IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 December 2023, accepted 16 January 2024, date of publication 22 January 2024, date of current version 30 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3356722

RESEARCH ARTICLE

Assessment of Multiple Fiducial Marker Trackers on Hololens 2

GABRIEL M. COSTA[1,2], MARCELO R. PETRY[1], JOÃO G. MARTINS[1,2], AND ANTÓNIO PAULO G. M. MOREIRA[1,2]

[1]INESC TEC—Institute for Systems and Computer Engineering Technology and Science, 4200-465 Porto, Portugal
[2]Department of Electrical and Computer Engineering, Faculdade de Engenharia da Universidade do Porto (FEUP), 4200-465 Porto, Portugal

Corresponding author: Marcelo R. Petry (marcelo.petry@inesctec.pt)

This work is co-financed by Component 5–Capitalization and Business Innovation, integrated in the Resilience Dimension of the Recovery and Resilience Plan within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021–2026, within project Hi_reV, with reference 64. This work is also financed by National Funds through the Portuguese funding agency, FCT—Fundação para a Ciência e a Tecnologia, within project LA/P/0063/2020. DOI 10.54499/LA/P/0063/2020 | https://doi.org/10.54499/LA/P/0063/2020. The work of G. M. Costa was supported by the Foundation for Science and Technology (FCT) under the grant 2022.10849.BD.

ABSTRACT Fiducial markers play a fundamental role in various fields in which precise localization and tracking are paramount. In Augmented Reality, they provide a known reference point in the physical world so that AR systems can accurately identify, track, and overlay virtual objects. This accuracy is essential for creating a seamless and immersive AR experience, particularly when prompted to cope with the sub-millimeter requirements of medical and industrial applications. This research article presents a comparative analysis of four fiducial marker tracking algorithms, aiming to assess and benchmark their accuracy and precision. The proposed methodology compares the pose estimated by four algorithms running on Hololens 2 with those provided by a highly accurate ground truth system. Each fiducial marker was positioned in 25 sampling points with different distances and orientations. The proposed evaluation method is not influenced by human error, relying only on a high-frequency and accurate motion tracking system as ground truth. This research shows that it is possible to track the fiducial markers with translation and rotation errors as low as 1.36 mm and 0.015 degrees using ArUco and Vuforia, respectively.

INDEX TERMS Microsoft HoloLens 2, marker tracking, Vuforia, ArUco, ARToolKit, QR Code, Optitrack, augmented reality.

I. INTRODUCTION

With the uprising of new and more powerful commercial Augmented Reality (AR) glasses, such as the Microsoft HoloLens 2 (HL2), augmented reality is one of many technologies that are being explored as a means to overcome the data visualization challenge of the current digitalization era and enable seamless human-computer and human-robot interactions [1], [2].

The development of mixed-reality applications often require fundamental research and a combination of different sensors. To lower the barrier and ease research involving AR and robotic applications, Microsoft made available a set of APIs and tools for HL2, known as Research Mode. Despite not being intended for end-user applications, this mode was designed for academic and industrial researchers to explore new ideas by providing them access to numerous sensors input data present on the HL2, such as a depth and RGB camera, four grayscale cameras, and an inertial measurement unit (IMU), which are used for the device's real-time Simultaneous Localization and Mapping (SLAM), hand tracking, and spatial mapping [3], [4]. With these tracking capabilities, the HL2 can overlay and maintain, with millimetric precision, stability, and lower latency, the position and orientation of a virtual 3D model over static and moving parts and relative to the user's physical surroundings [4], [5], [6].

The calibration of optical see-through (OST) head-mounted displays (HMD) is a challenge that hampers their use in medical and industrial settings because it often require precise coherence between real and virtual objects. While some studies have proposed different online and offline calibration methods to overcome this challenge on the tracking

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

device side [1], [7], [8], [9], others have attempted to tackle it from the marker perspective [10], [11].

Even though there is no consensus on whether or not overlaid, adjacent, animated, or static-rendered 3D models and instructions are better for handling different task types [12], [13], [14], the fact is that in any case, misalignment of the AR model regarding its physical counterpart can significantly affect the user performance depending on the complexity and nature of the task. Therefore, computer vision tracking algorithms are of utmost importance in mixed reality systems to ensure the accuracy and repeatability of the displayed AR content.

The correct placement of digital information is imperative to provide a better user experience in augmented reality systems and to create a seamless integration between the real and virtual worlds. Therefore, an essential requirement of AR applications is the tracking system. These can be classified into two main categories: the head-anchored tracking system (also called ''inside out'') and the world-anchored tracking system (also called ''outside in''). Providing a line of sight similar to the user's is one of the main advantages and characteristics of Head-anchored tracking systems. On the other hand, HMD-embedded hardware size, power consumption, and computational cost directly affect the tracking performance. In a world-anchored tracking system, the pose of the tracking system remains unchanged with respect to the world coordinate system. Without the constraints imposed by power, computational resources, and the type of technology used, a world-anchored tracking system can be highly accurate. On the negative side, this approach depends on external, often expensive and bulky hardware, which makes the system inflexible [1], [13]. Examples of world-anchored systems are Natural Point's Optitrack and Vicon Motion Systems.

Head-anchored tracking systems can be classified into marker-based and markerless systems. The main difference between the two technologies is that the former relies on tracking natural features from objects in the user's field of view, such as points, corners, and borders, and the latter relies on detecting fiducial markers, such as QR codes and checkerboard patterns. Although markerless systems do not require structured environments to work, they tend to be more computationally expensive and more affected by the lighting conditions. Marked-based systems, on the other hand, tend to be robust and easier to implement but may suffer in industrial environments due to dirt, corrosion, and occlusion [15], [16], [17].

One of the uses of tracking systems is to create a shared spatial anchor between different world frames. With a fixed and known position and rotation, it is possible to compensate for spatial localization drifts and as a reference when realizing spatial transformations between sensors, robot links, the real world, and the user's augmented environment. The correct spatial transformation between each coordinate system is crucial for calculating, displaying, and exchanging data between the human and robotic agent in human-robot collaborative applications [2], [18]. For example, in activities such as programming by demonstration [19], [20], AR can enable the user to visualize the robot's digital twin's workspace and intentions and anticipate the physical robot's movement, enhancing the user's perceived safety towards the robotic manipulator [21]. Moreover, several authors, such as [22], [23], and [24], employed spatial transformations to evaluate the precision of the tracking devices.

Asserting the accuracy and precision of fiducial marker trackers in head-mounted displays echoes over current research trends on augmented reality [25], such as assessing the viability of using these devices in a range of applications, such as medicine [23], architecture and civil engineering [26], [27], simultaneous localization and mapping [4], human-robot collaboration [28], [29], and many others.

The rest of the paper is structured in this way: section II offers a review of related literature on the evaluation of fiducial markers, with a focus on wearable AR devices. The methodology is outlined in section III, while section IV provides a detailed description of the experiment. The results obtained are presented and analyzed in section V and section VI, respectively. Finally, section VII concludes the paper.

II. RELATED WORKS

Recent works in this field have mainly focused on assessing the maker tracking accuracy on medical applications, particularly within surgical procedures. This interest primarily stems from the fact that OST-HMDs hold promise as a potential solution for the current challenge faced by surgeons, which entails diverting their attention from the surgical site when utilizing conventional navigation systems [7], [8], [23]. The underlying objective of these AR applications is to streamline the interpretation of 3D spatial data derived from MRI and CT scans by digitally superimposing a virtual model onto the patient's body. Consequently, the success in guiding surgeons during medical interventions is fundamentally bound to the application's accuracy and precision.

Quian et al. [30] presents an ARToolKit tracking application for aiding the first assistant of robotic laparoscopic surgeries. It derives from [1] previous efforts to calibrate the photo/video (PV) camera using a 3D-to-3D approach. The author evaluated the precision and accuracy of the medical application by assessing the 2D overlay deviation of the augmented information regarding its physical counterpart and concluded that tracking accuracy and imperfections on the display calibration were the central causes of misalignment. This work also introduced a priority rating approach for selecting the most accurate and lower latency transformation data available in runtime for calculating spatial transformation between the components of their application. While transformations derived from known kinematics data received the highest priority, those acquired from fiducial markers tracking received a medium priority, and the HMD self-localization received the lowest priority.

Similarly, [31] developed an AR application with the intent of aiding surgeons to perform ventriculostomy, more

specifically, to facilitate the placement of the catheter on the patient's brain. The application can either use an ARToolKit marker or Vuforia's 3D model tracker algorithm in combination with the Optitrack system to calculate the transformation matrix between the real and the augmented environment to render the virtual information over the catheter and phantom model used in the experiment. Using a caliper, the authors assessed the misalignment of the digital content over its physical counterpart.

