IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 21, 2020, accepted December 8, 2020, date of publication December 11, 2020,
date of current version December 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3044184

Alignment-Free Offline Calibration of Commercial Optical See-Through Head-Mounted Displays With Simplified Procedures

XUE HU[1], FERDINANDO RODRIGUEZ Y BAENA[1], (Member, IEEE), AND FABRIZIO CUTOLO[2], (Member, IEEE)
[1]Mechatronics in Medicine Laboratory, Imperial College London, London SW7 2AZ, U.K.
[2]Department of Information Engineering, University of Pisa, 56122 Pisa, Italy

Corresponding author: Xue Hu (xue.hu17@imperial.ac.uk)

ABSTRACT Despite the growing availability of self-contained augmented reality head-mounted displays (AR HMDs) based on optical see-through (OST) technology, their potential applications across highly challenging medical and industrial settings are still hampered by the complexity of the display calibration required to ensure the locational coherence between the real and virtual elements. The calibration of commercial OST displays remains an open challenge due to the inaccessibility of the user's perspective and the limited hardware information available to the end-user. State-of-the-art calibrations usually comprise both offline and online stages. The offline calibration at a generic viewpoint provides a starting point for the subsequent refinements and it is crucial. Current offline calibration methods either heavily rely on the user-alignment or require complicated hardware calibrations, making the overall procedure subjective and/or tedious. To address this problem, in this work we propose two fully alignment-free calibration methods with less complicated hardware calibration procedures compared with state-of-the-art solutions. The first method employs an eye-replacement camera to compute the rendering camera's projection matrix based on photogrammetry techniques. The second method controls the rendered object position in a tracked 3D space to compensate for the parallax-related misalignment for a generic viewpoint. Both methods have been tested on Microsoft HoloLens 1. Quantitative results show that the average overlay misalignment is fewer than 4 pixels (around 1.5 mm or 9 arcmin) when the target stays within arm's reach. The achieved misalignment is much lower than the HoloLens default interpupillary distance (IPD)-based correction, and equivalent but with lower variance than the Single Point Active Alignment Method (SPAAM)-based calibration. The two proposed methods offer strengths in complementary aspects and can be chosen according to the user's needs. We also provide several update schemes for the two methods that can be integrated for an on-line viewpoint-dependent refinement of the calibration parameters. Both methods have been integrated into a Unity3D-based framework and can be directly applied to Unity-assisted devices.

INDEX TERMS Augmented reality, display-relative calibration, OST-HMD calibration.

I. INTRODUCTION

Visual Augmented Reality (AR), which supplements the user-perceived reality with computer-generated information, is quickly becoming a powerful tool to improve the experience of visual assistance. Within the AR domain, two major modalities exist for the content display: Video See-Through (VST) and Optical See-Through (OST). For VST displays, the user's direct eyesight is blocked: the view of the real scene is recorded by a world-facing RGB camera mounted on the HMD. The camera views are first digitally blended with virtual contents and then rendered on the display on the fly. By contrast, OST displays maintain almost unaltered the direct view of the world through a special semi-transparent optical combiner on which the computer-generated contents are being projected [1]. The ability to preserve the user's direct perception of the real world makes OST displays preferable tools for those activities with high safety requirements (e.g., surgical guidance). AR solutions based on head-mounted displays (HMDs) are

The associate editor coordinating the review of this manuscript and approving it for publication was Songwen Pei.

the most suitable output medium to support the hands-free and ergonomic interaction with the augmented scene [2], [3]. OST-HMDs have been widely researched to aid complex manual tasks, such as surgical navigation and training [4], [5] and industrial production and logistics [6]. OST-HMDs have already been identified as a key asset to enable technology within the fourth industrial revolution (i.e., Industry 4.0) [7], [8].

Nevertheless, successful deployment of OST-HMDs across highly challenging medical and industrial settings is still hampered by the complexity of the display calibration procedures required to ensure locational coherence between the real and the virtual elements [9]. As any misalignment between the virtual content and the real world may cause discomfort, confuse or even mislead users, display calibration is of the utmost importance [10]. In VST systems, the environment is recorded by one or two cameras and displayed to the user: the view of the real world is mediated by the camera(s). By contrast, in OST displays the ''video stream'' directly comes from the user's eye [11], and therefore it is not possible to use standard image processing techniques to align virtual contents with the scene [12]–[14]. Effective and efficient OST calibration represents an open research problem.

Thanks to the advance in optics design and embedded computational power, an increasing number of self-contained commercial OST-HMDs are now available on the market, with compact size and affordable price [15]. Display calibration is often simplified for these devices to improve usability, resulting in the sub-optimal AR overlay accuracy. While this is tolerable for ''gaming'' experience, calibration must be improved in applications for which both accuracy and convenience are equally important [16]. However, improving the calibration for these commercial systems can be rather challenging, as most of them are not open-source and the access to hardware parameters is often restricted by compatible interfaces [17].

To reduce the burden on users in terms of time and workload, in recent years much research effort has been dedicated to the implementation of two-step calibration procedures [9]. The first step, commonly performed offline in a controlled setup, aims to estimate the hardware-related display parameters for an arbitrary viewpoint position. The second step (i.e., online) subsequently updates the viewpoint-related calibration parameters by either performing additional but fewer user alignments [18], or alternatively, by adopting automatic algorithms that exploit eye-tracking cameras [10], [19]–[21]. The quality of the first step calibration is therefore paramount as it provides a starting point for the subsequent viewpoint-dependent refinements. For such offline stage calibration, some methods rely on multiple user alignments between real and virtual features [22], [23]. Those alignment-based methods can be easily implemented in hardware but they are tedious (i.e. several alignments are required) and subjective (i.e. the error increases with the poor-quality alignments performed by inexperienced users). Alternatively, alignment-free methods such as the display-relative calibration (DRC) proposed by Owen *et al.* [24] require several hardware calibration steps to model the viewpoint-display system as a pinhole camera. These procedures may be too complicated to replicate with commercial headsets outside laboratory environments.

Ideally, the calibration procedure should entail an alignment-free first phase that requires few or no hardware calibrations, and an easy-to-implement second phase. To fill the research gap, this article aims to achieve the alignment-free first stage calibration with less complicated hardware calibrations. For this purpose, we present two solutions, a camera-based and an object-based calibration method. Both methods are fully alignment-free and the result can be updated by several conventional online schemes. Our camera-based method utilises photogrammetry techniques for the estimation of a generic viewpoint-display model. It requires a simpler setup for the hardware-related calibration compared to the DRC method proposed by Owen *et al.*. Our object-based approach, unlike other state-of-the-art methods, directly manipulates the tracked 3D location of the target without modelling the viewpoint-display system as a pinhole camera. The parallax correction relies on the tracked ''gaze'' between the target and the viewpoint. To expand their applicability, we integrate the two methods in an AR experience development engine, Unity3D (Unity Technologies, San Francisco, US), so that our methods can directly be applied to any device supported by such game engine (e.g., Magic Leap, Microsoft HoloLens, Google Glass, etc.). A built solution is available for readers to test.

The main contributions of our work include:

- A camera-based method that estimates the projection model of the display for a generic viewpoint based on an homography transformation. The method requires no strict hardware calibration;
- An object-based method that effectively corrects the virtual-to-real misalignment without the pinhole camera model assumption and in a ''black-box'' fashion. The method requires no knowledge about projection properties;
- An experimental implementation and validation of the two methods on Microsoft HoloLens 1;
- A Unity3D implementation of the two methods, so that any Unity3D-supported AR devices can be benefited.

The paper is organised as follows: first, we briefly introduce related works. Next, we explain the rationale behind the two proposed methods, as well as the required calibration steps. We then describe the implementation of the two methods in Microsoft HoloLens and the tests designed for the performance evaluation. Results are compared with some state-of-the-art calibration methods. Finally, a suggestion on the online update schemes and a discussion of the two methods with other state-of-the-art methods are provided, alongside conclusions and future work.

II. NOTATION AND CONVENTION

The following notation is used throughout this article. Spatial coordinates are denoted by uppercase letters, such as the world coordinate system W. Scalars are denoted by lowercase letters, such as the focal length f. 2D/3D points/vectors are denoted by lowercase bold letters with a superscript denoting the reference coordinate system (e.g. a 3D point in the world $\mathbf{v}^W$). Matrices are denoted by uppercase bold letters, such as a rigid transformation ${}^B_A\mathbf{M}$ from coordinate A to B, the intrinsic matrix associated to a generic pinhole camera $\mathbf{K}$, and a planar homography transformation $\mathbf{H}$. A 4×4 transformation can also be expressed by a 3×3 rotation matrix ${}^B_A\mathbf{R}$ and a 3×1 translation vector ${}^B_A\mathbf{t}$. For example, the rigid transformation between two corresponding points in the reference system A and B is (both expressed in homogeneous coordinates):

$$\mathbf{v}^B = {}^B_A\mathbf{M}\mathbf{v}^A = \begin{bmatrix} {}^B_A\mathbf{R} & {}^B_A\mathbf{t} \\ 0 & 1 \end{bmatrix} \mathbf{v}^A = \begin{bmatrix} {}^B_A\mathbf{R} & {}^B_A\mathbf{t} \end{bmatrix} \mathbf{v}^A \tag{1}$$

