Proceedings of the ASME 2022 17th International Manufacturing Science and Engineering Conference
MSEC2022
June 27-July 1, 2022, West Lafayette, Indiana, USA

MSEC2022-85440

VISUALIZATION ERROR ANALYSIS FOR AUGMENTED REALITY STEREO VIDEO SEE-THROUGH HEAD-MOUNTED DISPLAYS IN INDUSTRY 4.0 APPLICATIONS

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

ABSTRACT

Under the fourth industrial revolution (Industry 4.0), Augmented Reality (AR) provides new affordances for a variety of applications, such as AR-based human-robot interaction, virtual assembly assistance, and workforce virtual training. The see-through head-mounted displays (STHMDs), based on either optical see-through or video see-through technologies, are the primary AR device to augment the visual perception of the real environment with computer-generated contents through a hand-free headset. Specifically, the video see-through STHMDs process the superimposing of the real environment and virtual contents based on the digital images and output it to users, while optical see-through STHMDs display virtual contents through the optics-based near-eyes display with users' normal view of the real scene kept. For both types of AR devices, the accuracy of visualization is essential. For example, in AR-based human-robot interaction, the inaccurate rendering of 3D virtual objects with respect to the real environment, will lead to users' mistaking operations, and therefore, causes an invalid tool path planning result. In spite of many works related to system calibration and error reduction for optical see-through STHMDs, there are few efforts at figuring out the nature and factors of those errors in video see-through STHMDs. In this paper, taking consumer-available AR video see-through STHMDs as an example, we identify error sources of registration and build a mathematical model of the display progress to describe the error propagation in the stereo video see-through systems. Then, based on the mathematical model of the system, the sensitivity of each error source to the final registration error is analyzed. Finally, possible solutions of error correction are suggested and summarized in the general video see-through STHMDs.

1 Introduction

Currently, various manufacturing enterprises are adopting the techniques of Industrial 4.0 (the fourth industrial revolution) [1] into their manufacturing processes to enhance productivity and reduce cost, through applying the Internet of Things (IoT), robotics, Cyber-Physical Systems (CPS), and Augmented Reality (AR) [2]. AR display is used for superimposing the physical environment with virtual content and providing an intuitive perception and interaction [3]. In the Human-Robot Collaborative (HRC) area, AR shows promising advantages in some manufacturing applications with industrial robots where the robot and human operators share a collective workspace [4], because multimedia interfaces implemented in AR systems are well adapted in such production environments [4]. Several surveys indicate AR robotics applications in medical, robot control and planning, Human-Robot Interaction (HRI), and swarm robot [5].

*Corresponding author.

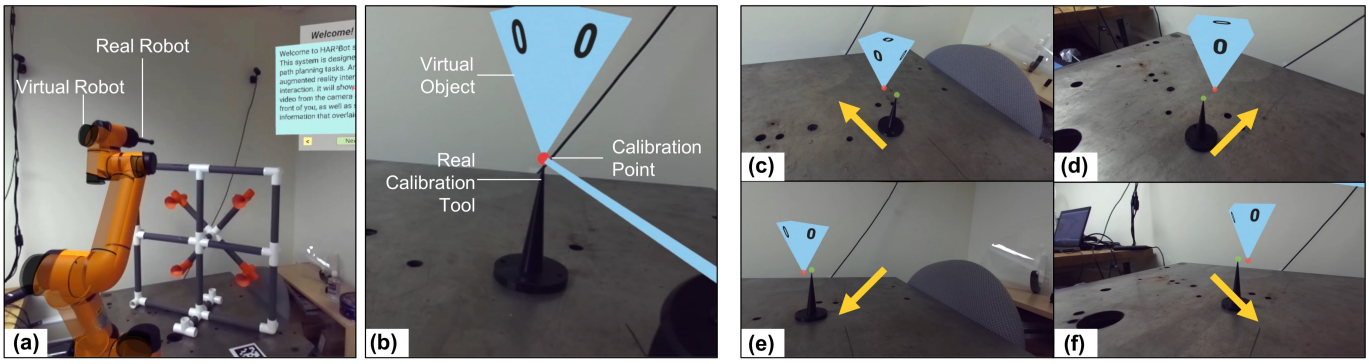

FIGURE 1. One primary problem in the AR system is misregistration. (a) the virtual robot model does not align perfectly with the real robot; (b) we put a virtual object on the top of a calibration tool, where is the calibration point, and the calibration tool is fixed in the real environment; (c-f) when observing the calibration tool from different viewpoint and distance, the virtual object does not align with the calibration tool.

Makhataeva and Varol [5] categorize different AR technology into mobile AR, wearable AR, and spatial AR. Particularly, wearable AR head-mounted displays (HMDs) are considered as the most ergonomic and reliable solutions in complicated manual tasks [6–8]. Thus, current trends in HRI undertake the highly-performed HMDs as the major medium for related applications [5]. Wearable AR HMDs provide a stereo photograph with depth perception and ergonomic user interaction for complex tasks through Near-Eye Display (NED) and inter-sensory interactions. Another advantage of AR HMDs is that the mobility of HMDs provides an immersive perception with intuitive interaction and 3D-space spatial registration [9]. In addition, compared with handheld devices, AR HMDs are hand-free and provide more flexibility [10].

Specifically, AR HMDs consist of optical see-through (OST) and video see-through (VST) HMDs [11]. The OST displays remain the direct view of the real world from users' eyes without an alternate viewpoint and field of view (FOV), while the cameras of the VST have variations from human eyeballs resulting in perspective conversion. OST provides a better depth perception of the real world, but it suffers from the limited FOVs and challenging of occlusion handing [12]. On the other hand, VST displays have outstanding features to handle the rendering and image processing problems, including the occlusion and consistency between real and synthetic views [12]. Due to these advantages, VST HMDs have been utilized in recent works [13,14].

