is virtual circle grid is expected to align with the intersections on the physical checkerboard.

jective of this qualitative evaluation is to confirm the observed trends and distributions observed in the simulations. In the quantitative evaluation, we aim to assess the effectiveness of the calibration approach by comparing it with the utilization of the manufacturer's default settings. This quantitative assessment serves as a measure of the calibration's performance and its potential benefits over using default configurations.

6.1 Qualitative Evaluation

In the qualitative experiment, a virtual point grid is registered onto a physical checkerboard, as depicted in Figure 11. The target consists of a 9×7 pattern of circles with a pitch of 20 *mm*, where each circle has a diameter of 5 *mm*. The objective is to align the virtual circles with the intersections of the grid on the physical checkerboard and examine the alignment from various viewpoints. Following the calibration process, the virtual points accurately align with their respective positions on the checkerboard, as observed during the viewpoint changes depicted in Figure 12(a-c).

Subsequently, intentional errors are introduced to the HMD-to-Camera transformation. When solely applying a translation along the Z-axis in the camera coordinates, the virtual grid undergoes a positive translation, resulting in a zoomed-in appearance as depicted in Figure 12(d). Conversely, a negative translation causes the grid to be zoomed out, indicating that the virtual camera is positioned behind the real camera. Additionally, an offset in the virtual grid is observed when adjusting the translation along the X and Y axes, as shown in Figure 12(d). Furthermore, individual rotation errors are evaluated. When a rotation error occurs along the Z-axis, the virtual content rotates around the image center, as illustrated in Figure 12(g). Similarly, rotations along the X and Y axes lead to a displacement of the virtual grid, as shown in Figure 12(h) and (i).

The qualitative results obtained from the conducted experiments validate the trends and distribution observed in the simulations presented in Section 4. By employing a plane registration target, it becomes evident that the individual factors within the HMD-to-Camera transformation contribute to distinct misregistration errors. Furthermore, the observed trends and distribution of these misregistrations exhibit clear distinctions between the separate factors. This not only confirms the effectiveness of the mathematical model but also establishes a solid groundwork for future research aimed at analyzing more intricate error models.

6.2 Quantitative Evaluation

In this section, we employ a quantitative method to measure the improvement in calibration compared to using the default settings (before calibration). The accuracy of visualization is assessed through frames defined in the AR system, with the ground truth of the frame pose measured through a calibration tool. To define a frame in the AR system, we introduce and implement a modified 3-point method, commonly used for defining user frames in robotics.

The traditional 3-point method in robotics involves using three reference points: the origin, the X direction, and the Y direction, to redefine a coordinate system as a user frame. However, these three points contribute differently to the accuracy of the frame definition. To address this, we apply a modified version of the method to mitigate the influence of individual points.

In the modified 3-point method, three points need to be defined in 3D space: one along the X direction, one along the Y direction, and one along the Z direction. And they share the same fixed offset from the origin. It is known that a minimum of three non-collinear points can define the pose of a rigid body in three-dimensional space. These reference points, referred to as landmarks, are equidistant from the origin but located on different axes. Users are guided to define these landmarks using a 3D-printed calibration tool (shown in Figure 14). The task is simple, where users align virtual balls with the calibration tool based on their visual perception. By determining the position of each landmark, the pose of the frame can be calculated. If the initial registration of the frame is accurate, regardless of camera viewpoint changes, the virtual balls should consistently align with the physical balls. To facilitate repeated experiments, we consider two conditions for camera movement, as illustrated in Figure 13:

1. The camera remains fixed at a specific position while rotating to capture the target in different regions of the image, including the top, bottom, left, and right.
2. The camera changes its position while ensuring that the target remains centered in the image.

To facilitate comparison, we gather two sets of AR images: one taken before calibration and the other after calibration. In each experimental condition, four images are captured using the left-eye camera. To ensure consistent observation positions, the HMD with the camera is securely mounted on a manipulator, specifically the Aubo i5, as depicted in Figure 14. The robot manipulator possesses a repeatability of ±0.05 *mm* and is programmed to maintain consistent observation positions throughout the experiment.