Comparatively, but for neuronavigation, Frantz et al. [32] took advantage of the Vuforia SDK 3D model tracking feature to overlay digital information over a skull phantom model and assess the hologram spatial deviation within a surgical workspace by measuring the hologram drift from distances between 400 to 800 mm and angles ranging from $-90°$ to $+90°$ with a $45°$ step between each iteration. Graph paper and a ruler were the preferred instruments for measuring the fluctuation on the xy-plane and z-axis, respectively.

Using the ArUco tracking library, Doughty et al. [9] presents an initial design and assessment of an AR application intended to facilitate the visualization of MRI-derived virtual models in thoracotomy procedures. Their application employs an interactive runtime calibration approach that requires users to align a virtual 2D frame to a printed ArUco marker and collect correspondence points to compute the transformation matrix between the application and the real-world coordinate systems. The author assesses the precision and repeatability of the application through point-and-tracing experiments in four different display paradigms: on a monitor adjacent to a template (used as ground truth), with the augmented image displayed adjacent to the template using an HMD, with the augmented image displayed over the template with and without calibration using an HMD. Displaying the augmented image over the template with calibration was the most accurate approach.

The performance of the different HL2 built-in cameras was evaluated in Brand et al. [33] by comparing the reprojection error between mono and stereo cameras. Based on the calibrate-and-test procedure proposed in [1], Brand et al. presented the residual error for uncalibrated, isometric, affine, and perspective calibration models, where the affine and perspective models had the lowest residual error with a negligible difference between them. Thabit et al. [23] extended Brand's work by investigating the tracking performance of ArUco markers on the different cameras of HL2, as well as the effect of the marker size, distance from the camera, and camera resolution. While the HL2 stereo cameras were limited to a resolution of 640×480, the photo/video (PV) camera was tested at resolutions of 640×360, 760×428, 960×540, 1280×760, and 1920×1080 pixels. This study compared markers of three sizes (60 mm, 80 mm, and 100 mm), placed at 500 mm, 800 mm, 1100 mm, and 1400 mm apart from the HL2, with a reference position established at 300 mm. Their results indicated that lower camera resolutions had higher estimation errors but faster processing times. Notably, the stereo camera exhibited reduced localization error at 500 mm and 800 mm compared to the monocular camera with equivalent resolutions, but this advantage perished as the distance increased. Regarding the marker size, surprisingly, the results did not confirm the hypothesis that the larger the marker, the better the pose estimation at further distances.

Kunz et al. [34] tested HL2 stereo cameras with the passive retroreflective marker spheres typically used by current optical navigation systems. Gsaxner et al. [13] went further and implemented a similar algorithm with and without an Extended Kalman Filter and compared their performance with a monocular ArUco tracking. Other works, such as [35], used the HL2 Live-Capture camera for marker tracking. However, as pointed out by [36], an augmented object pose perceived by the user wearing the HMD might differ from those observed with the Live-Capture camera. This effect is even more visible when the object overlaid with the virtual content is not in the center of the image.

Kyaw et al. [37] developed an augmented reality application based on the Twinbuild software, intended to be an alternative for replacing plywood templates by superimposing 3D CAD models over wooden boards to aid carpenters on the fabrications of glulam beams. By positioning the QR Codes beforehand in CAD software, the author evaluated the registration accuracy throughout the complete extension of the glulam beam on the xy-plane using a digital caliper of 3 different setups, in different marker frequencies and with three different marker sizes. Their results indicated that increasing the number or size of the markers improves the registration accuracy to a certain threshold, after which the improvement is neglectable. The edge alignment was the most accurate, and the frame alignment was the most flexible and the fastest regarding setup time. Moreover, the authors concluded that the closer the beam features are to the markers, the higher the registration accuracy.

Table 1 summarizes the results of the works mentioned above. None of these works, however, compare the performance of multiple marker tracking algorithms on the HL2. Therefore, this work intends to benchmark different algorithms by assessing the accuracy and precision of their pose estimation relative to the HL2 using the Optitrack system as ground truth. From the algorithms available, Vuforia [38] is proprietary, while Mixed Reality ToolKit QR Code [39], ARToolKit [1] and ArUco [9] are open-source. This work only compares marker detection accuracy and precision in favorable and stable lighting and visibility conditions. Other studies such as [16], [40], and [41] have already evaluated/benchmarked the behavior of trackers in AR applications under multiple occlusion and lighting conditions.

III. METHODOLOGY

To benchmark the fiducial marker trackers on HL2, four algorithms were selected: Vuforia, Mixed Reality Toolkit QR Code, ARToolKit, and ArUco. Although several other algorithms are available for general-purpose processors and other operating systems, these are the only ones with

TABLE 1. Summary of the translation and orientation errors estimated in related works.

Reference	Tracking approach	Other info	Translation Error (mm)				Rotation Error (°)			
			mean	X	Y	Z	RMS	θX	θY	θZ
[9][a]	ArUco		1.02 ± 0.6							
[31]	ARToolkit			1.39 ± 0.57	1.17 ± 0.83	1.39 ± 0.88				
	Vuforia			0.96 ± 0.43	1.11 ± 0.27	1.44 ± 1.05				
[32][b]	Vuforia	-90°	0.83							
		-45°	1.46							
		0°	1.24							
		+45°	0.08							
		+90°	3.42							
		average	1.41 ± 1.08							
[30][c]	ARToolkit		4.27 ± 3.09							
[33][d]	ARToolKit	mono uncalib.	43.7 ± 10.3	10.2 ± 2.3	19.2 ± 1.2	37.1 ± 12.7				
		mono isometric	12.0 ± 6.7	-0.1 ± 2.4	-0.4 ± 3.7	-4.1 ± 12.5				
		mono affine	3.5 ± 2.0	-0.8 ± 1.3	-0.2 ± 1.3	0.0 ± 3.6				
		mono perspective	3.6 ± 1.8	-0.7 ± 1.3	-0.1 ± 1.2	0.0 ± 3.6				
		stereo uncalib.	227.7 ± 62.3	-28.0 ± 4.2	-223.1 ± 62.3	-4.0 ± 35.8				
		stereo isometric	40.3 ± 34.2	-2.8 ± 4.0	2.7 ± 5.8	-34.6 ± 39.3				
		stereo affine	20.2 ± 17.4	-2.1 ± 2.4	4.8 ± 3.8	-18.2 ± 18.2				
		stereo perspective	19.7 ± 13.9	-2.2 ± 2.9	4.6 ± 3.1	-16.9 ± 15.9				
[1][d]	ARToolKit	Head Anchored isometric		1.82 ± 1.08	2.05 ± 1.36	4.58 ± 3.31				
		Head Anchored affine		0.94 ± 0.74	0.83 ± 0.63	3.51 ± 2.67				
		Head Anchored perspective		1.00 ± 0.81	0.91 ± 0.68	3.55 ± 2.62				
		World Anchored isometric		3.64 ± 2.75	6.14 ± 3.88	3.43 ± 2.93				
		World Anchored affine		2.44 ± 1.98	2.98 ± 2.52	3.21 ± 3.01				
		World Anchored perspective		2.47 ± 2.04	3.01 ± 2.49	3.20 ± 3.01				
[23][e]	ArUco	500 mm		-1.5	-0.9	0.5		-0.5°	0.1°	-0.1°
		800 mm		-0.6	-0.2	-5.0		2.0°	-1.3°	0.1°
		1100 mm		0.4	1.7	-2.7		5.1°	-1.3°	-0.3°
		1400 mm		1.9	6.7	2.7		-7.5°	-1.3°	0.2°
[13][f]	ArUco		6.09 ± 1.15				6.73° ± 3.47°			
	stereo passive marker		8.07 ± 0.91				0.90° ± 0.18°			
	SCAAT-EKF stereo		1.70 ± 0.81				1.11° ± 0.39°			
[37][g]	QR Code	1 marker	10.86							
		2 markers	3.09							
		4 markers	1.17							
		6 markers	1.07							
		8 markers	0.97							

[a] 2D error between drawings, with the image registered directly onto the trace template using the HMD, after calibration
[b] Average Perceived drift measured between 400 mm and 800 mm away from the scanned object.
[c] 2D overlay accuracy
[d] Reprojection error
[e] Values from the experiment using the 80 mm marker and PV1920 resolution in a straight line, relative change in the marker pose.
[f] Root Mean Square (RMS)
[g] Average values using the edge alignment

implementations that currently support the Universal Windows Platform (UWP), which is necessary for the HL2. ARToolKit, ArUco, and Vuforia have implementations for late versions of Unity 2019.3 and above using the OpenXR Plugin, while the Mixed Reality Toolkit QR Code has an implementation for Unity 2020.3 and above using either the Windows XR Plugin or the OpenXR plugin. All of them can be used in new and existing Unity projects, though minor modifications may be needed.