There is a critical problem for VST HMDs, which is that the virtual object is difficult to be aligned with the real environment correctly. For example, as shown in Fig.1 (a), the virtual robot can not be aligned with the physical robot perfectly, because it is hard to transfer the accurate pose of the real robot in the virtual space. In the experimental setup shown in Fig.1 (b)-(f), the virtual pyramid object is supposed to be aligned with the top of the physical calibration tool (Fig.1 (b)). However, when users observe the virtual object from different distance (see Fig.1 (c) and (d)), and different viewpoints (see Fig.1 (e) and (f)), the virtual object cannot not be aligned with the calibration tool correctly. For HRI applications presented by Yang et al. [14], this type of error prevents further advancement that requires high accuracy.

The registration accuracy is also important in other applications. Pentenrieder and Meier mentioned the necessity of accuracy in industrial AR applications [15]. Additionally, 3D-AR devices applied in certain clinical applications that demand locating 3D models, for example, anatomy, heavily relied upon registration accuracy. Andrews et al. [16] summarized registration techniques in medical exercises mainly focused on OST HMDs (HoloLens), and suggested a feasible registration error of 2 mm in clinically relevant tasks. Therefore, the robustness and accuracy of robotics-related applications are essential, and a simple-to-implement registration method is in an urgent need [5]. Since there are multiple error sources from multiple components of the system, seldom previous works provide a comprehensive enumeration of the error sources and connected their effect on the displayed AR scene. To solve these issues, we present a registration model of VST HMDs, enumerate and categorize the error sources, and suggest possible corrections for each error source. Specifically, the main contributions of this paper are as follows:

1) A registration model of VST HMDs displays is built to represent the error propagation.
2) The registration error sources are enumerated and categorized based on the system structure model, and the nature of both static and dynamic errors of each error type are explained.
3) The possible error corrections of error sources are presented to improve the system registration accuracy.

2 Related Works

The concept of calibrating a VST HMDs system can be tracked to 1995. Tuceryan et al. [17] identified the calibration requirements and procedures for a monitor-based augmented reality system, which was the precursor of current VST HMDs. Their prototyping system is quite different from the ones nowadays, and especially, they visualized an AR image on a monitor. They generally described the calibration categories of an AR system. In 1997, Holloway and Richard [11] proposed four types of registration errors in the AR system, namely, linear registration errors, lateral registration errors, depth registration errors, and angular registration errors (shown in Fig.2).

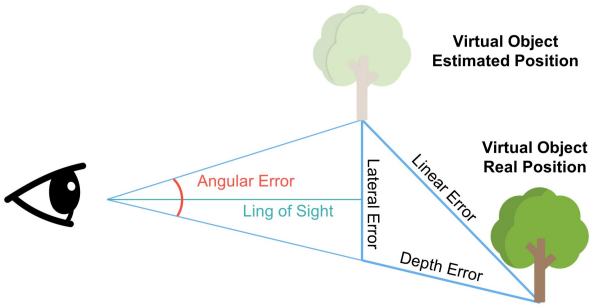

FIGURE 2. Four registration errors in AR systems: linear error, lateral error, depth error, and angular error [11].

Many factors weigh heavily for misregistration, so several researchers specialize in finding sources of registration errors. Registration errors are broadly classified into two categories: static and dynamic [18, 19]. The dynamic registration error used to dominate the AR system's registration error, due to the limitation of the hardware and the computation capacity [20]. For example, time delay as a dynamic error is the primary source in primitive AR systems [11]. There are five components of the AR system's latency that are summarized by Zheng: tracking delay, application delay, rendering delay, scanout delay, and display delay [19]. Previous research focuses on reducing dynamic errors through latency minimization [21], just-in-time image generation [22], predictive tracking [23], or video feedback (such as VST AR systems) [24]. This situation has been changed due to the rapid improvement of the hardware, such as graphic cards, processors, and tracking systems. Therefore, the weight of dynamics errors decreases.

Early research groups focused on developing specific technology that collectively calibrates their wearable HMDs in certain applications to reduce registration errors. Baillot et al. [25] implemented an easy and intuitive single point method to simultaneously align two transformations, namely from the sensor on HMD to display and from the base of the tracker to the world. This is a blended calibration method and it depends on the subjective consciousness of the operator. Nevertheless, they only took into account display errors and tracker alignment errors.

Then the majority of research prioritizes precise and stable tracking techniques. Zhou et al. [12] emphasized that spatial registration is much more important for mixed reality (MR) than virtual reality (VR) and classified AR tracking method into sensor-based, vision-based, and hybrid tracking techniques. However, Makhataeva et al. [5] mentioned the limitations of poor tracking stability in current wearable devices. Thus, there is various research conducted for the STHMDs calibration.

At the same time, since the camera works as an essential component in the VST HMDs system, camera calibration is another hot research question to improve AR registration. Two main classifications of strategies, namely linear and non-linear techniques, are applied in AR systems for camera calibration [26]. However, the current camera calibration methods applied in the wearable AR HMD system are computationally extravagant and manually extravagant [27]. To address this issue, researchers are seeking self-calibration and calibration-free methods [28]. Sahu et al. [29] reviewed the application of AI in AR techniques, and proposed some potential AI approaches for camera calibration in AR systems, including single image method (a single image frame as input) and non-single image method (multiple images as input).

3 Experimental System Setup

3.1 Hardware & Software

We built up our experiment platform based on the AR HRI system presented by Yang et al. [14]. As shown in Figure 4, the AR VST system is assisting users with robotics path planning tasks. Users can define waypoints and simulate the virtual robot in the AR scene to avoid obstacles in the real environment. The hardware of the AR VST system consists of an Oculus Rift headset, a Zed mini stereo camera, handheld controllers, and Oculus tracking sensors. The Zed mini camera captures the depth images, and the associated Zed mini SDKs together with the Oculus SDKs are integrated through the Unity 3D platform to handle the visualization and occlusion of the virtual objects and the physical scene.