The physical calibration tool used in this study consists of 3D printed axes with three spheres positioned at the top of each axis (refer to Figure 14). Each sphere has a diameter of 10 *mm*. The calibration tool is positioned at a distance of 1 *m* from the camera. The alignment process involves aligning three virtual balls of the same size with the physical spheres.

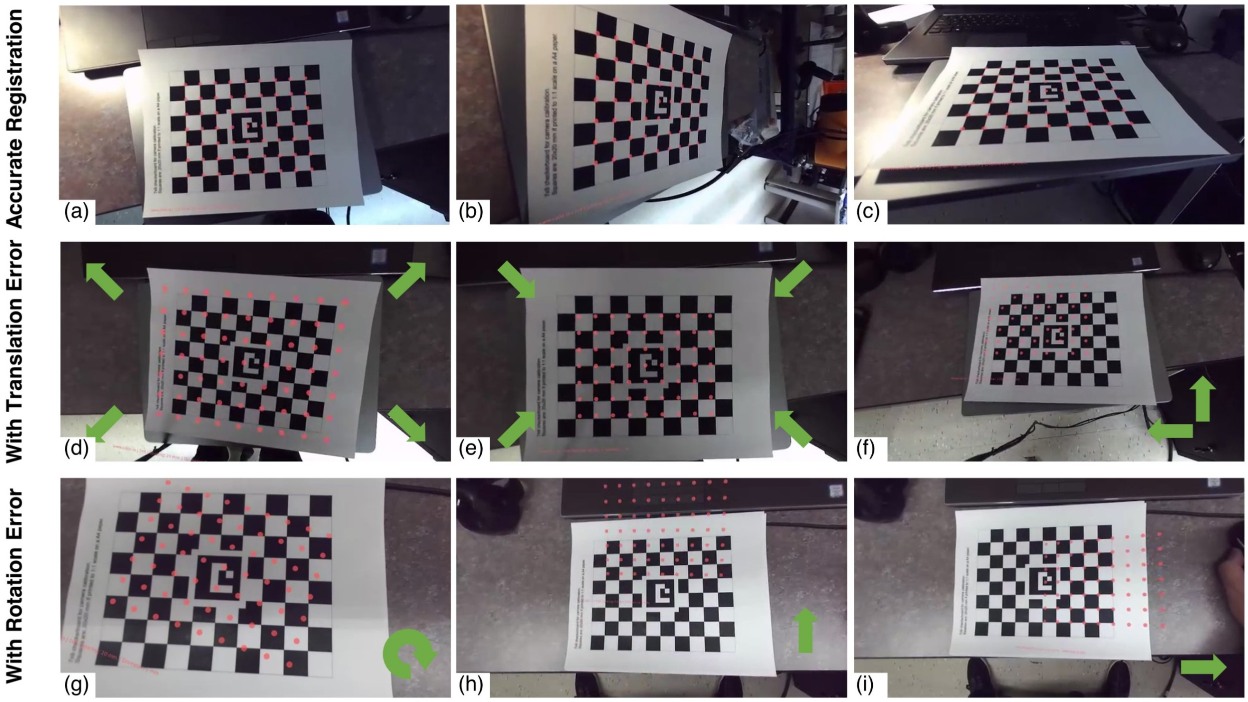


Fig. 12. The first row (a-c) shows the registration after calibration from different points of view. Translation errors on HMD-to-Camera are introduced in the second row (d-f). Zoom-in and zoom-out effects are caused by translation along the Z-axis in (d) and (e). The offset in (f) is due to a combination of translation along the X-axis and Y-axis. Rotation errors are appended in the last row (g-i). (g) indicates the consequence of additional rotation error along the Z-axis. Rotation along the X-axis and Y-axis also results in offset misregistration, respectively, in (h) and (i).

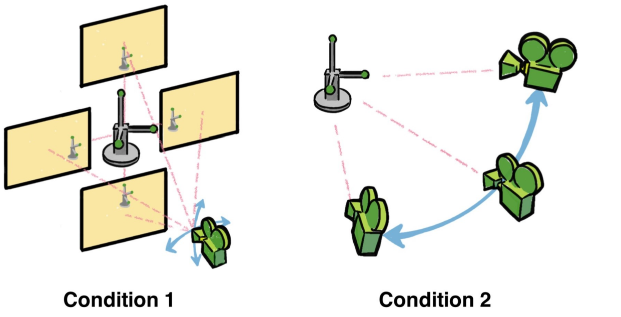


Fig. 13. The measurement data are collected under two conditions. In condition 1, the camera stays in the same position and captures images by rotating the view direction. In condition 2, the camera changes position and captures images from different points of view.