Due to its sub-millimeter accuracy, the Optitrack system was chosen as ground truth to assess the performance of the marker tracking algorithms. Motive 3.0[1] was used to pair OptiTrack cameras, capture the motion of the rigid bodies defined by a set of retroreflective markers, and stream tracking data via Virtual-Reality Peripheral Network (VRPN). Although Optitrack can track active markers, only 19 mm diameter passive spheres were used in this experiment.

To accurately track the HL2, four markers were asymmetrically attached to the visor cover, creating a rigid body with no orientation ambiguity. Customized 200 mm x 300 mm acrylic boards were manufactured for tracking the fiducial marker pose. The boards contained a 100 mm x 100 mm printed pattern, selected according to the chosen tracking algorithm.

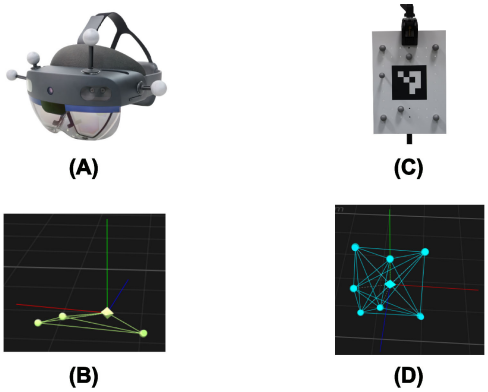


FIGURE 1. Hardware used in the experiments and their rigid body definitions: (A) Microsoft HoloLens 2; (B) Microsoft HoloLens 2 rigid body; (C) Marker board; (D) Marker board rigid body.

[1] https://optitrack.com/software/motive/

Additionally, a matrix of 4 mm holes spaced 25 mm between each other was drilled into the board by a CNC machine. This pattern allowed us to attach seven markers at known positions regarding the center of the board and consecutively from the center of the marker. Fig. 1 shows the HL2 and the board with the passive markers attached and their respective rigid bodies.

ROS 2 Humble was used as middleware for gathering and assessing the experiments' data. A TCP endpoint running as a ROS node[2] sends and receives messages from the Unity app. Rigid body's poses sent by Motive 3.0 were received by the ROS2 VRPN driver[3] and published as *geometry_msgs/PoseStamped* and *tf2*. All messages were recorded in rosbag2 files for post-processing.

Accurate timekeeping is achieved through the Network Time Protocol (NTP) client/server[4] deployed in the experiment devices. Time synchronization is essential to ensure ROS messages are being recorded in the correct order. After data collection, the message synchronization was performed using message filtering[5]. The policy used for synchronization was *ApproximateTime* to account for the different sampling rates between HL2 and Optitrack.

The system hardware configuration and communication are illustrated in Fig. 2 and depict the six main components:

- Cameras: Optitrack Prime^{x} 22 cameras.
- Rigid bodies: fiducial marker board and HL2.
- ROSMaster: laptop running Ubuntu 22.04, ROS 2 Humble, ROS TCP endpoint, NTP server, and the rosbag tool.
- Mocap: desktop running Windows 10, Motive 3.0, and the NTP client.
- Power over Ethernet (POE) switch: local switch to provide power to the Optitrack cameras and data interface between the cameras and Mocap.
- Wi-Fi Router: local network router to connect the ROSMaster and the Mocap via Ethernet cable, and the ROSMaster and the HL2 via Wi-Fi.

A. MARKER-CAMERA TRANSFORMATIONS

The Unity application sets its global reference frame in the location where it was started. Considering that the application does not restart and that the HL2 is kept static, it is safe to assume that the app's origin does not change during the experiments.

From the HL2 applications, it is possible to estimate the following transformation matrices:

- ${}^{U}T_{M}$ - Transformation from the fiducial marker (M) to the Unity origin (U).
- ${}^{U}T_{C}$ - Transformation from the HL2 PV camera (C) to the Unity origin (U).

Therefore, the transformation matrix from M to C can be computed through the following:

$$HLT_{ta} = {}^{C}T_{M} = ({}^{U}T_{C})^{-1} \cdot {}^{U}T_{M} \tag{1}$$

where *ta* represents the tracking algorithm.

B. GROUND TRUTH TRANSFORMATIONS

To properly approach the geometrical relations between the Optitrack and the rigid bodies, three assumptions were considered:

(I) Since the boards were manufactured in a CNC machine, the pattern of the fiducial markers printed on the boards and the position of the holes drilled to attach the retroreflective markers have a sub-millimeter error.
(II) Since the retroreflective markers were attached to the corners of a square centered at the fiducial marker pattern, the origin of the board and the fiducial marker are concentric. The rotation, though, varies according to each tracking algorithm.
(III) Since the retroreflective markers are rigidly attached to the HL2 front enclosure, the transformation between the markers and the HL2 PV camera is constant.

From the board design, it is possible to define the transformation ${}^{B}T_{M}$ from the Fiducial Marker (M) to the Board Rigid Body (B).

From the data received from Motive 3.0, it is possible to derive the following transformation matrices:

- ${}^{O}T_{H}$ - Transformation from the HL2 Rigid Body (H) to the Optitrack Origin (O).
- ${}^{O}T_{B}$ - Transformation from the Board Rigid Body (B) to the Optitrack's Origin (O).

The transformation between the HL2 rigid body and the HL2 PV Camera was estimated with a custom extrinsic calibration algorithm. We exploited the fact that retroreflective markers attached to the board have a spherical shape and can be simultaneously detected by Optitrack, which returns their 3D position, and the Hough Circle algorithm, which estimates the pixel-wise center of each circle. The pose of the PV camera was then determined by the SolvePnP algorithm: given the 3D coordinates of the six retroreflective markers on the board and their corresponding 2D coordinates in the HL2 PV Camera image, along with the camera's intrinsic parameters and distortion coefficients (A), SolvePnP[6] employs an iterative optimization process to estimate the rotation and translation vectors that best align the projected 3D points with the observed 2D image points:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \cdot {}^{C}T_{O} \cdot \begin{bmatrix} X_W \\ X_Y \\ X_Z \\ 1 \end{bmatrix} \tag{2}$$

This minimization of the reprojection error ensures an accurate estimation of the Optitrack coordinate system into the HL2 PV camera's frame (${}^{C}T_{O}$). From that, it is possible

[2]https://github.com/Unity-Technologies/ROS-TCP-Endpoint
[3]https://github.com/alvinsunyixiao/vrpn_mocap
[4]https://www.timesynctool.com/
[5]https://github.com/ros2/message_filters
[6]OpenCV Perpective-n-Point

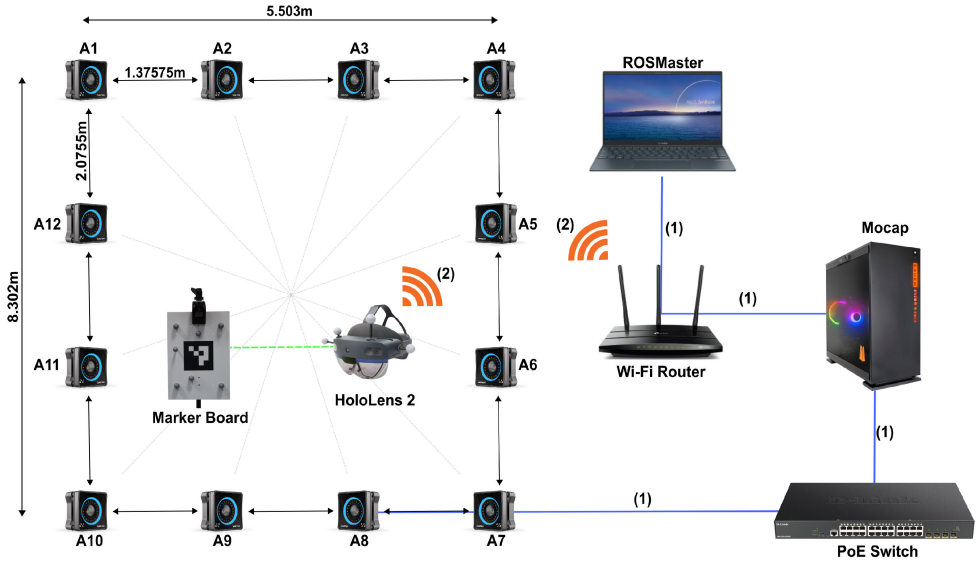


FIGURE 2. Experiment's architecture and setup: (1) Ethernet cable connection; (2) Wireless connection.