This AR VST system is further integrated with a collaborative robot through Unity 3D and Robot Operating System (ROS) robot control platform. We built a virtual 3D world and put a virtual camera with the same setting (intrinsic and external parameters) as the physical camera to record the scene view at each frame. And then the rendering pipeline combines both images to render an AR image with Occlusion property. The pipeline of the VST AR system to generate and display an AR visualization is presented in Figure 3. It is noteworthy that at each frame, either the physical camera or the virtual camera capture a pair of images representing the two eyes view, which are rendered on

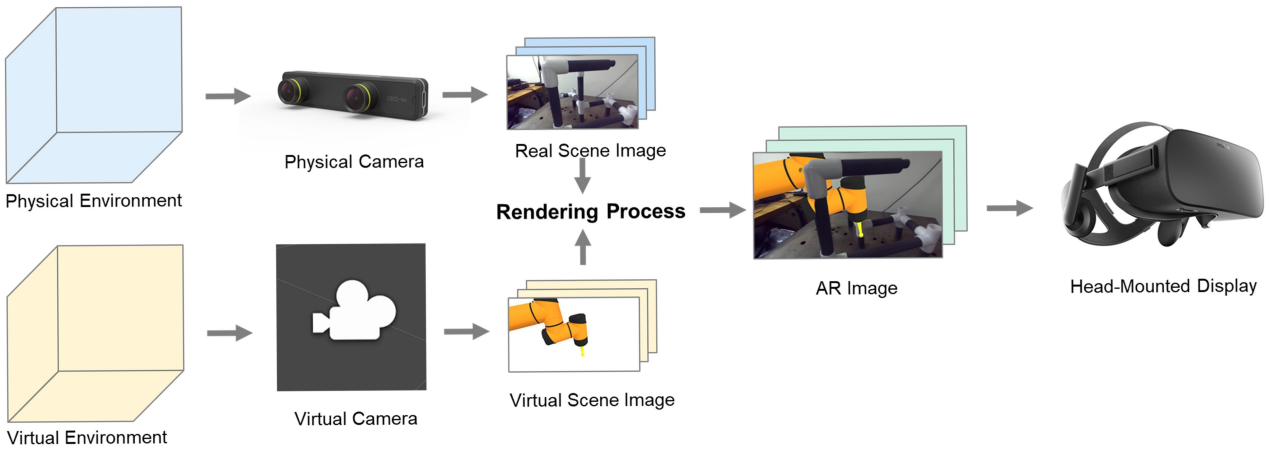

FIGURE 3. A stereoscopic video see-through (VST) augmented reality (AR) system combines video frames between a physical stereo camera and a virtual stereo camera and renders a stereo AR video which is displayed on a head-mounted display (HMD).

corresponding lenses inside of the HMD. The AR images have a resolution of 1280 * 720 pixels per eye with a refresh rate of 30Hz.

The virtual information in this HRI system includes robot model, path, waypoints, and UI browsers which are displayed in a 3D model in a virtual 3D environment. All the CAD models are kept in the appropriate size and relative location similar to the real work cell. A virtual camera is placed in this virtual environment to capture the virtual environment as output, which is a depth video stream. Depth information is used for occlusion rendering in the rendering process.

3.2 Registration Matrix

For a 3D AR system with a combination of virtual objects and the real world, it is crucial to have a reference coordinates system that locates the real and virtual components. Since we use a 2D image as an output of the system, a transformation matrix is required through different hardware. Normally, those objects stay at a fixed position. However, during the running time, users are free to move the HMDs and their viewpoint changes every frame, as well as the camera. Thus, the transformation matrix is complicated, where errors propagate easily and result in the misregistration in the final output 2D images.

There are assorted coordinates to achieve the final AR rendering ability, which is illustrated in Figure 5. The output images are located in a 2D image coordinates system (ICS), where the rendering process combines the image of real-world and virtual contents. The real-world image is obtained from the physical camera coordinates system (PCCS), which is movable following the users' motion. In order to synchronize the virtual camera's motion with one of the physical cameras, we need the camera's accurate pose in the World Coordinates System (WCS). A

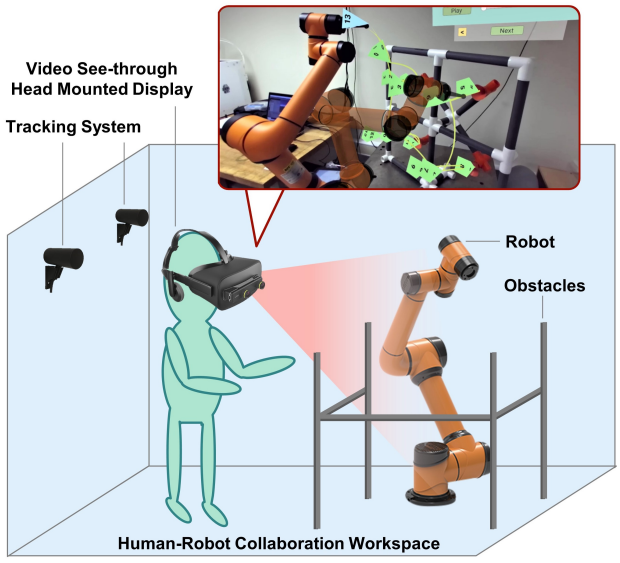

FIGURE 4. A video see-through augmented reality system is implemented for a human-robot interface in path planning tasks to avoid collisions. With the first-view stereo AR display, users can define some virtual waypoints and use a virtual robot for in-situ simulation.

fast and reliable solution is that, since the camera is mounted on the HMD, we can get the PCCS based on the Head-Mounted Display Coordinates System (HMDCS) with a tiny transformation. The Head-Mounted Display Coordinate System (HMDCS) is tracked by Oculus sensors in the Tracking Coordinates System (TCS). The virtual camera is located in the Virtual Camera Coordinates System (VCCS). Virtual HMD Coordinate System (VHMDCS) is where we synchronize the physical HMD pose.