Initially, three sets of balls are registered from a specific starting point of view using the aforementioned modified 3-point method. Subsequently, eight images are captured from two additional points of view.

The registration error is quantified by calculating the pixel offset of the ball center. A comparative analysis is performed using a two-tailed t-test to assess the misregistration in 12 sets under each condition. The results are presented graphically in Figure 15, demonstrating a significant reduction in misregistration after the calibration process.

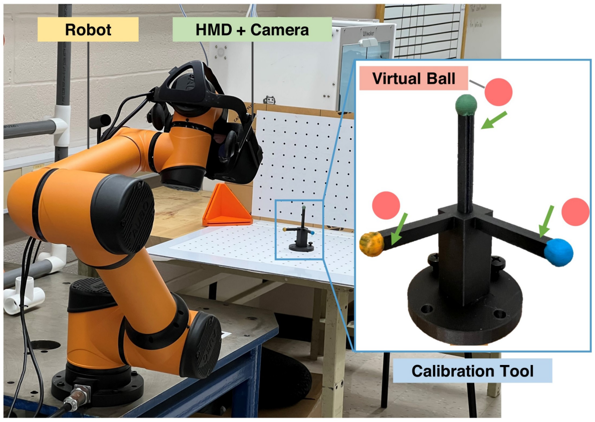


Fig. 14. The experimental setup of the quantitative evaluation. The HMD with the camera is mounted on the manipulator, which is programmed to change preset points of view. Three virtual balls are measured to align with the physical calibration tool.

7 Conclusions

This study focuses on addressing the misregistration issues that occur in Video See-Through Augmented Reality (AR) systems, which limit the potential for shared workspaces between humans and machines in shop-floor applications. The sources of error are analyzed and presented within a comprehensive system framework. A mathematical

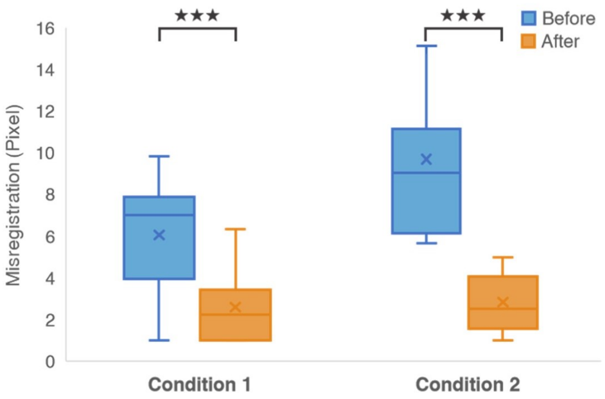


Fig. 15. Significant differences were observed between the calibration results under two conditions, with p-values of 0.002852 and 0.000015 obtained from a two-tailed t-test for each condition.

model is proposed to characterize the impact and sensitivity of the error specifically in the HMD-to-Camera transformation. The model examines the six individual factors comprising the transformation matrix, including three translations and three rotations, and simulates the resulting misregistration effects. To mitigate the HMD-to-Camera error, a closed-loop calibration method is introduced and applied in a prototyping system. The calibration process involves the initial estimation and fine adjustment of the HMD-to-Camera transformation. Both qualitative and quantitative evaluations are conducted to validate the mathematical model and the calibration approach. The results demonstrate successful global registration between virtual objects and the physical environment.

The limitation of this research is that how the registration accuracy reflects the depth of the view target is unexplored. The points in the shared workspace won't share a consistent registration error. The distribution of misregistration has yet to be studied. Another future research question is how to achieve accurate registration in a specific application. For example, in the AR-based robot programming task, the virtual world is required to align with the real world via the robot base to share the same robot workspace. The aforementioned potential problems will be prioritized in the future.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Award No. DGE-2125362 and 2222853. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Azuma, R. T., 1997. "A survey of augmented reality". *Presence: teleoperators & virtual environments,* **6**(4), pp. 355–385.