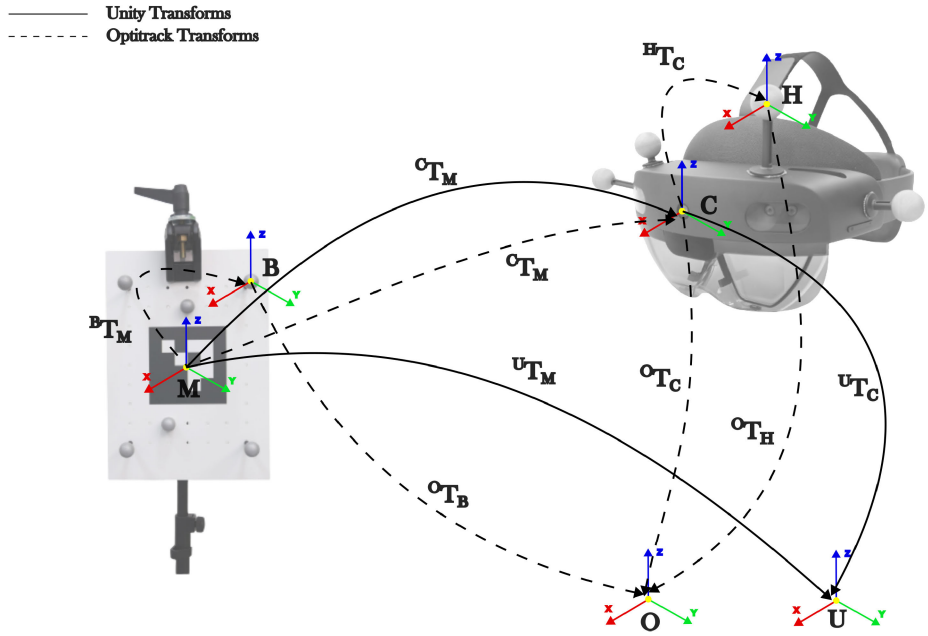


FIGURE 3. Definition of the spatial transformation between the microsoft Hololens 2 and the fiducial marker: Continuous lines indicate transformations provided by our unity app; Dotted lines indicate transformations provided by Optitrack. The board (B) and marker (M) referential frames are coincident. However, for better visualization and interpretation, they are represented separately.

to derive the transformation from (H) and (C):

$$^{C}T_{H} = {}^{C}T_{O} \cdot {}^{O}T_{H} \tag{3}$$

Finally, the Ground Truth (GT) is given by the transformation matrix from (M) to (C):

$$GT = {}^{C}T_{M}\ {}^{C}T_{H} \cdot ({}^{O}T_{H})^{-1} \cdot {}^{O}T_{B} \cdot {}^{B}T_{M} \tag{4}$$

Fig. 3 details the spatial transformation between each system component.

C. ERROR ESTIMATION

Comparing equations (1) and (4), it can be expected that:

$$GT \approx HLT_{ta} \tag{5}$$

The difference between the transformations is the fiducial marker tracking error, and can then be obtained by:

$$E = (GT)^{-1} \cdot HLT_{ta} \tag{6}$$

Since a general transformation matrix ${}^{B}T_{A}$ of a coordinate frame A concerning a coordinate frame B can be represented as a 4×4 matrix:

$$ {}^{B}T_{A} = \begin{pmatrix} {}^{B}R_{A} & {}^{B}t_{A} \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{4\times4} \tag{7} $$

where ${}^{B}R_{A} \in \mathbb{R}^{3\times3}$ is a rotation matrix and ${}^{B}t_{A} \in \mathbb{R}^{3}$ is a three-dimensional translation vector. It is possible to derive the translation and rotation deviation between the Optitrack and the tracking algorithm running on HL2 as:

$$ E_t = \| {}^{B}t_{A} \| $$

or

$$ E_{t_{x,y,z}} = \left[E(t_x)\ E(t_y)\ E(t_z) \right]^{\top} \tag{8} $$

$$ E_r = E({}^{B}R_{A}) \tag{9} $$

where E_t is the absolute and $E_{t_{x,y,z}}$ the axis-wise distance deviation expressed in meters and E_r is the axis-wise rotation deviation that can be expressed either as quaternion or as Euler angles.

From Equation 6, the resulting error is expressed in a pose with covariance equivalent to the transformation between the expected and the measured values. Therefore, the algorithm's accuracy is extracted from the pose, expressed in each axis, and the Euclidean accuracy from the pose norm. The precision is estimated from the covariance of the pose.

IV. EXPERIMENTS

A. OPTITRACK

Optitrack's setup comprises twelve Prime^{x} 22 cameras evenly distributed in a 3D capture volume of approximately 8.3m x 5.5m x 3.0m (WxDxH). The cameras were aligned toward the center of the capture volume to create a common ground while minimizing occlusions.

The system was calibrated with Motive 3.0 and the Optitrack's calibration wand (CW-500), which essentially computes the cameras' pose and the images' distortions. After the calculation, Motive 3.0 reported a mean ray error of 0.592 mm[7] and a mean wand error of 0.155 mm[8]. Global coordinate axes and ground plane were defined using the calibration square (CS-400), a foldable L-shaped frame with adjustable leveling feet and three retroreflective markers placed on its vertices.

B. MICROSOFT HoloLens 2

To properly benchmark these applications among each other, all applications were modified to the same unity version. Those using the frontal HL2 PV camera to track the marker were set to 1920 × 1080 resolution, which is the HL2 PV camera default. In Vuforia, it was done by setting the API to use the *optimize quality mode*. The only application that uses the HL2 stereo tracking cameras is the QR Code one, which is limited to a resolution of 640 × 480 [23]. Moreover, the Unity Robotics Hub (URH) was used to establish the communication between the HL2 and the ROSMaster via Wi-Fi. All applications were set to identify single image targets. These details are summarized in Table 2.

TABLE 2. Camera type and resolution for each tracking algorithm.

Reference	Tracking algorithm	Camera Type	Camera Resolution
[1]	ARToolKit[a]	PV	1920x1080
[9]	ArUco[b]	PV	1920x1080
[38]	Vuforia[c]	PV	1920x1080
[39]	QR Code[d]	Stereo	640x480

[a] https://github.com/qian256/HoloLensARToolKit
[b] https://github.com/doughtmw/display-calibration-hololens
[c] https://developer.vuforia.com/
[d] https://github.com/microsoft/MixedReality-QRCode-Sample

(a) (b) (c)

FIGURE 4. Markers used in the experiments: (a) ARToolkit; (b) ArUco; (c) QR Code / Vuforia.

The global position and rotation of the HL2 RGB Camera and the gameObject positioned at the target location by the tracking algorithm were sent to ROS as *geometry_msgs/PoseStamped*. Moreover, a script to synchronize the HL2 clock with the NTP server running on ROSMaster was also added.

Since the QR Code marker obtained the highest possible quality rating in Vuforia Target Manager due many features, it was also used as a localization target when using the Vuforia SDK. The image target for each application is shown in Fig. 4.

Although it is not possible to modify the HL2 PV Camera intrinsic parameters used by the Vuforia API,[9] it is possible to extract its intrinsics through the *GetCameraIntrinsics* method. From these values, all tracking approaches that used the HL2 PV Camera were set to the same intrinsic parameters.

C. EXPERIMENT PROTOCOL

The marker board was positioned at different distances and orientations from the HL2 PV Camera (C) to assess the influence of the marker distance and rotations on the pose estimation. The HL2 was placed on a table and kept static, whereas the board was mounted on a tripod and moved to 25 sampling points within the experiment workspace. The height of the board and the HL2 were kept constant throughout the experiment.