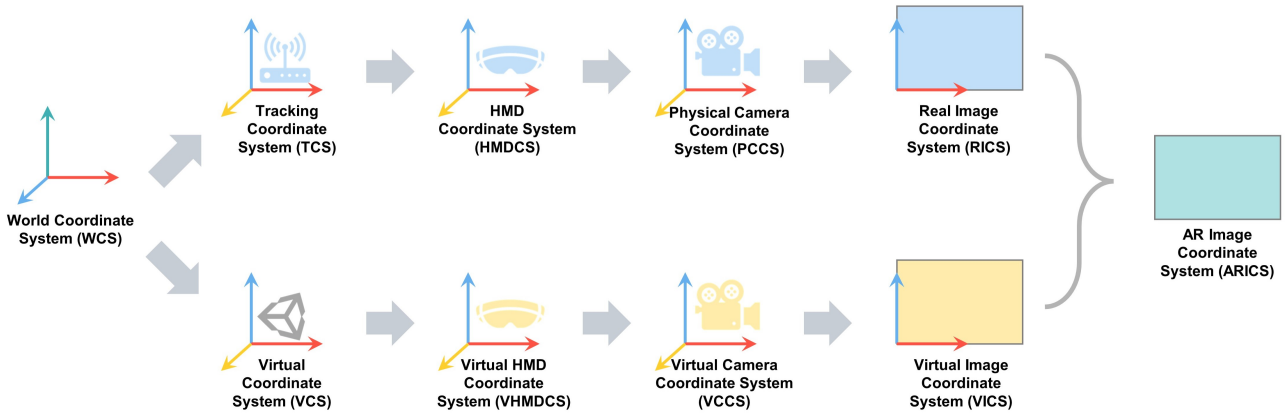

FIGURE 5. The transformations needed to compute through a pipeline in the VST AR systems.

Both VCCS and VHMDCD are located in the Virtual Coordinate System (VCS) built in Unity. Note that, all aforementioned 3D coordinate systems use a right-handed theory.

Camera Matrix Model The camera model presents a transformation from camera coordinates system to image coordinate system by using projective imaging geometry. The pinhole camera model [30] is assumed for the cameras, HMD display, and human eyes, which is an idealized model implemented in computer vision and graphic processing. However, this model cannot interpret nonlinear optical effects, like redial distortions, which is common property in current customer cameras.

The origin of the camera coordinates system, O_c, is the center of projection, with viewport pointing towards to $+z$ axis. The image plane is located at a distance of f, which is the focal length of the camera. A ray is targeting a point P_c and leaving an intersection, P_i, on the image plane. With homogeneous coordinates system, the camera coordinate point $P_c = [x_c, y_c, z_c, 1]^T$ can be mapped to the image point $P_d = [u, v, 1]^T$ through the similar triangles,

$$P_i = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f s_u x_c / z_c + u_0 \\ f s_v y_c / z_c + v_0 \\ 1 \end{bmatrix}, \tag{1}$$

where (u_0, v_0) accounts for the origin of the image plane, and s_u and s_v are the pixel size along the two axes of the image plane. If we let $f_u = f s_u$ and $f_v = f s_v$, the previous mapping will be adjusted to

$$P_i = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f s_u x_c / z_c + u_0 \\ f s_v y_c / z_c + v_0 \\ 1 \end{bmatrix} = \begin{bmatrix} f_u x_c / z_c + u_0 \\ f_v y_c / z_c + v_0 \\ 1 \end{bmatrix} \tag{2}$$

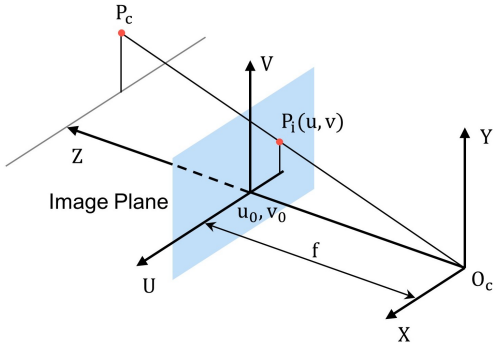

FIGURE 6. Schematic diagram of the pinhole camera model.

If we multiple a perspective scale parameter $w = z_c$ on both sides, the formula should be

$$wP_i = \begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = MP_c \tag{3}$$

Form now, we can represent the transformation between a point in spatial space and its image coordinates by a matrix-vector relationship. If we further decompose the equation to

$$wP_i = MP_c = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \end{bmatrix} P_c = K \begin{bmatrix} I & 0 \end{bmatrix} P_c, \tag{4}$$

where the matrix K is often referred to as the camera intrinsic matrix. However, two additional parameters can be added to this

matrix: skewness and distortion. In ideal, the image is skewed, which means the u and v axes are perpendicular to each other. Practically, that angle between the two axes is slightly larger or smaller than 90 degrees. Thus a camera intrinsic matrix accounting for skewness should be represented as

$$K = \begin{bmatrix} f_u & -f_u cot\theta & u_0 \\ 0 & f_v/sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

This method ignores distortion effects in the camera model matrix.

The transformation from the world coordinate to the camera reference system

We need an additional transformation to describe the object in the camera coordinate system because normally, the object is available and rigid in world coordinates. For a given point, this transformation is rigid as follows,

$${}^{C}P = {}^{C}_{W}M\,{}^{W}P = \left[\begin{array}{c|c} {}^{C}_{W}R & {}^{C}_{W}T \\ \hline 0\,0\,0 & 1 \end{array}\right] {}^{W}P \tag{6}$$

Where the ${}^{C}_{W}R$ is the 3×3 rotation matrix and the ${}^{C}_{W}T$ is the translation matrix, they are defined as the extrinsic parameters of the camera. Both of them are augmented into the transformation matrix, ${}^{C}_{W}M$, mapping a point from the world coordinate system to the camera coordinate system.

Thus, given a point in the WCS ${}^{W}P$, we can computer is camera coordinates by substituting Equation (6) into (4) as follows,

$${}^{I}w P = K \begin{bmatrix} I & 0 \end{bmatrix} {}^{C}P = K \begin{bmatrix} I & 0 \end{bmatrix} {}^{C}_{W}M\,{}^{W}P = K \begin{bmatrix} {}^{C}_{W}R & {}^{C}_{W}T \end{bmatrix} {}^{W}P \tag{7}$$

Since we want to get an accurate transformation matrix of the camera from the WCS and the camera is rigidly mounted on the HMD, we can depend on tracking the HMD by Infrared sensor to indirectly get the camera coordinate system. So the multiple transformation equation is

$${}^{C}P = {}^{C}_{H}M\,{}^{H}P = {}^{C}_{H}M\,{}^{H}_{T}M\,{}^{T}P = {}^{C}_{H}M\,{}^{H}_{T}M\,{}^{T}_{W}M\,{}^{W}P \tag{8}$$

4 Error Propagation

Based on the registration matrix proposed in section 3.2, the sources of final registration error are categorized into seven main classes, which are illustrated in figure 7.