III. RELATED WORK

A. PINHOLE CAMERA MODEL

The combined eye-display system of an OST display is commonly modeled as a general off-axis pinhole rendering camera. This pinhole camera model provides the basis for most of the state-of-the-art calibration methods. The nodal point of the user's eye corresponds to the projection centre of the pinhole camera E and the see-through virtual screen corresponds to the camera image plane S. The intrinsic matrix of a pinhole camera model can be expressed as:

$${}^{\text{off-E}}\mathbf{K} = \begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where f_u and f_v are the focal lengths of the see-through display in pixels and they are proportional to the distances between the image plane and the pinhole camera projection center. For cameras with non-perfectly square pixels, f_u and f_v are unequal. (c_u, c_v) defines the principal point, which is the intersection between the principal axis of the display and its image plane (Figure 1). For off-axis cameras, both c_u and c_v are not zero. s is the skew factor that is non-zero if the axes of the image plane are not orthogonal. The intrinsic matrix maps an arbitrary point $\mathbf{v}$ (in 3×1 format) in the rendering camera space E to the associated 2D point $\mathbf{i}$ on the display plane:

$$\lambda \mathbf{i}^E = {}^{\text{off-E}}\mathbf{K}\mathbf{v}^E \tag{3}$$

where λ is a generic scale factor due to the equivalence between points in homogeneous coordinates. In practice, points are tracked in a 3D coordinate (e.g., W). For the calculation of the overall perspective projection $\mathbf{P}$ that maps a tracked point onto the display plane, the extrinsic transformation from W to E also needs to be encapsulated:

$$\lambda \mathbf{i}^E = \underbrace{{}^{\text{off-E}}\mathbf{K}}_{\text{intrinsic}} \underbrace{\begin{bmatrix} {}^E_W\mathbf{R} & {}^E_W\mathbf{t} \end{bmatrix}}_{\text{extrinsic}} \mathbf{v}^W = {}^E\mathbf{P}\mathbf{v}^W \tag{4}$$

The resultant overall projection ${}^E\mathbf{P}$ is a 3×4 matrix with 11 independent parameters.

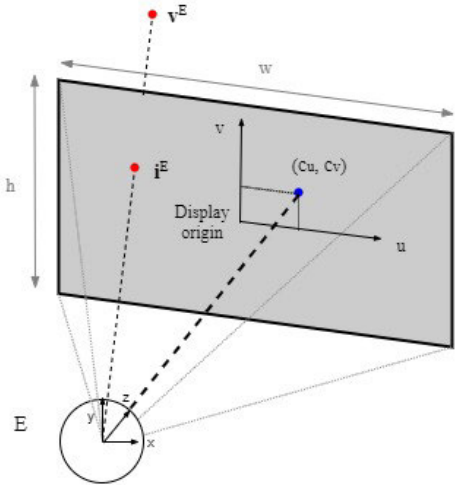


FIGURE 1. Representation of the off-axis eye-display pinhole model comprising the eye as projection center and the see-through virtual screen as image plane.

B. OST HMD CALIBRATION

OST HMD calibration aims to estimate the projection matrix ${}^E\mathbf{P}$ by which the rendered pixel can be aligned with the target perceived in the user's line-of-sight. State-of-the-art methods can be divided into manual and automatic methods, depending on whether user-instructed alignments are required.

1) MANUAL CALIBRATION

Manual calibration procedures can be done either in one step or two steps. In one step solutions, all 11 unknown parameters of the projection matrix ${}^E\mathbf{P}$ are directly solved by using at least 6 pairs of user alignment between tracked 3D reference points (i.e., $\mathbf{v}^W$ in (4)) and 2D image points (i.e., $\mathbf{i}^S$) displayed on the see-through display. Thus, the projection relation is determined in a black-box fashion (i.e., without accessing rendering properties) [21]. The most widely applied example is the Single Point Active Alignment Method (SPAAM) introduced by Tuceryan *et al.* [22]. These methods are tedious and time-consuming as they require many reliable alignments per calibration. To increase usability and lessen the burden on the users, the overall calibration can be broken into two phases based on the pinhole camera model: a first offline phase in which all the projection parameters of the OST display are determined through a sort of ''factory calibration'', ideally in a controlled setup, and a second online phase in which the calibration is refined for a small subset of viewpoint-dependent parameters. The first stage can be a standard SPAAM calibration [22] or alternatively, an alignment-free Display-Relative Calibration (DRC) [24] that uses multi-view captures to reconstruct the 3D virtual display. An online stage could then be used to update the estimated projection by applying a 2D screen warping based on a few extra pairs of user alignments [18], [24].

2) AUTOMATIC CALIBRATION

Automatic calibration methods aim to free users from the manual prior-to-use alignments during the online stage. A fist

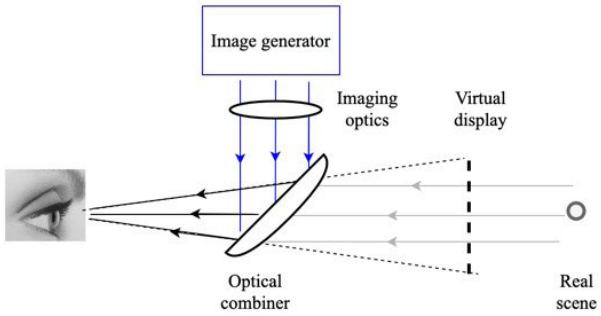


FIGURE 2. Main components of a near-eye optical see-through headset.

offline calibration phase is here required to estimate the hardware-related parameters, whereas during the online session eye-trackers, integrated into the HMD, are used to track the user's viewpoint, from which the user-dependent component of ${}^{E}\mathbf{P}$ is updated in real-time. Examples include Corneal-Imaging Calibration (CIC) [10] and INteractionfree DIsplay CAlibration (INDICA) [19], [21]. Therefore, these automated methods cannot disregard an offline calibration phase dedicated to the estimation of the physical display properties such as physical focal length. These parameters can be calibrated by a DRC [19], [24] procedure, roughly measured by a manually focused camera [19], or decomposed from the projection matrix calculated by an interactive SPAAM [19].

3) IMPLEMENTATION IN COMMERCIAL AR HEADSETS

The basic architecture of any Near-eye-Display consists of three main components (Figure 2): the image generator (i.e., the microdisplay where the virtual images are generated), the optical combiner that merges virtual and real contents together, and the imaging optics that magnify and collimate the virtual image at a comfortable viewing distance [25].

For commercial headsets, some rendering-related parameters, at least in their ideal factory specifications (e.g., display resolution in pixels, angle of view, focal length in pixels, etc.) are provided by the manufacturer, so that different rendering effects can be explored for the gaming experience. By contrast, physical optical parameters such as the focal length of the eyepiece of the display are not fully available to the end-user. Since these properties are explicitly considered during the offline calibration, hardware-related calibration is inevitable to ensure reliable results.

Compared to the DRC method, SPAAM-like methods are easier to implement due to their weak reliance on specific hardware [9]. Azimi *et al.* proposed a black-box SPAAM-based method that focuses on the transformation from a tracked 3D object $\mathbf{v}^{W}$ to its 3D representation in the rendering camera frame [17]. 20 user alignments are needed for their calibration. The performance was tested on HoloLens and Moverio BT-300 with both head-anchored and world-anchored tracking. Guo *et al.* implemented an online SPAAM-based calibration method for HoloLens [11]. The display was first calibrated in the entire workspace with a time-consuming offline calibration by 9×20 alignment tasks. The transformation was then corrected online using 5 additional alignments prior to every use. Itoh *et al.* tested its automatic INDICA calibration method on NVIS nVisor ST60 [21]: here the authors used a simplified DRC method to calibrate the physical display properties during the offline stage prior to their online viewpoint-dependent "recycled setup".

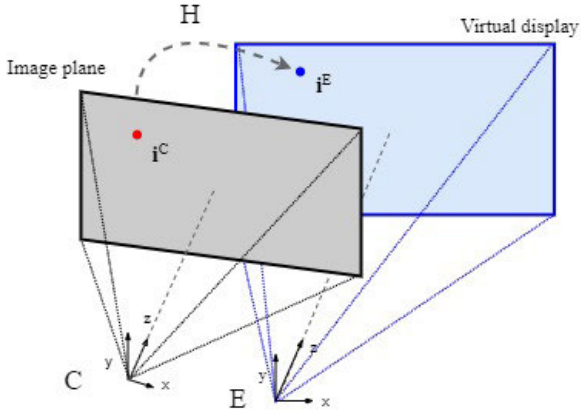


FIGURE 3. Homography transformation for the camera-based calibration. *C* is a generic viewpoint camera and *E* is the on-axis camera relative to the display.

C. HOMOGRAPHY CORRECTION BETWEEN PINHOLE CAMERAS

Instead of directly handling the off-axis intrinsic matrix of the OST rendering camera system, as done in most state-of-the-art OST calibration methods, a planar homography transformation can be modelled to relate the off-axis model ${}^{\text{off-E}}\mathbf{K}$ with the ideal on-axis model of the rendering camera ${}^{\text{on-E}}\mathbf{K}$ (Figure 3). More details of this homography-based model of off-axis rendering camera can be found in [26]. Here we report its main steps.

To estimate the homography correction, a viewpoint camera C, used as a replacement of the user's eye, is placed within the eye-box of the see-through display.