Sampling points were located at 300 mm, 600 mm, 900 mm, 1200 mm, and 1500 mm with $-30°$, $-15°$, $0°$, $15°$, and $30°$ with respect to C. The marker's pose was recorded during 500 consecutive observations at each sampling point to estimate the mean error and standard deviation. A visual

[7] Represents the preciseness of the calculated 3D points during calibration.
[8] Reports the error of the detected wand length compared to the expected wand length throughout the calibration.
[9] Vuforia Unity API Reference

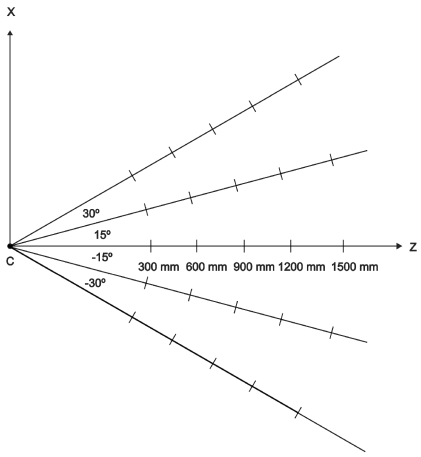


FIGURE 5. Experiment's workspace: The HL2 PV Camera (*C*) was positioned in the origin of the reference frame, and the board in each of the 25 sampling points. Sampling points range from 300 to 1500 mm and from −30° to 30°. The board is always perpendicular to the line connecting the sampling point and *C*.

representation of the experiment workspace is depicted in Fig. 5.

V. RESULTS

Figs. 6, 7, 8, and 9 depict the translation and the orientation error in the tridimensional space. From that, it is possible to verify that all marker tracking algorithms increase the translation error as the marker moves away from HL2, regardless of the marker's orientation. Tables 3, 4, 5, and 6, on the other hand, present the contribution of each axis in the errors.

It is also possible to observe the presence of significant errors at [300 mm; +30°] in the ARToolKit results (Fig. 8), at [300 mm; +30°] and [300 mm; −30°] in the ArUco results (Fig. 8), and at [900 mm; 0°] in the QR Code results (Fig. 9). In those sampling points, the respective algorithms could not properly detect the fiducial marker's pattern in such oblique angles (first three cases) or at further distances (fourth case). Although the estimations did not differ significantly among the 500 observations, the whole set of samples contained incorrect values. Therefore, recollecting the observations on those sampling points would not solve the problem.

A. VUFORIA

Regarding the Vuforia algorithm, negative board angular directions had lower translation errors than positive ones, as seen in Fig. 6. On the X-axis(width), the error for all angular directions tends upward as the board moves further from the HL2. Moreover, on the X-axis, negative board angular directions had lower translation errors than positive ones, and the error for all angular directions also trended upwardly. On the Y-axis(height), the closer the board is to the HL2, the smaller its influence over the error. Moreover, while positive angular directions tend upward as the board moves away from the HL2, negative angular directions tend downward. On the Z-axis(depth), the closer the board is to the HL2, the smaller its influence over the error. Moreover, on the Z-axis, for all angular directions of the board, the error increases on negative values as the board moves away from the HL2, indicating a downward trend, and positive board angular directions had higher translation errors than positive ones.

Regarding the influence of the distance between the HL2 and the marker on the rotation error, the error for all angular directions tends to stay within a short range of values, even though no clear pattern can be defined as can be seen in Fig. 6. On the X-axis (pitch), for all angular directions, the error contributions slightly tend towards negative values as the board moves away from the HL2. On the Y-axis(yaw) and the Z-axis(roll), the rotation error for all angular directions stays within a short range of values. On the Y-axis, no defined pattern was observed. As for the Z-axis, although there were similar variations in the results for positive angles, in −15° and −30° it was possible to observe the lowest and highest negative values, respectively.

B. ARToolKit

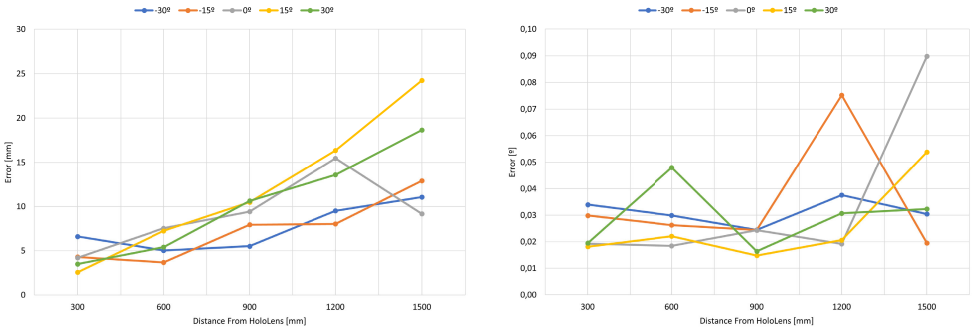
Regarding the ARToolKit algorithm, positive board angular directions had lower translation errors than negative ones, and an upward trend can be observed for all angular directions, as seen in Fig. 7. On the X, Y, and Z-axis, as the board moves further from the HL2, the error for all angular directions tends upward toward positive values. Moreover, positive board angular directions had lower translation errors than negative ones. Furthermore, in all axes, the closer the board is to the HL2, the smaller its influence over the error.

Regarding the influence of the distance between the HL2 and the marker on the rotation error, the error tends to stay within a short range of values except for the results at 0° and 1500 mm, as can be seen in Fig. 7. On the X-axis, all angular directions show a downward trend toward negative values as the board moves away from the HL2. On the Y-axis, all angular directions show an upward tendency toward positive values as the board moves away from the HL2. On the Z-axis, all angular directions stay within a short range of values.

C. ArUco

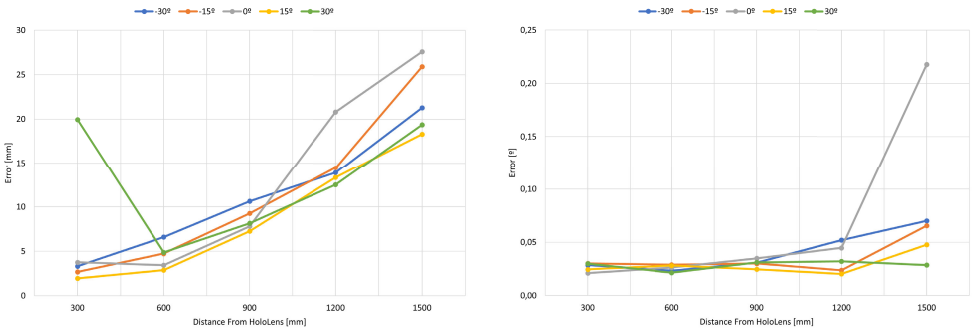
Regarding the ArUco algorithm, the translation error on the X-axis for all angular directions tends towards positive values as the board moves further from the HL2, as seen in Fig. 8. On the Y-axis, the distance between the board and the HL2 influence is more noticeable at −15° and lesser at −30°. On the Z-axis, as the board moves further from the HL2, a downward tendency can be seen for −30° and 15° angular directions.

Regarding the influence of the distance between the HL2 and the marker on the rotation error, the error stays within a short range of values despite the 0° and 1200 mm result, with a slightly upward tendency toward positive values as the marker moves further from the HL2, as can be seen in Fig. 8. The error also stays within a short range of values on the X-axis, and a slight downward tendency is observed. On the Y axis, no defined pattern is observed. On the Z-axis, the



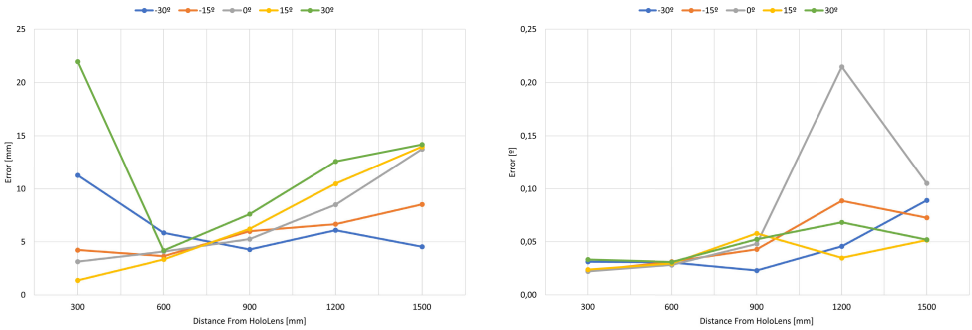
(a) (b)

FIGURE 6. Vuforia errors: (a) Translation; (b) Orientation.



(a) (b)

FIGURE 7. ARToolKit errors: (a) Translation; (b) Orientation.