4.1 Tracker Alignment Error

Tracker alignment error emerges when the transformation between Tracking Coordinate System and Virtual world CS is inaccurate. This transformation demands prior knowledge of the tracker's absolute pose in the world based. In other words, we need to know there are physical trackers put in the world. And we need to set the virtual world (the virtual tracker) base relating to the world coordinate.

For the implemented system, two sensors are mounted on the wall and remain in a fixed pose. So actually, we assume that the tracker-based coordinate system is absolutely fixed in the registration matrix. Since the virtual tracker coordinate is used defined based on the WCS and the WCS is rigid to TCS, we can manually set the VCS-to-TCS transformation as a fixed origin of the virtual 3d world, where is the base for handling other virtual components, such as computer-generated 3D objects, capturing cameras and rendering displays.

4.2 Tracking Error

Tracking errors arise from the measurement of HMD's pose by the sensors. This is the matrix ${}^{H}_{T}M$, where pose deviation exists from position and orientation. Since this transformation is mapped to the virtual space and affects the pose of virtual cameras. The tracking error is subject to the tracking techniques, several tracking sensors, and spatial arrangement. Tracking techniques take on various types including optical, electromagnetic, and acoustic signals and mechanical systems.

For the implementation of AR systems, Oculus Rift (HMD) and touch controllers apply constellations of infrared LEDs built-in, which can be recognized by two IR sensors fixed at a conventional desktop or wall and toward the workspace. Then back algorithms extrapolate positional movement values based on the placement of LEDs in a frame. One camera is enough, but additional cameras can improve the quality of tracking. In addition, other positional measurements, like magnetometer, gyroscope, and accelerometer, are also implemented in the headset to increase the accuracy.

4.3 Camera extrinsic calibration error

Camera calibration is an area of three-dimensional machine vision, including intrinsic and extrinsic parameters. Camera calibration in AR systems is a prerequisite in the virtual environment for rendering and tacking. Camera extrinsic parameters are related to the spatial position and orientation of the camera coordinate system relative to a certain world coordinate system. Therefore, camera extrinsic calibration error generates from the estimation of measurement of the camera's pose.

Specifically in the implemented system, since the world coordinate system is based on the fixed tracker coordinate system, the Camera extrinsic parameters are identified as the ${}^{T}_{C}M$. Then because the camera is rigidly mounted on the HMD and the

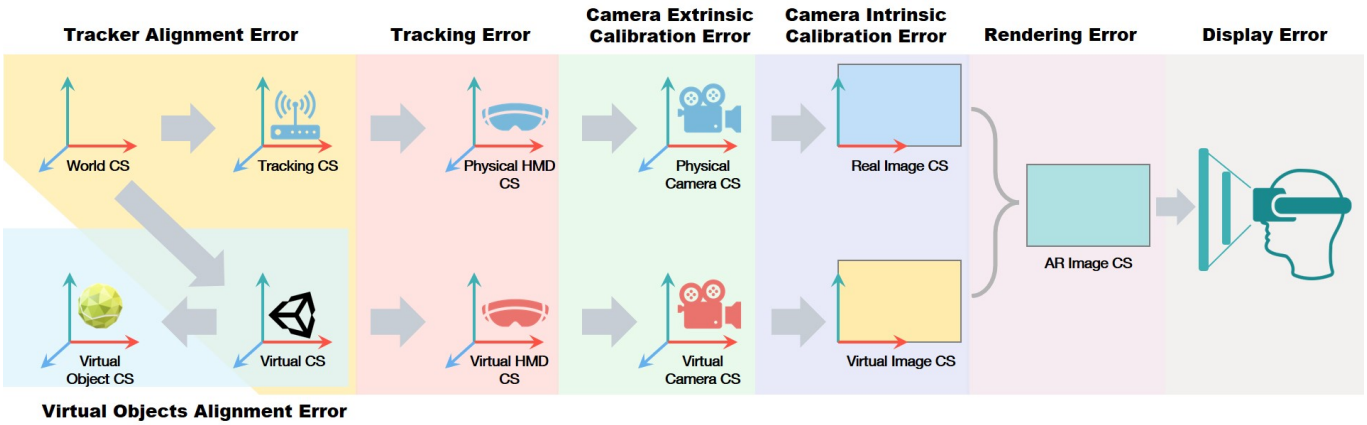

FIGURE 7. Registration errors are categorized into 7 types based on the processes of the VST AR system, namely, tracker alignment error, tracking error, camera extrinsic calibration error, camera intrinsic calibration error, rendering error, display error, and virtual objects alignment error.

HMD is tracked in the TCS, that transformation is decomposed into ${}^{T}_{C}M = {}^{T}_{H}M \cdot {}^{H}_{C}M$.

4.4 Camera intrinsic calibration error

Camera intrinsic calibration is the process of determining the internal geometric and optical characteristics of the cameras, including parameters such as focal length, optical center, distortion, and field-of-view (FOV). These parameters differ between products, but in most applications they are constant, and just once accurate calibration is enough. In practice, the camera intrinsic calibration in AR systems is a process to access accurate internal parameters of the real camera to recreate a virtual camera in the virtual environment [29].

For the implemented system, ZED software development kits (SDK) are applied in developing the application and provide a factory calibration profile. However, there are systematic errors due to the calibration limitation and unsystematic errors during the measurement process. As a result, intrinsic parameters may deviate from the real value, including focal length, image center, and radial and tangential distortion coefficients.