The points (in 3×1 homogeneous format) displayed on the image plane of C can be related to the points on the image plane of the on-axis OST display E through a planar homography:

$$\lambda \mathbf{i}^{E} = {}^{E}_{C}\mathbf{H}\mathbf{i}^{C} \tag{5}$$

where $\mathbf{i}^{C}$ are generated by the perspective projection relation introduced in (4):

$$\lambda \mathbf{i}^{C} = {}^{C}\mathbf{K}\left[{}^{C}_{W}\mathbf{R}\ {}^{C}_{W}\mathbf{t}\right]\mathbf{v}^{W} \tag{6}$$

and where the plane-induced homography ${}^{E}_{C}\mathbf{H}$ is [27]:

$${}^{E}_{C}\mathbf{H} = {}^{\text{on-E}}\mathbf{K}({}^{E}_{C}\mathbf{R} + \frac{{}^{E}_{C}\mathbf{t}(\mathbf{n}_{C})^{\mathrm{T}}}{d_{C\to\pi}}){}^{C}\mathbf{K}^{-1} \tag{7}$$

$\mathbf{n}_{\mathbf{C}}$ is the normal unit vector of the see-through display expressed in the camera coordinate system, $d_{C\to\pi}$ is the distance from the camera center C to the display image plane

π, and ${}^{\text{on-E}}\mathbf{K}$ is the ideal on-axis intrinsic that is dictated by the display manufacturer's specifics such as width/height (w, h) and horizontal/vertical angle-of-view ($hAOV$, $vAOV$). Without losing generality, here the skew factor is ignored:

$$ {}^{\text{on-E}}\mathbf{K} = \begin{bmatrix} \frac{w}{2\tan(\frac{hAOV}{2})} & 0 & \frac{w}{2} \\ 0 & \frac{h}{2\tan(\frac{vAOV}{2})} & \frac{h}{2} \\ 0 & 0 & 1 \end{bmatrix} \tag{8} $$

By substituting (7) and (6) into (5), the following relation can be obtained after algebraic manipulations:

$$ \lambda \mathbf{i}^E = {}^{\text{on-E}}\mathbf{K}(\mathbb{I}_{3\times 3} + \frac{{}^E_C\mathbf{t}(\mathbf{n}_E)^\top}{d_{C\to\pi}}){}^E_C\mathbf{R}\begin{bmatrix} {}^C_W\mathbf{R} & {}^C_W\mathbf{t} \end{bmatrix}\mathbf{v}^W \tag{9} $$

which represents the perspective projection relation of the off-axis pinhole model of the OST display. The same relation in matrix form is:

$$ \lambda \mathbf{i}^E = \underbrace{{}^{\text{on-E}}\mathbf{K}\mathbf{H}}_{{}^{\text{off-E}}\mathbf{K},\text{intrinsic}} \underbrace{\begin{bmatrix} {}^E_W\mathbf{R} & {}^E_C\mathbf{R}^C_W\mathbf{t} \end{bmatrix}}_{\text{extrinsic}} \mathbf{v}^W \tag{10} $$

Since $\mathbf{n}_E = [0, 0, 1]^\top$, and $d_{C\to\pi} = d_{E\to\pi} - {}^E_C\mathbf{t}_z$, the homography matrix $\mathbf{H}$ has only 3 degrees of freedoms (DOFs) (i.e., the 3D translation ${}^E_C\mathbf{t}$):

$$ \mathbf{H} = \begin{bmatrix} 1 - \frac{{}^E_C\mathbf{t}_z}{d_{E\to\pi}} & 0 & \frac{{}^E_C\mathbf{t}_x}{d_{E\to\pi}} \\ 0 & 1 - \frac{{}^E_C\mathbf{t}_z}{d_{E\to\pi}} & \frac{{}^E_C\mathbf{t}_y}{d_{E\to\pi}} \\ 0 & 0 & 1 \end{bmatrix} \tag{11} $$

The product of ${}^{\text{on-E}}\mathbf{K}$ and $\mathbf{H}$ therefore characterises the off-axis intrinsic matrix of the OST display ${}^{\text{off-E}}\mathbf{K}$ at a generic viewpoint C. The 3×3 homography correction $\mathbf{H}$ encapsulates the shift and scaling effect due to a particular viewpoint position. It also accounts for the deviation of the real optical features of the see-through display from the ones provided by the specifications.

IV. CAMERA-BASED REDUCTION OF THE PARALLAX-RELATED MISALIGNMENT

In this section, we extend the algorithm introduced in Section III-C to a camera-based calibration routine that can be applied to any commercial OST HMD. We will show that our method does not require any user alignment or robust offline calibration for estimating the physical focal distance of the display.

A. RATIONALE

If the camera C represents the user's eye, (10) defines the location of a pixel $\mathbf{i}^E$ that properly aligns with the line-of-sight between the user's viewpoint and the 3D real-world point $\mathbf{v}^W$. If we consider the off-axis rendering camera (e.g., the left camera-screen system $L-S$ of a binocular headset) of a game engine, the rendered pixel is determined by the associated off-axis pinhole camera projection:

$$ \lambda \mathbf{i}^L = {}^L\mathbf{P}\begin{bmatrix} {}^L_W\mathbf{R} & {}^L_W\mathbf{t} \end{bmatrix}\mathbf{v}^W \tag{12} $$

where ${}^L\mathbf{P}$ is the projection matrix of the rendering engine. Before performing the camera-based calibration, ${}^L\mathbf{P}$ is not calibrated according to the observation viewpoint so the projected pixel $\mathbf{i}^L$ is not aligned with the target perceived by user.

To display the pixel $\mathbf{i}^L$ at the correct location, we need to ensure

$$ \lambda \mathbf{i}^L = \lambda \mathbf{i}^E \tag{13} $$

After adapting all the transformation matrices to the 4×4 convention, this equilibrium can be written as:

$$ {}^L\mathbf{P}\begin{bmatrix} {}^L_W\mathbf{R} & {}^L_W\mathbf{t} \\ 0 & 1 \end{bmatrix} = {}^{\text{on-E}}\mathbf{K}\mathbf{H}\begin{bmatrix} {}^E_W\mathbf{R} & {}^E_C\mathbf{R}^C_W\mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{14} $$

$$ {}^L\mathbf{P} = {}^{\text{on-E}}\mathbf{K}\mathbf{H}\begin{bmatrix} {}^E_W\mathbf{R} & {}^E_C\mathbf{R}^C_W\mathbf{t} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} {}^L_W\mathbf{R} & {}^L_W\mathbf{t} \\ 0 & 1 \end{bmatrix}^{-1} \tag{15} $$

This is the calibrated projection matrix of the rendering engine that can correctly render the virtual contents according to the viewpoint position.

To solve ${}^L\mathbf{P}$ for commercial headsets, we adapt the matrices in (15) to the OpenGL convention [13]. The ideal on-axis projection becomes:

$$ {}^{\text{on-E}}\mathbf{K} = \begin{bmatrix} \frac{1}{\tan\frac{hAOV}{2}} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan\frac{vAOV}{2}} & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{16} $$

where n and f define a rendering depth range from the near to far clipping plane. They are user-specified parameters dictated during application design.

The 3×3 planar homography in (11) is thus expanded to 4×4 by including the redundant z dimension:

$$ \mathbf{H} = \begin{bmatrix} 1 - \frac{{}^E_C\mathbf{t}_z}{d_{E\to\pi}} & 0 & \frac{{}^E_C\mathbf{t}_x}{d_{E\to\pi}} & 0 \\ 0 & 1 - \frac{{}^E_C\mathbf{t}_z}{d_{E\to\pi}} & \frac{{}^E_C\mathbf{t}_y}{d_{E\to\pi}} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17} $$

Since ${}^E_W\mathbf{R} = {}^L_W\mathbf{R}$ as it is dictated by the orientation of the display focal plane, the extrinsic part equals to:

$$ \begin{bmatrix} {}^E_W\mathbf{R} & {}^E_C\mathbf{R}^C_W\mathbf{t} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} {}^L_W\mathbf{R} & {}^L_W\mathbf{t} \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \mathbb{I}_{3\times 3} & {}^E_C\mathbf{R}^C_W\mathbf{t} - {}^L_W\mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{18} $$

${}^E_C\mathbf{R}^C_W\mathbf{t} - {}^L_W\mathbf{t}$ measures the distance between the viewpoint camera location and the off-axis rendering camera.

For commercial OST displays, the parameters in ${}^{\text{on-E}}\mathbf{K}$ (e.g., $vAOV$ and $hAOV$) are available from the manufacturer, but parameters such as the physical focal length $d_{E\to\pi}$ requires for additional and robust calibrations. Differently, according to (17), $\mathbf{H}$ does not depend on the absolute and exact value of ${}^E_C\mathbf{t}$ and $d_{E\to\pi}$, but rather on their ratio. Therefore, in the next section, we will show that the up-to-scale $d_{E\to\pi}$ and ${}^E_C\mathbf{t}$

TABLE 1. Parameters involved in camera-based method, with encapsulated matrices and acquisition methods.

Parameters	Definition	Matrix	Method
$hAOV$,$vAOV$, h, w	Display rendering properties	K	Manufacturer's specifics
$d_{E\to\pi}$	Arbitrarily chosen focal distance	H	Arbitrarily fixed
${}^{E}_{C}\mathbf{t}$	viewpoint shift scaled with $d_{E\to\pi}$	H	PnP calibration
${}^{E}_{C}\mathbf{R}$	Rotation between C and E	Extrinsic	PnP calibration
${}^{C}_{W}\mathbf{t}$	viewpoint location	Extrinsic	Stereo calibration, device tracking
${}^{L}_{W}\mathbf{t}$	device location	Extrinsic	Device tracking

can be easily calibrated by performing Perspective-n-Point (PnP)-based step that is much easier than the conventional DRC-like hardware calibrations and it still provides robust results. This makes our solution more implementable than state-of-the-art camera-based methods such as the DRC. For the extrinsic part, transformations such as ${}^{L}_{W}\mathbf{t}$ can be obtained from external tracking systems or internal self-tracking that are enabled in many commercial HMDs. Table 1 lists all the involved parameters and the way to obtain them.