[2] Billinghurst, M., Clark, A., Lee, G., et al., 2015. "A survey of augmented reality". *Foundations and Trends® in Human–Computer Interaction,* **8**(2-3), pp. 73–272.

[3] Rehman, U., and Cao, S., 2016. "Augmented-reality-based indoor navigation: A comparative analysis of handheld devices versus google glass". *IEEE Transactions on Human-Machine Systems,* **47**(1), pp. 140–151.

[4] Birlo, M., Edwards, P. E., Clarkson, M., and Stoyanov, D., 2022. "Utility of optical see-through head mounted displays in augmented reality-assisted surgery: A systematic review". *Medical Image Analysis,* **77**, p. 102361.

[5] Fang, W., Chen, L., Zhang, T., Chen, C., Teng, Z., and Wang, L., 2023. "Head-mounted display augmented reality in manufacturing: A systematic review". *Robotics and Computer-Integrated Manufacturing,* **83**, p. 102567.

[6] Condino, S., Carbone, M., Piazza, R., Ferrari, M., and Ferrari, V., 2019. "Perceptual limits of optical see-through visors for augmented reality guidance of manual tasks". *IEEE Transactions on Biomedical Engineering,* **67**(2), pp. 411–419.

[7] Funk, M., Kosch, T., and Schmidt, A., 2016. "Interactive worker assistance: comparing the effects of in-situ projection, head-mounted displays, tablet, and paper instructions". In Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing, pp. 934–939.

[8] Westerfield, G., Mitrovic, A., and Billinghurst, M., 2015. "Intelligent augmented reality training for motherboard assembly". *International Journal of Artificial Intelligence in Education,* **25**, pp. 157–172.

[9] Siew, C. Y., Ong, S.-K., and Nee, A. Y., 2019. "A practical augmented reality-assisted maintenance system framework for adaptive user support". *Robotics and Computer-Integrated Manufacturing,* **59**, pp. 115–129.

[10] Nee, A. Y., Ong, S., Chryssolouris, G., and Mourtzis, D., 2012. "Augmented reality applications in design and manufacturing". *CIRP annals,* **61**(2), pp. 657–679.

[11] Evans, G., Miller, J., Pena, M. I., MacAllister, A., and Winer, E., 2017. "Evaluating the microsoft hololens through an augmented reality assembly application". In Degraded environments: sensing, processing, and display 2017, Vol. 10197, SPIE, pp. 282–297.

[12] Quintero, C. P., Li, S., Pan, M. K., Chan, W. P., Van der Loos, H. M., and Croft, E., 2018. "Robot programming through augmented trajectories in augmented reality". In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 1838–1844.

[13] Gallala, A., Hichri, B., and Plapper, P., 2019. "Survey: The evolution of the usage of augmented reality in industry 4.0". In IOP Conference Series: Materials Science and Engineering, Vol. 521, IOP Publishing, p. 012017.

[14] Makhataeva, Z., and Varol, H. A., 2020. "Augmented reality for robotics: A review". *Robotics,* **9**(2), p. 21.

[15] Ong, S.-K., Yew, A., Thanigaivel, N. K., and Nee, A. Y., 2020. "Augmented reality-assisted robot programming system for industrial applications". *Robotics and Computer-Integrated Manufacturing,* **61**, p. 101820.

[16] Van Krevelen, D., and Poelman, R., 2010. "A survey of augmented reality technologies, applications and limitations". *International journal of virtual reality,* **9**(2), pp. 1–20.

[17] Ballestin, G., Chessa, M., and Solari, F., 2021. "A registration framework for the comparison of video and optical see-through devices in interactive augmented reality". *IEEE Access,* **9**, pp. 64828–64843.

[18] Li, H., and Hartley, R., 2007. "The 3d-3d registration problem revisited". In 2007 IEEE 11th international conference on computer vision, IEEE, pp. 1–8.

[19] Rolland, J. P., and Fuchs, H., 2000. "Optical versus video see-through head-mounted displays in medical visualization". *Presence,* **9**(3), pp. 287–309.

[20] Liu, R., Peng, C., Zhang, Y., Husarek, H., and Yu, Q., 2021. "A survey of immersive technologies and applications for industrial product development". *Computers & Graphics,* **100**, pp. 137–151.