(a) (b)

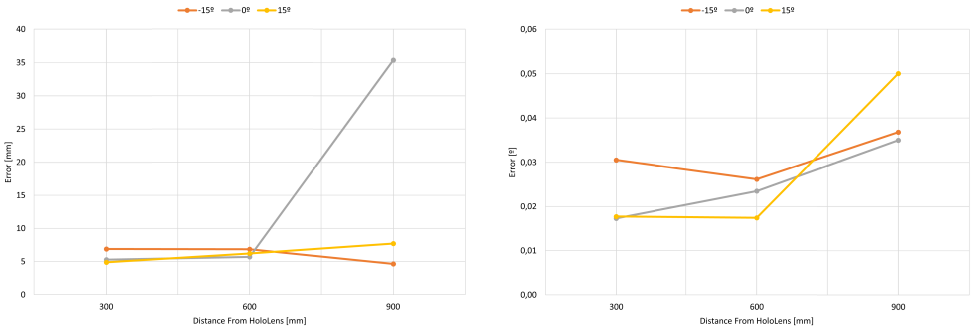
FIGURE 8. ArUco errors: (a) Translation; (b) Orientation.

rotation error also demonstrates a tendency to stay within a short range of values despite a measurement outside of this range at 30° at 300 mm, which did not exert much influence on the general analysis due to its small value. Moreover, a faint downward tendency towards negative values is noticeable.

D. QR CODE

Regarding the QR Code algorithm, the translation error for all angular directions tends to stay within a short range of values as the board moves further from the HL2, except for the 0° and 900 mm measurement, as can be seen at Fig. 9. On the X-axis, all angular directions demonstrate an upward tendency towards positive values, especially at 900 mm, where the errors show a higher increase rate. On the Y-axis, the error shows an increasing tendency toward positive values. Similarly, the Z-axis error also stood within a short range of values despite showing a slightly downward tendency.

Regarding the influence of the distance between the HL2 and the marker on the rotation error, the error stood within



(a) (b)

FIGURE 9. QR Code errors: (a) Translation; (b) Orientation.

a short range of values up to 600 mm and shows an increasing tendency in further distances, as can be seen in Fig. 9. An upward trend is observed on the X-axis for the 0° and 15° as the board went further away from the HL2, while the −15° behaved differently at 600 mm, breaking the trend. On the Y-axis, a downward tendency toward negative values is observed for all angular directions and was notably stronger for 15° at 900 mm. On the Z-axis, the error stood within a short range of values, being steadier up to 600 mm and demonstrating a converging behavior at 900 mm.

Unfortunately, the QR Code algorithm could not identify the marker at far distances and obtuse angles because of the stereo camera's limited field of view and lower resolution.

VI. DISCUSSION

Fig. 10 and Fig. 11 represent on the experiment's workspace, the algorithms that best performed in terms of accuracy and precision at each position, for translation and rotation, respectively. Among all applications, from a distance-wise perspective, regarding translation error, ArUco was the most accurate at all measured distances. ARToolKit had good accuracy at short distances, from 300 to 600 mm, but was the worst at further distances. Vuforia had low accuracy at mid-range distances, between 600 mm and 1200 mm, being the most inaccurate at 600 mm. At a distance of 300 mm, the QR Code had the worst accuracy. Despite being the most accurate, the ArUco algorithm was the most precise only at 300 mm and the worst at further distances, from 900 to 1500 mm. Vuforia was, at 900 mm, the most precise, though it was the worst in precision at 300 mm and 600 mm. QR Code also showed high imprecision from 600 to 900 mm. Regarding rotation error, the ArUco algorithm was the most inaccurate at all distances. QR Code was the most precise at shorter distances, 300 to 600 mm, and Vuforia from 900 to 1500 mm. ARToolKit also showed high accuracy at further distances, from 1200 to 1500 mm. Besides being the most inaccurate, the ArUco algorithm was also the most imprecise at distances from 600 to 1500 mm, and QR Code was the most imprecise at 300 mm, although it was the most accurate. On the other hand, ARToolKit was the most precise at all distances. Vuforia also had high precision at all distances.

From an angular direction-wise perspective, regarding translation error, the ArUco was the most accurate tracking algorithm in all measured directions. The ARToolKit was the most inaccurate from −30° to 0°, and Vuforia was the most inaccurate at −15° and 30°. The QR Code also presented high inaccuracy at −15° and 0°. Besides being the most accurate overall, the ArUco was the most precise at −15° and 0°. On the other hand, it was the most imprecise at −30° and 30° and presented high imprecision at 15°. Although the ARToolKit was the most imprecise at −15°, it was the most precise at 15° and high precision at 0°. Vuforia was the most precise at 30°. It also presented high accuracy at −15°. On the other hand, it was the most imprecise at 15°. The QR Code demonstrated high imprecision at −15° and 0°. Regarding rotation error, the ArUco was the most inaccurate in all angular directions. Vuforia was the most accurate in all angular directions. ARToolKit and QR Code showed high accuracy at −30° and 15°, respectively. Besides being the most inaccurate, ArUco was also the least precise in all angular directions. ARToolKit was the most precise from 0° to 30° and at −30°. Vuforia had the highest precision at −15° and presented high precision at 15° and 30°. The QR Code algorithm showed low precision at 15°.

In summary, ArUco is the most accurate algorithm in translation estimation, while the most inaccurate were ARToolKit and Vuforia. Regarding precision, on the other hand, the exact opposite happens is observed. Regarding rotation estimation, the most inaccurate was the ArUco, and the most accurate were the Vuforia and the QR Code algorithms. Regarding precision, ArUco was also the most imprecise, while ARToolKit was the most precise. Table 7 summarizes the strengths and weaknesses of the tested algorithms.

A. ArUco COMPARISON WITH RELATED WORKS

Before comparing this paper's results with the related works that used the ArUco algorithm, it is necessary to highlight again that these papers used different measurements and evaluation techniques. First, [9] assessed the mean translation

TABLE 3. Translation and rotation errors of the Vuforia marker tracker along the X, Y, and Z axes.

Angle	Distance	Translation Error (mm)			Rotation Error(°)		
		X	Y	Z	θX	θY	θZ
-30°	300 mm	-6.34 ± 0.13	-1.75 ± 0.08	0.47 ± 0.06	0.023 ± 0.000	0.011 ± 0.000	-0.021 ± 0.000
	600 mm	-4.66 ± 0.08	-1.82 ± 0.14	0.45 ± 0.24	0.018 ± 0.000	0.018 ± 0.001	-0.015 ± 0.000
	900 mm	-3.84 ± 0.10	-2.60 ± 0.16	-2.96 ± 0.10	0.001 ± 0.001	0.005 ± 0.004	-0.023 ± 0.000
	1200 mm	-3.44 ± 0.12	-4.00 ± 0.28	-7.90 ± 0.18	-0.024 ± 0.001	-0.021 ± 0.001	-0.019 ± 0.000
	1500 mm	-1.57 ± 0.11	-3.99 ± 0.20	-10.18 ± 0.25	-0.027 ± 0.002	-0.004 ± 0.002	-0.011 ± 0.000
-15°	300 mm	-4.03 ± 0.12	-0.79 ± 0.19	1.16 ± 0.14	0.002 ± 0.000	-0.029 ± 0.001	-0.006 ± 0.000
	600 mm	-2.35 ± 0.14	-0.89 ± 0.17	-2.67 ± 0.12	-0.010 ± 0.000	0.023 ± 0.000	-0.006 ± 0.000
	900 mm	-2.16 ± 0.04	-1.77 ± 0.10	-7.42 ± 0.02	-0.023 ± 0.000	-0.004 ± 0.000	-0.004 ± 0.000
	1200 mm	-0.02 ± 0.17	-1.32 ± 0.17	-7.91 ± 0.15	0.010 ± 0.000	-0.074 ± 0.000	-0.003 ± 0.000
	1500 mm	-0.96 ± 0.12	-2.31 ± 0.18	-12.67 ± 0.20	-0.013 ± 0.001	0.000 ± 0.001	-0.014 ± 0.000
0°	300 mm	-3.07 ± 0.09	-1.26 ± 0.14	-2.57 ± 0.10	-0.001 ± 0.000	0.013 ± 0.003	-0.013 ± 0.000
	600 mm	-2.12 ± 0.03	-0.92 ± 0.07	-7.17 ± 0.02	-0.003 ± 0.000	0.005 ± 0.000	-0.017 ± 0.000
	900 mm	0.08 ± 0.14	0.07 ± 0.22	-9.41 ± 0.22	-0.009 ± 0.000	0.018 ± 0.001	-0.012 ± 0.000
	1200 mm	1.40 ± 0.22	0.10 ± 0.23	-15.39 ± 0.42	-0.007 ± 0.001	0.009 ± 0.001	-0.014 ± 0.000
	1500 mm	4.00 ± 0.16	4.47 ± 0.32	-6.85 ± 1.41	0.085 ± 0.028	0.013 ± 0.009	-0.021 ± 0.001
15°	300 mm	-1.46 ± 0.10	-1.15 ± 0.19	-1.74 ± 0.16	0.000 ± 0.001	-0.001 ± 0.012	-0.013 ± 0.000
	600 mm	1.53 ± 0.10	-0.99 ± 0.16	-7.00 ± 0.34	-0.008 ± 0.000	0.016 ± 0.001	-0.011 ± 0.000
	900 mm	4.78 ± 0.04	-0.20 ± 0.12	-9.32 ± 0.02	-0.004 ± 0.000	-0.004 ± 0.000	-0.013 ± 0.000
	1200 mm	7.41 ± 0.20	-0.21 ± 0.28	-14.56 ± 0.25	-0.007 ± 0.001	0.011 ± 0.001	-0.014 ± 0.000
	1500 mm	10.32 ± 0.24	-0.31 ± 1.15	-21.83 ± 3.61	-0.029 ± 0.041	-0.025 ± 0.005	-0.019 ± 0.000
30°	300 mm	-1.91 ± 0.01	-0.52 ± 0.06	-2.86 ± 0.02	0.010 ± 0.000	-0.013 ± 0.000	-0.009 ± 0.000
	600 mm	1.88 ± 0.11	-0.23 ± 0.20	-5.05 ± 0.29	0.006 ± 0.002	-0.046 ± 0.001	-0.008 ± 0.000
	900 mm	5.86 ± 0.04	0.47 ± 0.11	-8.86 ± 0.02	0.005 ± 0.000	-0.006 ± 0.000	-0.014 ± 0.000
	1200 mm	9.25 ± 0.12	1.46 ± 0.20	-9.84 ± 0.03	-0.010 ± 0.000	0.025 ± 0.000	-0.014 ± 0.000
	1500 mm	14.49 ± 0.10	2.11 ± 0.25	-11.55 ± 0.51	0.006 ± 0.003	-0.027 ± 0.000	-0.014 ± 0.000