B. OFFLINE CALIBRATION

1) CALIBRATION FOR HOMOGRAPHY MATRIX

Assuming an arbitrary focal distance of the display ($d_{E\to\pi}$), the correspondingly scaled physical pixel size α can be calculated according to the display vertical angle-of-view $vAOV$ and height h in pixels:

$$\alpha = \frac{h'}{h} = \frac{2d_{E\to\pi}\tan(\frac{vAOV}{2})}{h} \propto d_{E\to\pi} \tag{19}$$

where h' is the physical display height obtained considering the assumed physical focal distance $d_{E\to\pi}$.

Then, a chessboard pattern with known resolutions in pixels is displayed at the centre of the screen. The scaled physical size of the displayed grid can be calculated according to α. The projected grid pattern is then captured by a pre-calibrated viewpoint camera C. By solving a standard PnP problem [28], the transformation from C to S can be calculated. Notably, the rotational component ${}^{S}_{C}\mathbf{R}$ is accurate and independent of α, whereas the elements of the translation component are all proportional to α and thus to the arbitrary focal distance:

$${}^{S}_{C}\mathbf{t} \propto \alpha \propto d_{E\to\pi} \tag{20}$$

Since:

$${}^{E}_{C}\mathbf{t} = {}^{E}_{S}\mathbf{R}\,{}^{S}_{C}\mathbf{t} + {}^{E}_{S}\mathbf{t} = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}^{S}_{C}\mathbf{t} + [0,0,d_{E\to\pi}]^{T} \propto d_{E\to\pi} \tag{21}$$

${}^{E}_{C}\mathbf{t}/d_{E\to\pi}$ and thus the final value of $\mathbf{H}$ are not affected by the arbitrarily chosen value of $d_{E\to\pi}$. Such a simple PnP calibration is enough to robustly estimate the hardware and viewpoint-related parameters of the off-axis pinhole model of the OST display using our homography-based method.

2) EXTRINSIC-RELATED PARAMATERS

For the unknowns in the extrinsic matrix (i.e., ${}^{E}_{C}\mathbf{R}\,{}^{C}_{W}\mathbf{t} - {}^{L}_{W}\mathbf{t}$),

$${}^{E}_{C}\mathbf{R} = {}^{S}_{C}\mathbf{R} \tag{22}$$

which is given by the PnP calibration mentioned above.

$${}^{C}_{W}\mathbf{t} = {}^{C}_{T}\mathbf{R}\,{}^{T}_{W}\mathbf{t} + {}^{C}_{T}\mathbf{t} \tag{23}$$

where T is the coordinate of the target tracker (e.g., a world-facing camera on the headset for head-anchored tracking, or an external optical tracker for world-anchored tracking). ${}^{T}_{W}\mathbf{t}$ is given by the target tracking. As mentioned above, ${}^{L}_{W}\mathbf{t}$ can be obtained by the inside-out or outside-in device tracking. ${}^{C}_{T}\mathbf{R}$ and ${}^{C}_{T}\mathbf{t}$ (i.e., The relative pose between the viewpoint camera and tracker) are unknown and can be solved by a standard stereo-camera calibration.

V. OBJECT-BASED REDUCTION OF THE PARALLAX-RELATED MISALIGNMENT

Unlike many state-of-the-art methods, the object-based calibration proposed here does not model the camera-display system as a pinhole camera. Instead, it utilises the benefits of the 3D system representation inspired by the development of several commercial OST-HMDs.

A. RATIONALE

Instead of modifying the system's projection matrix to achieve the pixel correspondence in the 2D screen coordinates (as for the camera-based method and for traditional SPAAM/DRC methods), the location of the virtual target to be rendered ($\mathbf{t}$) is repositioned in 3D while the default projection matrix associated to the rendering camera is kept unchanged. As shown in Figure 4, a view gaze is defined as the ray starting from the viewpoint $\mathbf{c}$ (i.e., the user's eye nodal point or viewpoint camera's optical centre) to the tracked point $\mathbf{t}$ in a common 3D coordinate W. The virtual display plane is modeled as a 3D surface S in W. The corresponding pixel displayed by the see-through headset $\mathbf{i}^{S}$ can be localised by the intersection between the gaze $\overrightarrow{\mathbf{ct}}$ and the modelled screen. To make the actual pixel displayed at such a location, the system is forced to render based on a modified 3D virtual target point $\mathbf{t}'$ instead of the real tracked location $\mathbf{t}$. Taking the left rendering camera (whose optical centre is $\mathbf{o}$) as an example, the corrected location $\mathbf{t}'$ can be deduced from the spatial relationship:

$$\mathbf{t}' = |\overrightarrow{\mathbf{ot}}|\frac{\overrightarrow{\mathbf{oi}^{S}}}{|\overrightarrow{\mathbf{oi}^{S}}|} + \mathbf{o} \tag{24}$$

In practice, for a volumetric target, the viewpoint shift causes only the positional (i.e., parallax) but not orientation change on the rendered virtual object. The object-based correction only needs to be applied to an arbitrary point of the

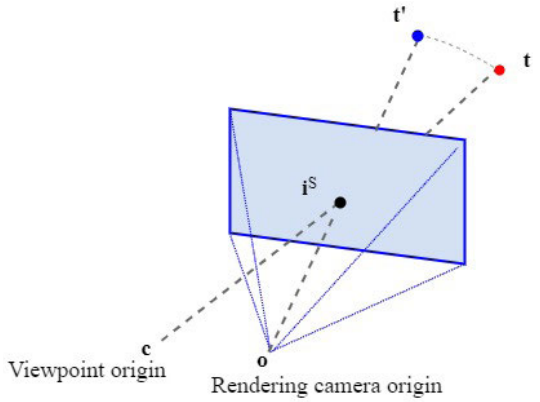


FIGURE 4. Reposition of the rendered virtual target to t′, according to the constructed 3D gaze and tracked target t. This is to ensure the gaze intersects the screen at the correct pixel $\mathbf{i}^S$.

target. The whole object can be rigidly translated according to the modified position of the chosen point to eliminate the effect of the viewpoint shift.

B. OFFLINE CALIBRATION

To determine the corresponding pixel $\mathbf{i}^S$ in 3D, both the position of the viewpoint $\mathbf{c}$ and the virtual display in the world W must be determined.

1) CALIBRATION FOR VIEWPOINT LOCATION

The offline calibration can be based on either a viewpoint camera or an actual eye position. If a viewpoint camera is used, the optical centre $\mathbf{c}$ can be localised by (23) and it requires device tracking and stereo calibration. If $\mathbf{c}$ is the eye nodal point, the viewpoint position can be retrieved by eye-tracking cameras as done in [21].

2) 3D DISPLAY REPRESENTATION

Because of the spherical aberrations produced by the optics of the HMD systems, the surface of the virtual display may resemble a curved surface more than a plane. This is especially true for systems with bulky aspheric reflective mirrors (e.g., Meta Vision) or freeform surface prisms. The shape of the curved surface should then be reconstructed in 3D by a full DRC routine [24] to model how the distance of the image plane of the display (i.e., the physical focal length) varies as the viewpoint moves away from the center of the eye-box.

Modern commercial OST HMDs (e.g., Microsoft HoloLens and the Magic Leap) use planar diffractive waveguides to tradeoff among the form factor, optical character, and mass production process [29]. These systems feature less optical aberration and the surface can thus be reasonably approximated by a flat plane, particularly for the viewpoint positions close to the center of the display eye-box. In these displays, a DRC-like procedure, to determine the almost constant focal length of the display is not strictly required and, instead, it could result in a too complex and error-prone procedure. Here, the focal length could be roughly estimated either by using a manually focused camera that is tuned to focus on the display [19], or by directly using the manufacturer's specifics.

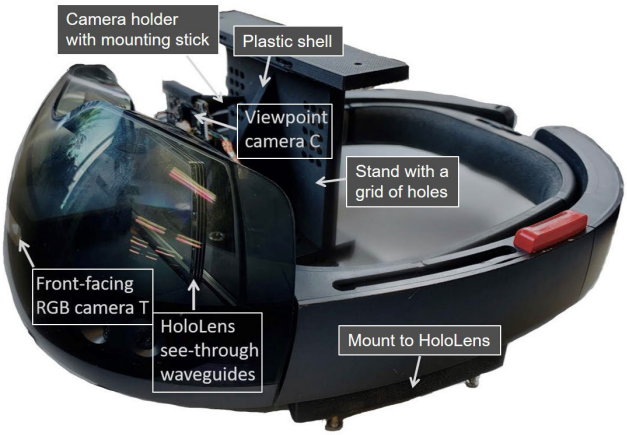


FIGURE 5. System arrangement (flipped upside down for convenience). HoloLens and viewpoint camera are rigidly fixed together by a plastic shell. The viewpoint camera can be translated within the eye box.

VI. ALGORITHM IMPLEMENTATION

A. HARDWARE DETAILS

We tested the proposed two methods on Microsoft HoloLens (1st generation, Microsoft Inc., Washington, U.S.), one of the most representative commercial OST HMDs currently available on the market [30]. The HoloLens has gained tremendous attention among other similar devices also for its potential use across highly complex applications in healthcare [15], [31]–[33] and the industrial field [7], [34]. The visor features an efficient self-tracking mechanism relying on on-board optical and inertial sensors, and a proprietary Simultaneous Localization and Mapping (SLAM) algorithm [35] for self-tracking. As for the optical sensors, the device includes four grayscale cameras, a time-of-flight (ToF) depth-sensing camera, and a world-facing colour camera that allows the user to record augmented videos and pictures (although they are not perfectly aligned with the user's line-of-sight). The HoloLens has two 720p, HD 16:9 light engines that render and display virtual contents via a pair of see-through waveguides [36]. The fixed focal distance d is around 2 m.