[21] Zhou, F., Duh, H. B.-L., and Billinghurst, M., 2008. "Trends in augmented reality tracking, interaction and display: A review of ten years of ismar". In 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, IEEE, pp. 193–202.

[22] Yang, W., Xiao, Q., and Zhang, Y., 2021. "An augmented-reality based human-robot interface for robotics programming in the complex environment". In International Manufacturing Science and Engineering Conference, Vol. 85079, American Society of Mechanical Engineers, p. V002T07A003.

[23] Yang, W., Xiao, Q., and Zhang, Y., 2023. "HAR^{2}bot: a human-centered augmented reality robot programming method with the awareness of cognitive load". *Journal of Intelligent Manufacturing*, pp. 1–19.

[24] Yang, W., and Zhang, Y., 2022. "Visualization error analysis for augmented reality stereo video see-through head-mounted displays in industry 4.0 applications". In International Manufacturing Science and Engineering Conference, Vol. 85819, American Society of Mechanical Engineers, p. V002T06A016.

[25] Samini, A., Palmerius, K. L., and Ljung, P., 2021. "A review of current, complete augmented reality solutions". In 2021 International Conference on Cyberworlds (CW), pp. 49–56.

[26] Howard, I., and Rogers, B., 2002. "Seeing in depth volume 2 depth perception (toronto: I porteous)".

[27] Diaz, C., Walker, M., Szafir, D. A., and Szafir, D., 2017. "Designing for depth perceptions in augmented reality". In 2017 IEEE international symposium on mixed and augmented reality (ISMAR), IEEE, pp. 111–122.

[28] Swan, J. E., Jones, A., Kolstad, E., Livingston, M. A., and Smallman, H. S., 2007. "Egocentric depth judgments in optical, see-through augmented reality". *IEEE transactions on visualization and computer graphics,* **13**(3), pp. 429–442.

[29] Ballestin, G., Solari, F., and Chessa, M., 2018. "Perception and action in peripersonal space: A comparison between video and optical see-through augmented reality devices". In 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), IEEE, pp. 184–189.

[30] Calabrò, E. M., Cutolo, F., Carbone, M., and Ferrari, V., 2017. "Wearable augmented reality optical see through displays based on integral imaging". In Wireless Mobile Communication and Healthcare, P. Perego, G. Andreoni, and G. Rizzo, eds., Springer International Publishing, pp. 345–356.

[31] Takagi, A., Yamazaki, S., Saito, Y., and Taniguchi, N., 2000. "Development of a stereo video see-through hmd for ar systems". *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR 2000)*, pp. 68–77.

[32] Cattari, N., Cutolo, F., D'amato, R., Fontana, U., and Ferrari, V., 2019. "Toed-in vs parallel displays in video see-through head-mounted displays for close-up view". *IEEE Access,* **7**, pp. 159698–159711.

[33] Cutolo, F., Fontana, U., and Ferrari, V., 2018. "Perspective preserving solution for quasi-orthoscopic video see-through hmds". *Technologies,* **6**(1), p. 9.

[34] Stereo Labs, 2017. Zed mini. https://www.stereolabs.com/zed-mini/. [Online; accessed 10-Jul-2023].

[35] Samini, A., Palmerius, K. L., and Ljung, P., 2021. "A review of current, complete augmented reality solutions". In 2021 International Conference on Cyberworlds (CW), IEEE, pp. 49–56.

[36] Li, K., Schmidt, S., Bacher, R., Leemans, W., and Steinicke, F., 2022. "Mixed reality tunneling effects for stereoscopic untethered video-see-through head-mounted displays". In 2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), IEEE, pp. 44–53.

[37] Maruhn, P., Dietrich, A., Prasch, L., and Schneider, S., 2020. "Analyzing pedestrian behavior in augmented reality—proof of concept". In 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), IEEE, pp. 313–321.

[38] Li, K., Choudhuri, A., Schmidt, S., Lang, T., Bacher, R., Hartl, I., Leemans, W., and Steinicke, F., 2022. "Stereoscopic video see-through head-mounted displays for laser safety: An empirical evaluation at advanced optics laboratories". In 2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), IEEE, pp. 112–120.