4.5 Rendering error

The rendering errors of registration refer to image combination and depth occlusion. The main registration error comes from that the virtual object should be rendered in front of the real scene but they disappear or are partially absent. Because in the rendering process, the first stage of generation of the pixel on the AR image depends on the depth information from both real environment image and virtual environment image. In other words, the nearer one will be rendered, which follows the basic principle of physics. However, accurate depth information gives rise to wrong selection. The virtual environment depth obtained from virtual cameras could be reliable, but the depth of the real environment is coarsely measured. The depth accuracy of the real environment is depending on camera resolution, detection distance, and depth estimation algorithms. In practice, mirrors and glasses restrain the depth measurement.

Dynamic rendering errors relate to the time consumption of the depth estimation and rendering process. If the virtual object appears in front of the real scene, there is a rendering pipeline, involving geometry calculation, textures, surface treatment, the viewers' perspective, and lighting.

4.6 Display Error

Most display errors occur from the systematic HMD hardware and display technique. A kind of display error is caused by the one situation that the actual eyepoint does not coincide with the corresponding modeled eyepoint. People have different-sized heads and they are available to adjust the optimum positioning. Thus the eyepoints vary from users as well as every adjustment. Furthermore, the stereo video is captured by the camera with a fixed eye separation of 63 mm. However, users have various eye separations and have different perceptions in reconstructing the depth world. Screen-space errors are another display error source that the image space is not correctly mapped to the screen of the HMDs. Screen or image limited resolution restricts the tiny object's registration. Other potential error sources include electronics and corresponding optical properties.

As the HMD used in the implementation, Oculus Rift uses a pair of cheap magnifying lenses. The thin-lens equations for spherical lenses are practical in modeling the optics. Monochromatic aberrations are the main problem in optics displaying, including spherical aberration, coma, astigmatism, field curvature, and distortion.

Dynamic display errors consist of the signal travel time through a wire connection or wireless transmission. In addi-

tion, the technique of screen lighting and refresh rate contribute to time delay. For example, AMOLED screens are applied in Oculus Rift, which can switch colors in less than a millisecond.

4.7 Virtual objects alignment error

Virtual objects alignment errors are encountered in aligning the virtual object data into the AR system. The virtual objects are placed in the virtual environment, which is based on the virtual coordinate system. However, the virtual coordinate system is invisible. And the transformation between VCS and virtual objects is defined in the virtual environment (Unity 3D) without respect to the real scene. Thus virtual objects alignment errors occur from the calibration of the specific virtual objects in the virtual space.

In the implementation system, a virtual robot should be aligned with the real robot, so that they can share the same workspace. Because all waypoints defined in the virtual space are required to be transferred into the robot base space. An open-loop registration approach is commonly used to place the virtual robot through a fiducial marker, which needs prior knowledge of the distance between the fiducial marker and the real base origin. The measurement errors propagate through the coordinate transformation to the virtual coordinate system origin.

5 Error Correction

5.1 Camera Intrinsic Calibration

Since the virtual camera uses the same parameters to form an image, the extrinsic and intrinsic camera parameters are required as prior knowledge at each frame. In other words, camera calibration is a process to estimate those parameters. Therefore, an essential step that should be done before setting up the system is to calibrate the physical camera and get the accurate intrinsic parameters, which are rigid and inherent to a given camera. Then we need to set up the virtual camera with obtained intrinsic parameters. Extrinsic parameters are external to the camera and change every frame, we will talk about how to correct the estimation in the next subsection.

Some corresponding SDKs provide camera calibration functions, which are available to certain devices like cameras, Microsoft Hololens, Magic Leap, Schenker META, Android devices, Apple devices, and Lightform projectors [29], the intrinsic calibration can be executed based on the integrated techniques for appropriate devices. Other derivative independent SDKs, such as Wikitude [31], Vuforia [32] and Kudan [33], also provide manual or automatic calibration through calibration patterns.

Several popular calibration methods are presented and widely applied for a general camera, such as Zhang's method [34], Direct Linear Transformation (DLT) [35], Tsai'salgorithm [36]. Particularly, linear calibration techniques are traditionally applied to compute the projection matrix [26]. We introduce a possible method to manually calibrate a VST HMD. The main idea behind it is that we reduce some given points based on the accessed images. Concretely, the intrinsic camera matrix K can be solved through Equation (7). We can implement some patterned image, like a checkboard, whose dimension is known. Zed mini is calibrated through the manufacturing process with a proprietary algorithm. The calibrated parameters are stored in the device and are shared by the SDK plug-in in unity to the virtual camera to form images.

Given a set of points in the WCS ${}^{W}P_i$ and corresponding image points ${}^{I}P_i$, a linear system of equations is established based on Equation (7),

$$ {}^{I}P_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = K \begin{bmatrix} {}^{C}_{W}R & {}^{C}_{W}T \end{bmatrix} {}^{W}P = M {}^{W}P_i = \begin{bmatrix} m_1 {}^{W}P_i / m_3 {}^{W}P_i \\ m_2 {}^{W}P_i / m_3 {}^{W}P_i \end{bmatrix} \quad (9) $$

Where, m_1,m_2 and m_3 are rows of the matrix M. To solve the 11 unknown parameters in Equation (7), at least 6 correspondences are required. In practice, more correspondences are applied with the least square method. Given n observed points, the entire linear system matrix is written as:

$$ \begin{bmatrix} {}^{W}P_1^T & O^T & -u_1 {}^{W}P_1^T \\ O^T & {}^{W}P_1^T & -v_1 {}^{W}P_1^T \\ & \vdots & \\ {}^{W}P_n^T & O^T & -u_n {}^{W}P_n^T \\ O^T & {}^{W}P_n^T & -v_n {}^{W}P_n^T \end{bmatrix} \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix} = Pm = o \quad (10) $$

In theory, $Pm = o$ is a overdetermined equation when $2n > 11$. Thus the purpose to solve this equation is using the least square minimization with a constrain:

$$ \begin{aligned} &\min_m \ \left\| \mathbf{P}m \right\|^2 \\ &subject\ to\ \left\| m \right\|^2 = 1 \end{aligned} \quad (11) $$