TABLE 4. Translation and rotation errors of the ARToolKit marker tracker along the X, Y, and Z axes.

Angle	Distance	Translation Error (mm)			Rotation Error(°)		
		X	Y	Z	θX	θY	θZ
-30°	300 mm	-2.52 ± 0.05	0.01 ± 0.07	2.14 ± 0.05	0.013 ± 0.000	-0.019 ± 0.000	-0.015 ± 0.000
	600 mm	1.73 ± 0.07	1.70 ± 0.09	6.13 ± 0.04	-0.011 ± 0.000	0.010 ± 0.000	-0.016 ± 0.000
	900 mm	4.91 ± 0.05	2.83 ± 0.10	8.97 ± 0.04	-0.021 ± 0.000	0.011 ± 0.000	-0.018 ± 0.000
	1200 mm	7.66 ± 0.07	4.05 ± 0.13	10.81 ± 0.06	-0.047 ± 0.000	0.016 ± 0.000	-0.013 ± 0.000
	1500 mm	10.22 ± 0.07	6.05 ± 0.15	17.66 ± 0.05	-0.060 ± 0.000	0.031 ± 0.000	-0.018 ± 0.000
-15°	300 mm	-2.62 ± 0.21	-0.46 ± 0.12	-0.23 ± 0.12	0.001 ± 0.002	-0.017 ± 0.019	-0.014 ± 0.001
	600 mm	1.16 ± 0.05	1.85 ± 0.09	4.19 ± 0.06	-0.014 ± 0.000	0.017 ± 0.000	-0.017 ± 0.000
	900 mm	3.84 ± 0.05	3.15 ± 0.11	7.81 ± 0.08	-0.017 ± 0.000	0.015 ± 0.000	-0.018 ± 0.000
	1200 mm	6.21 ± 0.12	4.78 ± 0.19	12.11 ± 0.26	-0.012 ± 0.003	0.009 ± 0.003	-0.017 ± 0.001
	1500 mm	8.21 ± 1.09	7.49 ± 1.44	23.36 ± 6.31	-0.019 ± 0.002	0.057 ± 0.027	-0.018 ± 0.000
0°	300 mm	-2.03 ± 0.05	-0.50 ± 0.07	-3.11 ± 0.08	0.003 ± 0.000	-0.010 ± 0.000	-0.017 ± 0.000
	600 mm	1.00 ± 0.12	0.44 ± 0.19	-3.25 ± 0.10	-0.014 ± 0.000	0.010 ± 0.001	-0.019 ± 0.000
	900 mm	3.84 ± 0.05	3.92 ± 0.13	5.54 ± 0.06	-0.017 ± 0.000	0.026 ± 0.000	-0.014 ± 0.000
	1200 mm	6.80 ± 0.06	7.97 ± 0.19	17.95 ± 0.10	-0.040 ± 0.000	0.001 ± 0.000	-0.018 ± 0.000
	1500 mm	6.49 ± 0.06	4.29 ± 0.19	26.48 ± 0.30	0.179 ± 0.001	-0.122 ± 0.001	-0.003 ± 0.000
15°	300 mm	-1.12 ± 0.08	-0.26 ± 0.11	-1.55 ± 0.18	0.011 ± 0.001	-0.011 ± 0.002	-0.017 ± 0.001
	600 mm	2.05 ± 0.04	1.32 ± 0.09	-1.51 ± 0.05	-0.015 ± 0.000	0.019 ± 0.000	-0.013 ± 0.000
	900 mm	4.01 ± 0.06	4.65 ± 0.15	3.86 ± 0.06	-0.019 ± 0.000	0.001 ± 0.000	-0.014 ± 0.000
	1200 mm	5.29 ± 0.06	9.31 ± 0.19	7.92 ± 0.06	-0.016 ± 0.000	-0.003 ± 0.000	-0.011 ± 0.000
	1500 mm	6.88 ± 0.07	12.11 ± 0.24	11.84 ± 0.06	-0.017 ± 0.000	0.042 ± 0.001	-0.012 ± 0.000
30°	300 mm	-7.59 ± 11.33	3.88 ± 6.38	-3.52 ± 12.02	-0.011 ± 0.004	0.022 ± 0.016	0.002 ± 0.012
	600 mm	0.18 ± 0.10	4.22 ± 0.16	2.41 ± 0.18	-0.001 ± 0.000	0.017 ± 0.000	-0.011 ± 0.000
	900 mm	2.93 ± 0.04	6.87 ± 0.12	3.26 ± 0.05	-0.018 ± 0.000	0.022 ± 0.000	-0.010 ± 0.000
	1200 mm	4.58 ± 0.06	9.57 ± 0.17	6.60 ± 0.06	-0.027 ± 0.000	0.010 ± 0.000	-0.012 ± 0.000
	1500 mm	7.03 ± 0.10	12.92 ± 0.23	12.58 ± 0.08	-0.017 ± 0.000	0.020 ± 0.000	-0.009 ± 0.000

TABLE 5. Translation and rotation errors of the ArUco marker tracker along the X, Y, and Z axes.