[39] Pfeil, K., Masnadi, S., Belga, J., Sera-Josef, J.-V. T., and LaViola, J., 2021. "Distance perception with a video see-through head-mounted display". In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–9.

[40] Tuceryan, M., Greer, D. S., Whitaker, R. T., Breen, D. E., Crampton, C., Rose, E., and Ahlers, K. H., 1995. "Calibration requirements and procedures for a

monitor-based augmented reality system". *IEEE Transactions on Visualization and Computer Graphics,* **1**(3), pp. 255–273.

[41] Grubert, J., Itoh, Y., Moser, K., and Swan, J. E., 2017. "A survey of calibration methods for optical see-through head-mounted displays". *IEEE transactions on visualization and computer graphics,* **24**(9), pp. 2649–2662.

[42] Moser, K. R., Arefin, M. S., and Swan, J. E., 2018. "Impact of alignment point distance and posture on spaam calibration of optical see-through head-mounted displays". In 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), IEEE, pp. 21–30.

[43] Moser, K., Itoh, Y., Oshima, K., Swan, J. E., Klinker, G., and Sandor, C., 2015. "Subjective evaluation of a semi-automatic optical see-through head-mounted display calibration technique". *IEEE transactions on visualization and computer graphics,* **21**(4), pp. 491–500.

[44] Besl, P. J., and McKay, N. D., 1992. "Method for registration of 3-d shapes". In Sensor fusion IV: control paradigms and data structures, Vol. 1611, Spie, pp. 586–606.

[45] Fitzgibbon, A. W., 2003. "Robust registration of 2d and 3d point sets". *Image and vision computing,* **21**(13-14), pp. 1145–1153.

[46] Rusinkiewicz, S., and Levoy, M., 2001. "Efficient variants of the icp algorithm". In Proceedings third international conference on 3-D digital imaging and modeling, IEEE, pp. 145–152.

[47] Gold, S., Rangarajan, A., Lu, C.-P., Pappu, S., and Mjolsness, E., 1998. "New algorithms for 2d and 3d point matching: pose estimation and correspondence". *Pattern recognition,* **31**(8), pp. 1019–1031.

[48] Granger, S., and Pennec, X., 2002. "Multi-scale em-icp: A fast and robust approach for surface registration". In European conference on computer vision, Springer, pp. 418–432.

[49] Fuhrmann, A., Schmalstieg, D., and Purgathofer, W., 1999. "Fast calibration for augmented reality". In Proceedings of the ACM symposium on Virtual reality software and technology, pp. 166–167.

[50] Kato, H., and Billinghurst, M., 1999. "Marker tracking and hmd calibration for a video-based augmented reality conferencing system". In Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99), IEEE, pp. 85–94.

[51] Hu, X., y Baena, F. R., and Cutolo, F., 2021. "Rotation-constrained optical see-through headset calibration with bare-hand alignment". In 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), IEEE, pp. 256–264.

[52] Faugeras, O., 1993. *Three-dimensional computer vision: a geometric viewpoint.* MIT press.

[53] Hartley, R., and Zisserman, A., 2003. *Multiple view geometry in computer vision.* Cambridge university press.

[54] Zhang, Z., 2000. "A flexible new technique for camera calibration". *IEEE Transactions on pattern analysis and machine intelligence,* **22**(11), pp. 1330–1334.

[55] Jost, T. A., Nelson, B., and Rylander, J., 2021. "Quantitative analysis of the oculus rift s in controlled movement". *Disability and Rehabilitation: Assistive Technology,* **16**(6), pp. 632–636.

[56] Eger Passos, D., and Jung, B., 2020. "Measuring the accuracy of inside-out tracking in xr devices using a high-precision robotic arm". In HCI International 2020-Posters: 22nd International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings, Part I, Springer, pp. 19–26.

[57] Tsai, R. Y., Lenz, R. K., et al., 1989. "A new technique for fully autonomous and efficient 3 d robotics hand/eye calibration". *IEEE Transactions on robotics and automation,* **5**(3), pp. 345–358.

[58] Bajura, M., and Neumann, U., 1995. "Dynamic registration correction in augmented-reality systems". In Proceedings Virtual Reality Annual International Symposium'95, IEEE, pp. 189–196.