The singular value decomposition (SVD) method is applied by letting $P = UDV^T$ in solving this minimization problem, we can get m from the last column of V. But this solution is scaled m, in other words, the true value should be multiplied with a scale, ρ. Then based on Equation (5) and (9), the transform matrix is represented as:

$$M = \frac{1}{\rho}\begin{bmatrix} f_u r_1^T - f_u \cot\theta r_2^T + u_0 r_3^T & f_u t_x - f_u \cot\theta t_y + u_0 t_z \\ \frac{f_v}{\sin\theta} r_2^T + v_0 r_3^T & \frac{f_v}{\sin\theta} t_y + v_0 t_z \\ r_3^T & t_z \end{bmatrix} = \begin{bmatrix} A & b \end{bmatrix} = \begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{12}$$

where r_1^T, r_2^T and r_3^T are the three rows of the extrinsic rotation matrix, t_1, t_2, and t_3 are the three components of extrinsic translation matrix. The intrinsic parameters are solved as follows,

$$\begin{aligned} \rho &= \pm\frac{1}{||a_3||} \\ u_0 &= \rho^2(a_1 \cdot a_3) \\ v_0 &= \rho^2(a_2 \cdot a_3) \\ \theta &= \cos^{-1}(-\frac{(a_1 \times a_3)\cdot(a_2 \times a_3)}{||a_1 \times a_3|| \cdot ||a_2 \times a_3||}) \\ f_u &= \rho^2||a_1 \times a_3|| \sin\theta \\ f_v &= \rho^2||a_2 \times a_3|| \sin\theta \end{aligned} \tag{13}$$

Then, the extrinsic parameters are computed,

$$\begin{aligned} r_1 &= \frac{a_2 \times a_3}{||a_2 \times a_3||} \\ r_2 &= r_3 \times r_1 \\ r_3 &= \rho a - 3 \\ T &= \rho K^{-1} b \end{aligned} \tag{14}$$

5.2 Consideration With Distortion

The pinhole camera doesn't account for the camera distortion effects. There are several distortion models proposed in other research. Heikkilä [37] proposed two pairs of parameters to describe radial distortion and tangential distortion respectively. The radial distortion can be represented by,

$$\begin{bmatrix} \Delta u_r \\ \Delta v_r \end{bmatrix} = \begin{bmatrix} K_x fX/Z(k_1 r^2 + k_2 r^4) \\ K_y fY/Z(k_1 r^2 + k_2 r^4) \end{bmatrix} \tag{15}$$

Where $r = \frac{f}{Z}\sqrt{K_x X^2 + K_y Y^2}$, k_1 and k_2 are coefficients for radial distortion. Δu_r and Δv_r are radial distortion effects.

The tangential distortion can be represented by,

$$\begin{bmatrix} \Delta u_t \\ \Delta v_t \end{bmatrix} = \begin{bmatrix} 2t_1 \frac{f^2}{Z^2} K_x K_y XY + t_2(r^2 + 2\frac{f^2}{Z^2} K_x^2 X^2) \\ t_1(r^2 + 2\frac{f^2}{Z^2} K_y^2 Y^2) + 2t_2 \frac{f^2}{Z^2} K_x K_y XY \end{bmatrix} \tag{16}$$

Where t_1 and t_2 are coefficients for tangential distortion. Δu_t and Δv_t are tangential distortion effects.

Therefore, an accurate camera model by combining the pinhole model with the correction for the radial and tangential distortion is written as,

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} u + \Delta u_r + \Delta u_t \\ v + \Delta v_r + \Delta v_t \end{bmatrix} \tag{17}$$

6 Conclusion and future works

Taking an AR implementation in human-robot interactions, we observed registration errors undermining the visualization accuracy. This registration error is intolerable in defining the waypoints of the robot end-effector in the real 3D cartesian space. Since the path is planned through the virtual waypoints in the robot working space, the misalignment of the final planned path is caused by the misregistration of the virtual waypoints, which impedes the accuracy of the robot path planning task. In this paper, we first developed a mathematical model of registration propagation in VST AR displays. Then we categorized and enumerated the registration error sources based on that model. The nature of both static and dynamic errors existing in each error type was explained. Some possible error corrections of influential error sources were summarized and presented to improve the system registration accuracy.

This work provided a qualitative analysis of the error source and sensitivity of the visualization error. A possible limitation is the missing quantitative comparison among each error source due to a lack of precise visualization as ground truth. We will firstly focus on the steps related to camera extrinsic error and propose a mathematical model to enable a quantitative analysis in the future.

Several open-loop registration methods were proposed and implemented in correcting registration errors from each step. However, during the VST AR HMDs frame, numerous error sources make the registration tedious and difficult. On the other hand, errors propagate through the pipeline and have an obvious impact on the final visualization. For future work, we would like to separate the registration process of VST AR HMDs into two main portions, namely system-related registration and task registration. System registration refers to the systematic AR display

method and depends on various devices. Task-related registration is specialized for tasks, scenarios, or applications, particularly, for the virtual objects alignment calibration.

As for system-related registration calibration, one possible solution is applying a closed-loop calibration method for certain VST AR systems, which has not been widely applied in current research. The reason is that closed-loop calibration methods have advantages in handling multiple registration errors and minimizing them at a global view without identifying the exact error sources.

Task-related registration is related to applications, where we need to align the predefined models in the AR space, such as specific industrial robot series, cyber work cells, and certain machines. We will propose a novel virtual objects alignment calibration algorithm to localize the virtual robot model with the physical robot.

ACKNOWLEDGMENT

This work is partially supported by the National Science Foundation under Award No. 20336157 and the Grant Writing Bootcamp Funding 2020 from the Rochester Institute of technology.

REFERENCES

[1] Mehrpouya, M., Dehghanghadikolaei, A., Fotovvati, B., Vosooghnia, A., Emamian, S. S., and Gisario, A., 2019. "The potential of additive manufacturing in the smart factory industrial 4.0: A review". *Applied Sciences,* ***9***(18), p. 3865.

[2] Fraga-Lamas, P., Fernandez-Carames, T. M., Blanco-Novoa, O., and Vilar-Montesinos, M. A., 2018. "A review on industrial augmented reality systems for the industry 4.0 shipyard". *Ieee Access,* ***6***, pp. 13358–13375.