We implemented both methods based on a viewpoint camera C, a consumer-level HD webcam Creative Live! Cam Sync (Creative Technology Ltd., Jurong East, Singapore). The camera has a resolution of 1280×720 and an average angular resolution of 2.83 arcmin/pixel. As shown in Figure 5, a 3D printed plastic shell was used to rigidly house the visor and the camera C. The shell has a grid of holes to match with the holder of the camera. The spacing between holes is 5 mm. The holder can be translated on the shell and the camera C can be translated on the holder as well.

B. OFFLINE CALIBRATION

An overview of involved coordinates is shown in Figure 6. During the offline calibration, camera C was positioned and

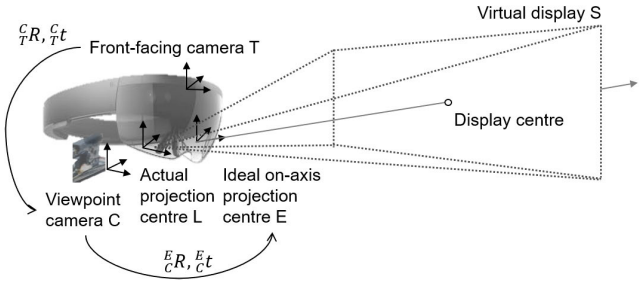


FIGURE 6. Spatial relationship between the involved coordinates and the transformations to be calibrated offline.

calibrated at several locations that correspond to the different interpupillary distance (IPD) [37]. This is to investigate whether the calibration quality is affected by the viewpoint locations. Considering the size of the HoloLens eye box and the mean human IPD of 64 mm, 6 IPDs were tested: 50, 55, 60, 65, 70 and 75 mm. At each location, the following calibrations were carried out:

1) INTRINSIC CALIBRATION FOR THE TRACKING CAMERA *T* AND VIEWPOINT CAMERA *C*

This was performed with a conventional calibration routine [38]. A planar pattern (i.e., OpenCV checkerboard) with known grid size was captured from multiple views by the camera to be calibrated. The MATLAB camera calibration toolbox (R2019b MathWorks Inc., Natick, MA, USA) was used. The toolbox automatically detects the grid corners of captured patterns and calculates the intrinsic camera parameters by optimising the reprojection residuals.

2) CALIBRATION FOR THE RELATIVE ROTATION ${}^{E}_{C}\mathbf{R}$ AND THE UP-TO-SCALE VIEWPOINT SHIFT ${}^{E}_{C}\mathbf{t}$ ACCORDING TO AN ASSUMED FOCAL DISTANCE $d_{E\to\pi}$

This is only required for the camera-based method. A virtual 6×5 chessboard pattern of 800×600 pixels was displayed at the centre of the see-through display (Figure 7). The OST display was physically occluded to prevent background distraction. The grid size was calculated according to the arbitrarily chosen focal length $d_{E\to\pi}$. The virtual pattern was captured by camera C and processed by the MATLAB corner detection algorithm. The relative pose between E and C was calculated by solving PnP correspondence between 3D grids and detected 2D corners (as explained in Section IV-B1).

3) CALIBRATION FOR THE RELATIVE POSE BETWEEN *T* AND *C*

A printed planar 6×5 chessboard pattern with a grid size of 10 mm was captured simultaneously by both the viewpoint camera C and the tracking camera T. The relative pose between C and T was rigidly fixed. The camera system was re-orientated relative to the pattern and more than 20 pairs of multi-view images were collected. The OpenCV library [39] was used to detect grid corners and optimise the relative pose $({}^{C}_{T}\mathbf{R}, {}^{C}_{T}\mathbf{t})$ by minimising the overall reprojection error using all pairs of images. During the optimisation, the camera intrinsic

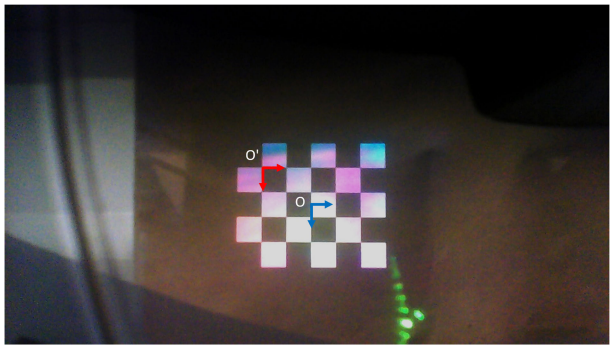


FIGURE 7. The virtual chessboard captured by the viewpoint camera C. Pattern centre O is displayed at the centre of the display. Note that in MATLAB the local image space originates from the corner O'. A translation is applied towards the MATLAB result to calculate ${}^{S}_{C}\mathbf{t}$.

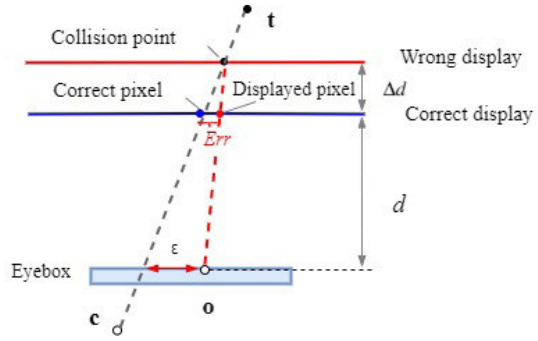


FIGURE 8. An aerial view of the camera-display system for the sensitivity analysis.

of T and C was fixed to the pre-calibrated values. The final stereo calibration error was 0.42 pixels.

4) REPRESENTATION FOR DISPLAY

As mentioned above, the HoloLens display can be reasonably modelled as a flat plane. Only the focal distance needs to be estimated. The display accuracy is, actually, not sensitive to d (i.e., the camera to display distance $d_{E\to\pi}$). As shown in Figure 8, an inaccurate focal distance estimation of Δd leads to the pixel misalignment of:

$$Err = \frac{\varepsilon}{1 + \frac{d}{\Delta d}} \tag{25}$$

In practice, since $d \gg \Delta d$ and the viewpoint shift ε is usually less than 10 mm, a 10% error in Δd will lead to an error of 0.9 mm in the display plane of HoloLens. The misalignment is even less noticeable from the perspective of a viewpoint camera C. As the HoloLens rough focal distance is known to be 2 m, we used it directly without detailed calibration for convenience.

VII. TEST AND RESULTS

A. TEST DESIGN

The accuracy of OST-HMD calibration cannot be objectively evaluated as we do not have direct access to the augmented image formed on the user's retina [11]. For alignment-based calibration methods, the tracked target can be re-projected

to the display using the calibrated projection matrix and the re-projected pixels are then compared to ''ground-truth'' pixels that are manually annotated on the virtual display plane by users [40], [41] or by dividing the calibration data sets into training and evaluation blocks [19]. Errors assessed by such methods are also affected by the user's interaction. Alternatively, some researchers have used a camera as eye replacement to compare the misalignment between the rendered virtual pixels and recorded target in the camera image plane [42], [43]. Some novel assessments that are unique to their calibration methods have also been reported. For example, eye positions are decomposed from calibration results and compared with measured eye positions [19].

Similar to the reported assessment in [42], [43], we used the camera-based evaluation as it is less user-biased by directly recording ''eye captures''. The target object is a 5×3 flat ChArUco board [44] with a uniform grid size of 38 mm. The locatable world-facing camera of the HoloLens was used as the target tracker T in our calibration tests. Scenes were recorded by T in real-time and processed by the OpenCV ChArUco board detection algorithm. The target pose was solved in the local tracker coordinate and further transformed into a global world coordinate W. The board was placed within arm's reach to simulate the near-field augmentation for manual tasks.

A Windows Mixed Reality application was developed in Unity3D for performance tests. As shown in Figure 9, two buttons were designed to switch on and off the two proposed calibration methods. The real scene recorded by the tracking camera T was displayed on a preview quadrilateral so that the user can ensure the target is within the tracking field. A corresponding virtual grid (with the same dimensions and size as the real ChArUco board) was rendered according to the tracked target pose. A double-tap gesture can turn the virtual rendering on/off.

We evaluated the overlay consistency between the rendered virtual grid and the perceived ChArUco board on the image plane of the viewpoint camera C. The overlay error e_{pixel} is defined as the Euclidean distance between a corresponding pair of pixels (p_i, q_i) that separately belongs to the virtual and real object:

$$e_{pixel} = \frac{\sum_{i=1}^{N} ||p_i - q_i||}{N} \tag{26}$$

where N is the number of sampled points for the evaluation. Because the error in pixels depends on the hardware resolution and the distance where the target board is placed, the overlay accuracy was also reported in terms of the visual angles in arcmin e_α (device and depth independent) and the physical distances in mm e_{mm} (device independent but depth dependent) according to:

$$e_\alpha = 2 \arctan \frac{e_{pixel}}{2f_C} \tag{27}$$

$$e_{mm} = \frac{d_t}{f_C} e_{pixel} \tag{28}$$

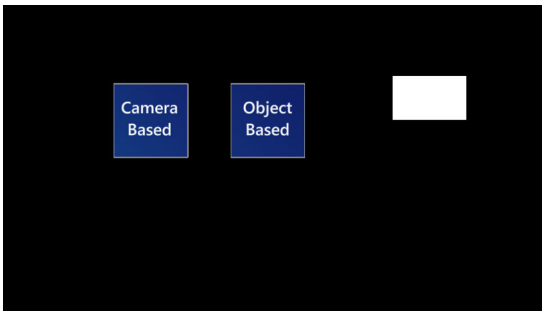


FIGURE 9. The Unity3D scene of the designed application. To avoid interruption for observation, buttons are world-locked so that they can be left out of the viewing field by moving the headset.

where d_t is the distance from C to the target board. f_C is the focal length of camera C in pixels.