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

[4] Krot, K., and Kutia, V., 2018. "Intuitive methods of industrial robot programming in advanced manufacturing systems". In International Conference on Intelligent Systems in Production Engineering and Maintenance, Springer, pp. 205–214.

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

[6] Sielhorst, T., Bichlmeier, C., Heining, S. M., and Navab, N., 2006. "Depth perception–a major issue in medical ar: evaluation study by twenty surgeons". In International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, pp. 364–372.

[7] Vávra, P., Roman, J., Zonča, P., Ihnát, P., Němec, M., Kumar, J., Habib, N., and El-Gendi, A., 2017. "Recent development of augmented reality in surgery: a review". *Journal of healthcare engineering,* ***2017***.

[8] Cutolo, F., Fida, B., Cattari, N., and Ferrari, V., 2019. "Software framework for customized augmented reality headsets in medicine". *IEEE Access,* ***8***, pp. 706–720.

[9] Zhu, Z., Liu, C., and Xu, X., 2019. "Visualisation of the digital twin data in manufacturing by using augmented reality". *Procedia Cirp,* ***81***, pp. 898–903.

[10] 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, International Society for Optics and Photonics, p. 101970V.

[11] Holloway, R. L., 1997. "Registration error analysis for augmented reality". *Presence: Teleoperators & Virtual Environments,* ***6***(4), pp. 413–432.

[12] 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.

[13] Cao, Y., Wang, T., Qian, X., Rao, P. S., Wadhawan, M., Huo, K., and Ramani, K., 2019. "Ghostar: A time-space editor for embodied authoring of human-robot collaborative task with augmented reality". In Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, pp. 521–534.

[14] 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.

[15] Pentenrieder, K., Meier, P., et al., 2006. "The need for accuracy statements in industrial augmented reality applications". In 5th IEEE and ACM International Symposium on Mixed and Augmented Reality. University of California at Santa Barbara (USA).

[16] Andrews, C. M., Henry, A. B., Soriano, I. M., Southworth, M. K., and Silva, J. R., 2020. "Registration techniques for clinical applications of three-dimensional augmented reality devices". *IEEE journal of translational engineering in health and medicine,* ***9***, pp. 1–14.

[17] 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.

[18] Axholt, M., 2011. "Pinhole camera calibration in the presence of human noise". PhD thesis, Linköping University Electronic Press.

[19] Zheng, F., 2015. "Spatio-temporal registration in augmented reality". PhD thesis, The University of North Carolina at Chapel Hill.
[20] Bauer, M., 2007. "Tracking errors in augmented reality". PhD thesis, Technische Universität München.
[21] Papadakis, G., Mania, K., and Koutroulis, E., 2011. "A system to measure, control and minimize end-to-end head tracking latency in immersive simulations". In Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry, pp. 581–584.
[22] Jerald, J., Fuller, A., Lastra, A., Whitton, M., Kohli, L., and Brooks, F., 2007. "Latency compensation by horizontal scanline selection for head-mounted displays". In Stereoscopic Displays and Virtual Reality Systems XIV, Vol. 6490, International Society for Optics and Photonics, p. 64901Q.
[23] Azuma, R. T., 1995. "Predictive tracking for augmented reality". PhD thesis, University of North Carolina at Chapel Hill.
[24] Kanbara, M., Okuma, T., Takemura, H., and Yokoya, N., 2000. "A stereoscopic video see-through augmented reality system based on real-time vision-based registration". In Proceedings IEEE Virtual Reality 2000 (Cat. No. 00CB37048), IEEE, pp. 255–262.
[25] Baillot, Y., Julier, S. J., Brown, D., and Livingston, M. A., 2003. "A tracker alignment framework for augmented reality". In The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings., IEEE, pp. 142–150.
[26] Malek, S., Zenati-Henda, N., Belhocine, M., and Benbelkacem, S., 2008. "Calibration method for an augmented reality system". *World Academy of science, Engineering and Technology,* ***45***, pp. 309–314.
[27] Donné, S., De Vylder, J., Goossens, B., and Philips, W., 2016. "Mate: Machine learning for adaptive calibration template detection". *Sensors,* ***16***(11), p. 1858.
[28] Taketomi, T., Okada, K., Yamamoto, G., Miyazaki, J., and Kato, H., 2014. "Camera pose estimation under dynamic intrinsic parameter change for augmented reality". *Computers & graphics,* ***44***, pp. 11–19.
[29] Sahu, C. K., Young, C., and Rai, R., 2021. "Artificial intelligence (ai) in augmented reality (ar)-assisted manufacturing applications: a review". *International Journal of Production Research,* ***59***(16), pp. 4903–4959.
[30] Potmesil, M., and Chakravarty, I., 1981. "A lens and aperture camera model for synthetic image generation". *ACM SIGGRAPH Computer Graphics,* ***15***(3), pp. 297–305.
[31] Wikitude, G. Wikitude augmentedreality:the world's leading cross-platform ar sdk. https://www.wikitude.com/.
[32] PTC. Vuforia: Market-leading enterprise ar—ptc. https://www.ptc.com/en/products/augmented-reality/vuforia.
[33] Inc, K. Kudan. https://www.kudan.io/.
[34] Zhang, Z., 1999. "Flexible camera calibration by viewing a plane from unknown orientations". In Proceedings of the seventh ieee international conference on computer vision, Vol. 1, Ieee, pp. 666–673.
[35] Abdel-Aziz, Y., and Karara, H., 1971. "Direct linear transformation into object space coordinates in close-range photogrammetry, in proc. symp. close-range photogrammetry". *Urbana-Champaign*, pp. 1–18.
[36] Tsai, R., 1987. "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses". *IEEE Journal on Robotics and Automation,* ***3***(4), pp. 323–344.
[37] Heikkila, J., and Silvén, O., 1997. "A four-step camera calibration procedure with implicit image correction". In Proceedings of IEEE computer society conference on computer vision and pattern recognition, IEEE, pp. 1106–1112.