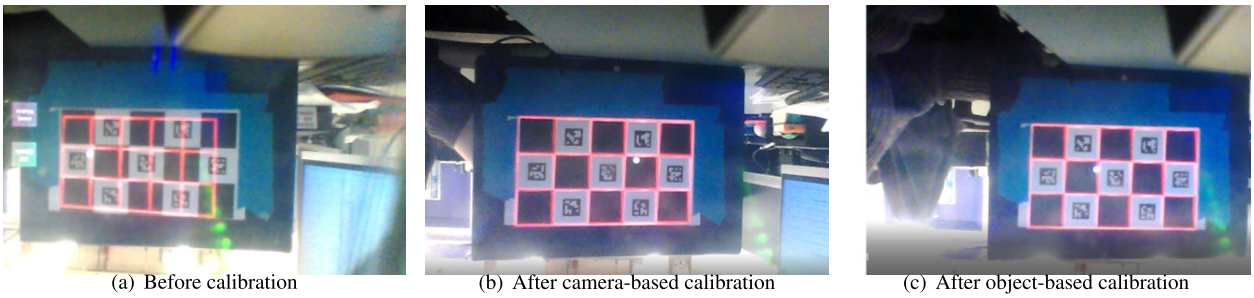
10 pairs of images were captured by C right after the calibration was done at each location. The target ChArUco board was randomly positioned and tilted to cover the image plane of viewpoint camera (Figure 10). For the identification of the correspondence between p_i and q_i, 8 inner corners of the ChArUco board are used as the sampled points for evaluation. During each pair of captures, the board was first photoed with the virtual rendering switched off (Figure 11(a)). Without moving the board or the camera, the correspondingly rendered grid was captured with the see-through waveguides occluded to avoid the distraction from the real-world background (Figure 11(b)). Two images were processed separately by a semi-automatic corner detection programme: for the captured ChArUco board, corners were automatically localised using the OpenCV corner detection algorithm. For the captured virtual grid, corners were first segmented based on the OpenCV contour detection and then refined by the user.

B. RESULTS AND ANALYSIS

The overlay misalignment is reported in means and standard deviations (Table 2). The physical display misalignment in the image plane of C is 1-1.5 mm (or 6-9 arcmin) for our camera-based method and 1.5-2 mm (or 9-12 arcmin) for our object-based method. The higher error and variance of the object-based method are expected since the camera-based calibration is tracking-independent whereas the object-based method directly relies on the tracked target pose. As the head-anchored tracking system is used in our experiment, the tracked target position can drift due to the ''accumulative errors among sensors'' [11]. Also, the asynchronism between the self-tracking and display refresh could lead to a jiggly display.

1) HORIZONTAL ERROR DISTRIBUTION

The misalignment is plotted with corresponding IPD values in Figure 12. Compared to the camera-based method, the object-based method has better consistency across different IPD values. In fact, in the original paper of the adopted camera-based method, the mean error increases two-fold



(a) Before calibration (b) After camera-based calibration (c) After object-based calibration

FIGURE 10. The target ChArUco board augmented with a virtual grid captured by the viewpoint camera. During our tests the target board was randomly moved to cover the whole image. Two methods can effectively align the scene with its virtual counterpart from the perspective of camera *C*.

(a) Captured real ChArUco board with virtual contents hided (b) Captured virtual grid with see-through waveguide being occluded

FIGURE 11. A pair of images captured by the viewpoint camera at the same calibrated location.

TABLE 2. The quantitative overlay misalignment of the proposed calibration methods with the viewpoint camera positioned in different IPD locations. The standard deviation (sd) is calculated using the data from all corners under variant target positions and poses.

		e_{pixel} (pixel)		e_{α} (arcmin)		e_{mm} (mm)	
IPD (mm)	Method	mean	sd	mean	sd	mean	sd
50	Camera-based	2.49	0.32	7.76	1.04	1.24	0.15
	Object-based	3.88	0.34	12.10	1.07	1.94	0.17
55	Camera-based	2.21	0.23	6.93	0.71	1.11	0.12
	Object-based	3.52	0.31	11.00	0.98	1.76	0.16
60	Camera-based	2.12	0.41	6.62	1.28	1.06	0.21
	Object-based	3.13	0.64	9.23	1.99	1.56	0.32
65	Camera-based	1.93	0.41	6.02	1.27	0.96	0.21
	Object-based	3.16	0.51	9.89	1.58	1.58	0.26
70	Camera-based	2.61	0.38	8.14	1.21	1.30	0.21
	Object-based	3.28	0.41	10.24	1.27	1.64	0.21
75	Camera-based	3.04	0.52	9.51	1.63	1.52	0.26
	Object-based	3.25	0.63	10.14	1.95	1.62	0.32

when the viewpoint camera translates horizontally [26]. In our tests, the camera-based method shows optimal performance with an IPD between 60-70 mm which corresponds to the positions around the eye box centre. This may be because the image distortion is not considered by the camera-based method but by the object-based method.

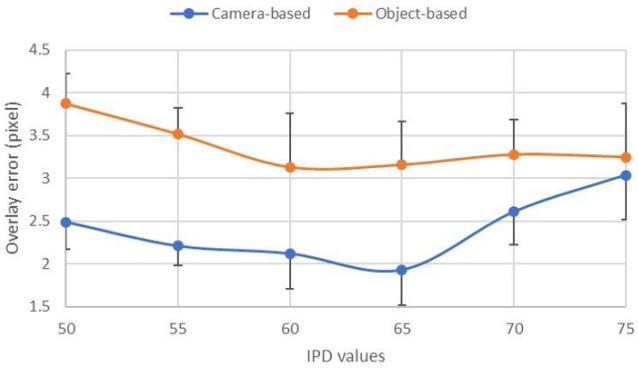


FIGURE 12. The variation of overlay accuracy with different IPD values. Error bars indicate the variance of misalignment at each location. For convenience only half the error bars are drawn as they are symmetric.

Since image distortions are non-linear and are larger around image peripherals, we, therefore, hypothesize that the AR overlay accuracy degrades as the viewpoint camera moves away from the eye box centre under the camera-based calibration.

2) QUANTITATIVE COMPARISON TO BENCHMARK METHODS

We compare the accuracy of our calibration methods with two benchmark calibrations:

1) Manufacturer's default calibration + IPD correction: this is the default HoloLens calibration method embedded in the device. Starting from manufacturer's calibrated intrinsic, the projection can be corrected in horizontal direction by IPD correction for different users. In practice, the user's IPD is calculated by an official calibration application on HoloLens: for each eye display, users are instructed to manually align a finger with six virtual markers displayed to them. For a viewpoint camera, the equivalent IPD was calculated according to the shell design: it equals to the distance between a pair of symmetric holding holes (Figure 5) plus two times the distance from the centre of the viewpoint camera to the mounting stick. The calculated IPD was manually set via the HoloLens portal.

TABLE 3. Comparison of our calibration methods with two benchmark calibrations.

	Number of user alignments	Accuracy e_{mm} (mm) mean	sd
Camera-based	0	1.20	0.20
Object-based	0	1.68	0.24
IPD correction	n*	6.97	0.31
Black-box SPAAM [17]	20	1.45	0.68

* n is the number of alignments for IPD measurements. It depends on the method used. For example, n=12 for the default HoloLens calibration, and n=0 if IPD is directly measured by a pupilometer.

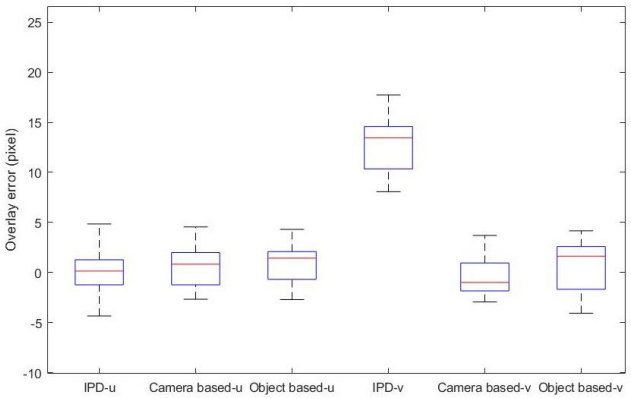


FIGURE 13. The horizontal and vertical error distribution of IPD correction method compared to our two methods.

2) A SPAAM-based black-box calibration proposed by Azimi *et al.*: the method corrects the transformation from a tracked 3D object to its representation in the virtual space based on SPAAM. 20 user alignments were collected for the calculation of a 4×4 perspective correction matrix [17]. The results were obtained also with the head-anchored tracking system. As our evaluation was done in the image plane of viewpoint camera and the depth information was lost, only the 2D components of their results were compared to our results.

Table 3 compares the physical calibration errors e_{mm} among four methods. IPD-correction gives the highest error as expected. This is because the IPD correction only compensates for an horizontal viewpoint shift. The vertical parallax still exists and it contributes most to the overall error (Figure 13). Our calibration methods achieve similar display accuracy (i.e., similar mean level) with the SPAAM-based calibration by Azimi *et al.* [17] but our methods are less variant (i.e., smaller standard deviation). This could be due to the fact that our methods are independent of user-alignments and are thus more objective.

VIII. DISCUSSION

A. ONLINE UPDATE SCHEME

The offline calibration is carried out for a generic viewpoint within the eye-box of the OST display. As suggested by Owen *et al.*, if the target is near the virtual display plane, the phase-two update is not necessary as the largest expected registration error is usually a few pixels that are hardly noticeable [24]. For the near field augmentation experience (e.g., AR-assisted manual tasks) where the target is not close to the focal plane, the virtual-to-real misalignment is more sensitive to the viewpoint shift and the phase-two refinement is necessary.

TABLE 4. Options for online update from a calibrated camera position to an actual user's eye.

Option	Update **U**	Explanation	Min. Aligns
0	Identity	No update	0
1	$\begin{bmatrix} 1 & 0 & t_x & 0 \\ 0 & 1 & t_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	t_x, t_y: 2D translation in image (neglect difference in depth)	1
2	$\begin{bmatrix} \alpha & 0 & t_x & 0 \\ 0 & \alpha & t_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	α: scaling; t_x, t_y: 2D translation in image	2
3	$\begin{bmatrix} \alpha & 0 & 0 & t_1 \\ 0 & \alpha & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	α: scaling; t_1, t_2, t_3: 3D translation in space	3

As suggested by Genc *et al.*, starting from an offline calibration at a viewpoint, pixels ($\mathbf{p}_i$) perceived from the old viewpoint (i.e., calibration camera C) can be transformed into the points ($\mathbf{p}_{i+1}$) perceived from a new viewpoint (i.e., user's eye) by an update matrix **U** [18]:

$$\mathbf{p}_{i+1} = \mathbf{U}\mathbf{p}_i \tag{29}$$

Several update models for **U** can be applied [18], [24]. As shown in Table 4 (in OpenGL convention), in most scenarios, the shift of viewpoint mainly causes a 2D warp in the image plane. Hence, the update can be modelled by linear translations and scaling (Option 1-2). In this case, the matrix **U** mainly corrects the intrinsic difference (i.e., **H**) caused by the different viewpoints. Alternatively, if the translation in space is dominant relative to the translation in the image, the extrinsic needs to be compensated by the matrix **U** (Option 3). In practice, Users can choose a specific update scheme depending on the applications.

For the object-based method, the calibration can also be updated automatically using the 3D nodal location tracked by an eye tracker. A similar implementation can be found in [19]. However, because of the limited tracking accuracy of eye cameras, the automatic update usually achieves higher calibration errors compared to the SPAAM-based update [19]. Besides, unless the eye-trackers are integrated into the headset (e.g., as in HoloLens 2), extra pose calibration between the trackers and the system is required. Therefore, compared to the eye-tracking based update, the update based on a few user alignments is more applicable and accurate with a reliable starting point provided by our offline calibrations.

TABLE 5. The comparison between our methods (in bold) with state-of-the-art methods for a starting viewpoint calibration. Methods are organised with descending calibration complexity.

Method	User-alignment	Projection-related info	Hardware-related info
DRC [24]	-	Off-axis intrinsic	Physical focal distance, HMD apex, sphere aberration parameters, etc.
INDICA (full setup) [19]	-	-	Physical focal distance, physical display pose, physical pixel size
Object-based	**-**	**-**	**Physical focal distance**
Camera-based	**-**	AOV, w, h	**-**
Black-box SPAAM [17]	YES	Off-axis intrinsic	-
SPAAM [22]	YES	-	-

B. COMPARISON BETWEEN TWO METHODS

Both methods can effectively correct the parallax-related registration error without involving user alignments during the first stage calibration, and thus, provide an accurate and objective starting point for the prior-to-use update phase. The two methods are different and complementary in some aspects. We here compare the two methods and provide some recommendations for choosing the method according to the application:

First, for the camera-based method, because of the involved PnP calibration, C can only be a viewpoint camera. By contrast, the object-based calibration can also be applied to actual human eyes.

Second, as indicated by their names, the camera-based method compensates the parallax-related registration error by correcting the projection matrix of the rendering camera. Once the display has been calibrated for a viewpoint, the augmentation for all tracked targets should be correctly aligned (i.e., independent on the target tracking). By contrast, the object-based method corrects the misalignment for individual targets based on their tracked locations. Therefore, if a large number of targets need to be virtually augmented, the camera-based method is potentially more efficient. Also if the tracking system is not accurate, the object-based method is not recommended.

The object-based calibration shows a "black-box" nature: given an input as the tracked 3D target location t, the algorithm outputs the modified 3D virtual object location t' for the viewpoint-dependent alignment. Hardware details and rendering procedures are sealed inside the "box".

The image distortion can be considered as an offset between the projected pixel and displayed pixel [43]. As shown in Figure 14, by default, the rendering camera can display a pixel at the correct location for a target $\mathbf{t}$ by taking image distortions into account (red arrow). Ideally, if users knew the exact value of optical distortion parameters, the undistortion mapping could be applied manually. However, these values are normally not provided to users. This is not a problem for the object-based calibration method, since the distortion correction is bypassed because that the method is "result driven" (i.e., get the corresponding pixel first, then find the required modifications). It is the rendering pipeline that is doing the distortion correction (Figure 14). Given a modified target location $\mathbf{t}'$, the display would automatically adjust according to the embedded distortion matrix that is invisible to users (blue arrow).

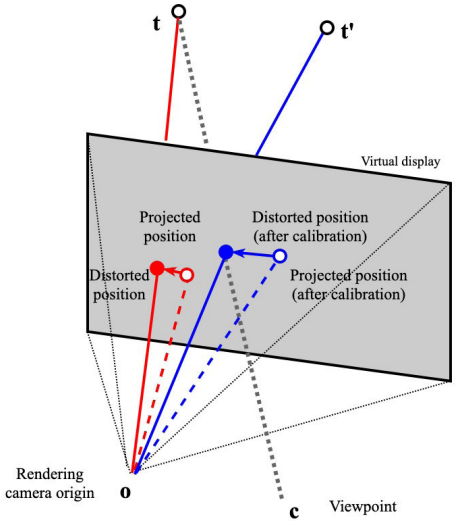


FIGURE 14. Object-based calibration with image distortions explicitly shown.

C. COMPARISON WITH STATE-OF-THE-ART METHODS

Table 5 shows the comparison between our methods and some well-known state-of-the-art solutions. Our camera-based calibration resembles the offline DRC method but the number of display parameters that need to be estimate is much less and therefore it is more implementable and less prone to numerical instability [38]. This is mainly because the adopted homography model allows us to skip the calibration of the exact physical display parameters (e.g., the distance from the virtual display plane to the calibration viewpoint). Our object-based calibration resembles the black-box SPAAM proposed in [17]: our method also corrects the parallax-related virtual-to-real registration error in 3D space and seal the rendering process in a "black-box" manner. However, the proposed object-based method requires less prior knowledge about the rendering properties and it is completely alignment-free. If calibrated with a viewpoint camera, our method can provide a less subjective and more stable offline starting point. If calibrated with eye tracking cameras, our method can be used for online automatic parallax correction. Overall, the two proposed methods provide a

compromise between the accurate but tedious DRC method and the straightforward but subjective SPAAM.

It is worth mentioning the main differences between our proposed camera-based calibration and the homography-based calibration proposed in [26]. First, although the model is the same, the original work did not fully investigate the benefit brought by the up-to-scale relationship in simplifying the calibration procedure. Second, in their implementation on a customised headset, the projection centre of rendering engine was made free to move to the exact location of calibrated viewpoint (i.e., $C = L$ by our expression), whereas for us, C and L are not overlaid since the off-axis rendering centre of commercial headsets is not physically controllable. Last but not least, our implementation was based on a universal game engine Unity3D.

IX. CONCLUSION AND FUTURE WORK

In this article, we present two alignment-free offline calibration methods that effectively correct the parallax-related virtual-to-real registration error for commercial OST-HMDs. Implementation and validation have been carried out on the Microsoft HoloLens 1. Our calibration methods are robust (as they are user alignment-free) and easily implementable (as they simplify the hardware-related calibration to different extents). They provide a good compromise between usability and accuracy. The two proposed methods have been integrated into an Unity3D-based calibration framework so that they can be potentially applied to other Unity-supported commercial HMDs.

The robust calibration for a generic viewpoint provides a reliable starting point for the subsequent prior-to-use update. We provided a few options for such update phase that ensures different degrees of calibration accuracy. In the future, we will test these update schemes and evaluate the performance of the two methods based on user studies. Furthermore, future work will involve detailed investigations on the effect of optical distortons for the camera-based calibration method.

REFERENCES

[1] T. Sielhorst, M. Feuerstein, and N. Navab, "Advanced medical displays: A literature review of augmented reality," *J. Display Technol.*, vol. 4, no. 4, pp. 451–467, Dec. 2008.

[2] P. Vávra, J. Roman, P. Zonča, P. T. Ihn, M. Němec, J. Kumar, N. Habib, and A. El-Gendi, "Recent development of augmented reality in surgery: A review," *J. Healthcare Eng.*, vol. 2017, Aug. 2017, Art. no. 4574172.

[3] F. Cutolo, B. Fida, N. Cattari, and V. Ferrari, "Software framework for customized augmented reality headsets in medicine," *IEEE Access*, vol. 8, pp. 706–720, 2020.

[4] L. Chen, T. W. Day, W. Tang, and N. W. John, "Recent developments and future challenges in medical mixed reality," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Oct. 2017, pp. 123–135.

[5] M. Eckert, J. S. Volmerg, and C. M. Friedrich, "Augmented reality in medicine: Systematic and bibliographic review," *JMIR mHealth uHealth*, vol. 7, no. 4, Apr. 2019, Art. no. e10967.

[6] S. Lang, M. S. S. Dastagir Kota, D. Weigert, and F. Behrendt, "Mixed reality in production and logistics: Discussing the application potentials of microsoft HoloLensTM," *Procedia Comput. Sci.*, vol. 149, pp. 118–129, 2019.

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

[8] T. Fernández-Caramés, P. Fraga-Lamas, M. Suárez-Albela, and M. Vilar-Montesinos, "A fog computing and cloudlet based augmented reality system for the industry 4.0 shipyard," *Sensors*, vol. 18, no. 6, p. 1798, Jun. 2018.

[9] J. Grubert, Y. Itoh, K. Moser, and J. Edward Swan, "A survey of calibration methods for optical see-through head-mounted displays," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 9, pp. 2649–2662, Sep. 2018.

[10] A. Plopski, Y. Itoh, C. Nitschke, K. Kiyokawa, G. Klinker, and H. Takemura, "Corneal-imaging calibration for optical see-through head-mounted displays," *IEEE Trans. Vis. Comput. Graphics*, vol. 21, no. 4, pp. 481–490, Apr. 2015.

[11] N. Guo, T. Wang, B. Yang, L. Hu, H. Liu, and Y. Wang, "An online calibration method for microsoft HoloLens," *IEEE Access*, vol. 7, pp. 101795–101803, 2019.

[12] Y. Genc, M. Tuceryan, and N. Navab, "Practical solutions for calibration of optical see-through devices," in *Proc. Int. Symp. Mixed Augmented Reality*, Oct. 2002, p. 169.

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

[14] K. R. Moser and J. E. Swan, "Evaluation of user-centric optical see-through head-mounted display calibration using a leap motion controller," in *Proc. IEEE Symp. 3D User Interfaces (3DUI)*, Mar. 2016, pp. 159–167.

[15] L. Qian, A. Barthel, A. Johnson, G. Osgood, P. Kazanzides, N. Navab, and B. Fuerst, "Comparison of optical see-through head-mounted displays for surgical interventions with object-anchored 2D-display," *Int. J. Comput. Assist. Radiol. Surgery*, vol. 12, no. 6, pp. 901–910, Mar. 2017.

[16] J. A. Jones, D. Edewaard, R. A. Tyrrell, and L. F. Hodges, "A schematic eye for virtual environments," in *Proc. IEEE Symp. 3D User Interfaces (3DUI)*, Mar. 2016, pp. 221–230.

[17] E. Azimi, L. Qian, N. Navab, and P. Kazanzides, "Alignment of the virtual scene to the tracking space of a mixed reality head-mounted display," 2017, *arXiv:1703.05834*. [Online]. Available: http://arxiv.org/abs/1703.05834

[18] Y. Genc, M. Tuceryan, and N. Navab, "Practical solutions for calibration of optical see-through devices," in *Proc. Int. Symp. Mixed Augmented Reality*, Oct. 2002, pp. 169–175.

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

[20] U. Fontana, F. Cutolo, N. Cattari, and V. Ferrari, "Closed–loop calibration for optical see-through near eye display with infinity focus," in *Proc. IEEE Int. Symp. Mixed Augmented Reality Adjunct (ISMAR-Adjunct)*, Oct. 2018, pp. 51–56.

[21] Y. Itoh and G. Klinker, "Performance and sensitivity analysis of indica: Interaction-free display calibration for optical see-through head-mounted displays," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Sep. 2014, pp. 171–176.

[22] M. Tuceryan, Y. Genc, and N. Navab, "Single-point active alignment method (SPAAM) for optical see-through HMD calibration for augmented reality," *Presence, Teleoperators Virtual Environ.*, vol. 11, no. 3, pp. 259–276, 2002.

[23] H. Hua, C. Gao, and N. Ahuja, "Calibration of a head-mounted projective display for augmented reality systems," in *Proc. Int. Symp. Mixed Augmented Reality*, Oct. 2002, pp. 176–185.

[24] C. B. Owen, J. Zhou, A. Tang, and F. Xiao, "Display-relative calibration for optical see-through head-mounted displays," in *Proc. 3rd IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2004, pp. 70–78.

[25] D. Lanman and D. Luebke, "Near-eye light field displays," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–10, Nov. 2013.

[26] F. Cutolo, U. Fontana, N. Cattari, and V. Ferrari, "Off-line camera-based calibration for optical see-through head-mounted displays," *Appl. Sci.*, vol. 10, no. 1, p. 193, Dec. 2019.

[27] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[28] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[29] Y. Zhang and F. Fang, "Development of planar diffractive waveguides in optical see-through head-mounted displays," *Precis. Eng.*, vol. 60, pp. 482–496, Nov. 2019.

[30] S. Condino, M. Carbone, R. Piazza, M. Ferrari, and V. Ferrari, "Perceptual limits of optical see-through visors for augmented reality guidance of manual tasks," *IEEE Trans. Biomed. Eng.*, vol. 67, no. 2, pp. 411–419, Feb. 2020.
[31] Philip Pratt, Matthew Ives, Graham Lawton, Jonathan Simmons, Nasko Radev, Liana Spyropoulou, and Dimitri Amiras, "Through the HoloLens looking glass: Augmented reality for extremity reconstruction surgery using 3D vascular models with perforating vessels," *Eur. Radiol. Exp.*, vol. 2, no. 1, p. 2, 2018.
[32] V. García-Vázquez, F. von Haxthausen, S. Jäckle, C. Schumann, I. Kuhlemann, J. Bouchagiar, A.-C. Höfer, F. Matysiak, G. Hättmann, J. P. Goltz, M. Kleemann, F. Ernst, and M. Horn, "Navigation and visualisation with HoloLens in endovascular aortic repair," *Innov. Surgical Sci.*, vol. 3, no. 3, pp. 167–177, Oct. 2018.
[33] J. W. Meulstee, J. Nijsink, R. Schreurs, L. M. Verhamme, T. Xi, H. H. K. Delye, W. A. Borstlap, and T. J. J. Maal, "Toward holographic-guided surgery," *Surgical Innov.*, vol. 26, no. 1, pp. 86–94, Feb. 2019.
[34] G. Evans, J. Miller, M. I. Pena, A. MacAllister, and A. E. Winer, "Evaluating the microsoft HoloLens through an augmented reality assembly application," in *Degraded Environments: Sensing, Processing, and Display* (International Society for Optics and Photonics), vol. 10197, J. J. N. Sanders-Reed and J. T. J. Arthur, III., Eds. Bellingham, WA, USA: SPIE, 2017, pp. 282–297.
[35] P. Hübner, K. Clintworth, Q. Liu, M. Weinmann, and S. Wursthorn, "Evaluation of HoloLens tracking and depth sensing for indoor mapping applications," *Sensors*, vol. 20, no. 4, p. 1021, Feb. 2020.
[36] B. C. Kress and W. J. Cummings, "Optical architecture of HoloLens mixed reality headset," *Proc. SPIE*, vol. 10335, Jun. 2017, Art. no. 103350K.
[37] N. A. Dodgson, "Variation and extrema of human interpupillary distance," *Proc. SPIE*, vol. 5291, pp. 36–46, May 2004.
[38] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.
[39] *Open Source Computer Vision Library*, OpenCV, 2015.
[40] E. McGarrity, Y. Genc, M. Tuceryan, C. Owen, and N. Navab, "A new system for online quantitative evaluation of optical see-through augmentation," in *Proc. IEEE ACM Int. Symp. Augmented Reality*, Oct. 2001, pp. 157–166.
[41] A. Tang, J. Zhou, and C. Owen, "Evaluation of calibration procedures for optical see-through head-mounted displays," in *Proc. 2nd IEEE ACM Int. Symp. Mixed Augmented Reality*, Oct. 2003, pp. 161–168.
[42] N. Makibuchi, H. Kato, and A. Yoneyama, "Vision-based robust calibration for optical see-through head-mounted displays," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 2177–2181.
[43] S. Lee and H. Hua, "A robust camera-based method for optical distortion calibration of head-mounted displays," *J. Display Technol.*, vol. 11, no. 10, pp. 845–853, Oct. 2015.
[44] OpenCV. *Detection of Aruco Boards*. Accessed: Mar. 28, 2020. [Online]. Available: https://docs.opencv.org/3.4/db/da9/tutorial_aruco_board_detection.html

XUE HU received the B.Eng. degree in aircraft propulsion from Beihang University (BUAA), Beijing, China, in 2017, and the M.Sc. degree in advanced mechanical engineering from the Imperial College London, U.K., in 2018, where she is currently pursuing the Ph.D. degree with the Mechatronics in Medicine Laboratory, Department of Mechanical Engineering. Her research interests include augmented reality, computer-assisted orthopaedic surgery, and machine-vision applications.

FERDINANDO RODRIGUEZ Y BAENA (Member, IEEE) received the M.Eng. degree in mechatronics and manufacturing systems engineering from the King's College London, U.K., in 2000, and the Ph.D. degree in medical robotics from the Imperial College London, in 2004. He is currently a Professor in medical robotics with the Department of Mechanical Engineering, Imperial College London, where he leads the Mechatronics in Medicine Laboratory. His research interests include mechatronic systems for diagnostics, surgical training, and surgical intervention.

FABRIZIO CUTOLO (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical and computer engineering and the Ph.D. degree in translational medicine from the University of Pisa, Pisa, Italy, in 2006 and 2015. He is currently a Post-Graduate Research Associate with the Department of Information Engineering, University of Pisa. His research interests include developing and evaluating new mixed reality solutions for image-guided surgery and surgical simulation, machine-vision applications, visual perception, ubiquitous tracking, and human–machine interfaces for rehabilitation. He has been involved in several national and international research projects. He is also the WP Leader of the HORIZON2020 Project VOSTARS, Call ICT-29-2016.