Angle	Distance	Translation Error (mm)			Rotation Error(°)		
		X	Y	Z	θX	θY	θZ
-30°	300 mm	5.50 ± 0.03	-0.65 ± 0.03	9.82 ± 0.03	-0.000 ± 0.000	-0.001 ± 0.000	-0.031 ± 0.000
	600 mm	-1.44 ± 0.15	-1.52 ± 0.08	5.45 ± 0.46	0.010 ± 0.007	-0.023 ± 0.011	-0.014 ± 0.001
	900 mm	-0.32 ± 0.08	-1.71 ± 0.15	3.86 ± 1.09	-0.009 ± 0.012	-0.002 ± 0.006	-0.016 ± 0.001
	1200 mm	0.64 ± 0.15	-1.50 ± 0.42	5.83 ± 1.69	-0.007 ± 0.055	0.010 ± 0.005	-0.019 ± 0.003
	1500 mm	1.99 ± 0.28	-2.05 ± 0.52	0.77 ± 3.58	-0.053 ± 0.079	0.001 ± 0.025	-0.022 ± 0.005
-15°	300 mm	-2.11 ± 0.02	-1.76 ± 0.05	3.20 ± 0.04	0.001 ± 0.001	-0.016 ± 0.000	-0.015 ± 0.000
	600 mm	-1.10 ± 0.03	-3.26 ± 0.06	-1.22 ± 0.03	-0.022 ± 0.001	0.014 ± 0.002	-0.016 ± 0.000
	900 mm	0.86 ± 0.05	-5.06 ± 0.19	-2.90 ± 1.22	-0.031 ± 0.015	0.001 ± 0.003	-0.026 ± 0.000
	1200 mm	3.36 ± 0.10	-5.41 ± 0.20	-0.36 ± 1.95	0.043 ± 0.089	0.011 ± 0.006	-0.018 ± 0.002
	1500 mm	6.18 ± 0.18	-4.80 ± 0.36	2.79 ± 1.90	-0.048 ± 0.019	0.050 ± 0.016	-0.008 ± 0.001
0°	300 mm	-2.05 ± 0.03	-2.36 ± 0.04	-0.07 ± 0.10	0.003 ± 0.003	0.002 ± 0.001	-0.021 ± 0.000
	600 mm	0.18 ± 0.07	-3.13 ± 0.09	-2.59 ± 0.26	-0.012 ± 0.002	-0.000 ± 0.006	-0.024 ± 0.001
	900 mm	5.18 ± 0.03	-0.49 ± 0.20	-0.72 ± 0.14	-0.015 ± 0.015	0.041 ± 0.002	-0.012 ± 0.001
	1200 mm	7.52 ± 0.18	-0.19 ± 0.67	3.38 ± 2.25	0.171 ± 0.113	-0.075 ± 0.077	-0.004 ± 0.004
	1500 mm	10.98 ± 0.10	-1.83 ± 0.99	-7.04 ± 4.40	0.029 ± 0.099	0.037 ± 0.017	-0.022 ± 0.001
15°	300 mm	-0.61 ± 0.14	-1.14 ± 0.08	0.32 ± 0.25	0.013 ± 0.003	-0.007 ± 0.002	-0.017 ± 0.001
	600 mm	2.70 ± 0.05	-1.08 ± 0.09	-1.61 ± 0.39	0.008 ± 0.007	-0.012 ± 0.019	-0.015 ± 0.000
	900 mm	6.03 ± 0.14	-0.46 ± 0.20	-0.59 ± 1.32	0.040 ± 0.030	-0.019 ± 0.022	-0.017 ± 0.002
	1200 mm	10.03 ± 0.17	-0.01 ± 0.38	-2.51 ± 1.81	0.003 ± 0.011	-0.023 ± 0.009	-0.021 ± 0.001
	1500 mm	12.39 ± 0.15	0.91 ± 0.36	-6.00 ± 2.19	-0.033 ± 0.023	0.017 ± 0.024	-0.019 ± 0.000
30°	300 mm	-21.79 ± 0.04	-2.40 ± 0.06	1.34 ± 0.08	-0.024 ± 0.000	0.018 ± 0.000	0.012 ± 0.000
	600 mm	3.90 ± 0.05	-0.64 ± 0.08	1.25 ± 0.49	0.013 ± 0.008	-0.022 ± 0.002	-0.013 ± 0.000
	900 mm	7.55 ± 0.12	-0.13 ± 0.19	-0.34 ± 0.90	-0.029 ± 0.012	0.032 ± 0.018	-0.019 ± 0.001
	1200 mm	12.04 ± 0.15	1.26 ± 0.30	3.18 ± 1.25	-0.056 ± 0.021	0.026 ± 0.015	-0.018 ± 0.002
	1500 mm	12.83 ± 0.23	-0.66 ± 0.31	4.70 ± 3.74	0.012 ± 0.025	-0.027 ± 0.028	-0.019 ± 0.001

TABLE 6. Translation and rotation errors of the QR Code marker tracker along the X, Y, and Z axes.

Angle	Distance	Translation Error (mm)			Rotation Error(°)		
		X	Y	Z	θX	θY	θZ
-15°	300 mm	-1.33 ± 0.14	1.56 ± 0.15	6.56 ± 0.17	0.014 ± 0.000	-0.026 ± 0.001	-0.004 ± 0.000
	600 mm	-0.32 ± 0.17	2.55 ± 0.17	6.34 ± 0.15	0.001 ± 0.000	-0.023 ± 0.008	-0.006 ± 0.000
	900 mm	1.72 ± 0.20	2.45 ± 0.37	0.89 ± 3.86	0.014 ± 0.005	-0.032 ± 0.002	-0.006 ± 0.000
0°	300 mm	-3.07 ± 0.18	0.83 ± 0.17	4.20 ± 0.22	0.008 ± 0.002	-0.002 ± 0.004	-0.014 ± 0.000
	600 mm	-0.87 ± 0.11	2.45 ± 0.14	5.05 ± 0.25	0.016 ± 0.002	-0.010 ± 0.004	-0.012 ± 0.000
	900 mm	9.74 ± 0.12	2.97 ± 0.25	-33.86 ± 0.28	0.025 ± 0.001	-0.022 ± 0.003	-0.007 ± 0.001
15°	300 mm	-2.97 ± 0.12	1.20 ± 0.16	3.67 ± 0.16	0.012 ± 0.001	0.004 ± 0.004	-0.010 ± 0.000
	600 mm	-0.80 ± 0.12	3.35 ± 0.21	5.16 ± 0.25	0.013 ± 0.003	-0.001 ± 0.004	-0.008 ± 0.000
	900 mm	6.73 ± 0.09	2.21 ± 0.28	3.00 ± 0.17	0.017 ± 0.000	-0.045 ± 0.000	-0.009 ± 0.000

TABLE 7. Strengths and weaknesses of each tracking algorithm.

Tracking algorithm	Strengths	Weaknesses
ARToolKit	Precise translation and rotation estimation Open Source Large marker library Does not block HL2 PV camera use	Inaccurate translation estimation Requires deep programming knowledge for implementations
ArUco	Accurate translation estimation Open Source Large marker library Does not block HL2 PV camera use	Inaccurate rotation estimation Requires deep programming knowledge for implementations
Vuforia	Good rotation estimation 2D and 3D tracking options Not restricted to a library Frequently updated SDK and active support Does not requires deep programming knowledge for implementations	Inaccurate translation estimation Block HL2 PV camera use Proprietary software
QR Code	Native support in HL2 Accurate rotation estimation Does not require deep programming knowledge for implementations	Limited resolution and working distance Does not support QR codes versions above 10 and with logos

error by computing the average symmetric surface distance obtained using a point-and-trace procedure. This technique can only asses the deviation in a 2D space, whether this paper assessed it in Euclidean space and with different angles.

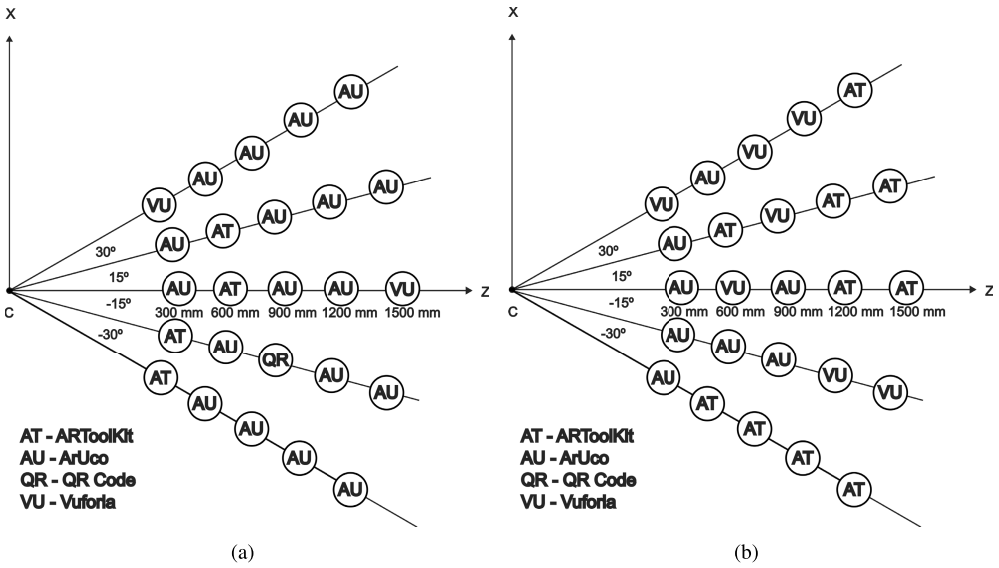


FIGURE 10. Spatial representation of the best algorithms for translation: (a) Accuracy; (b) Precision.

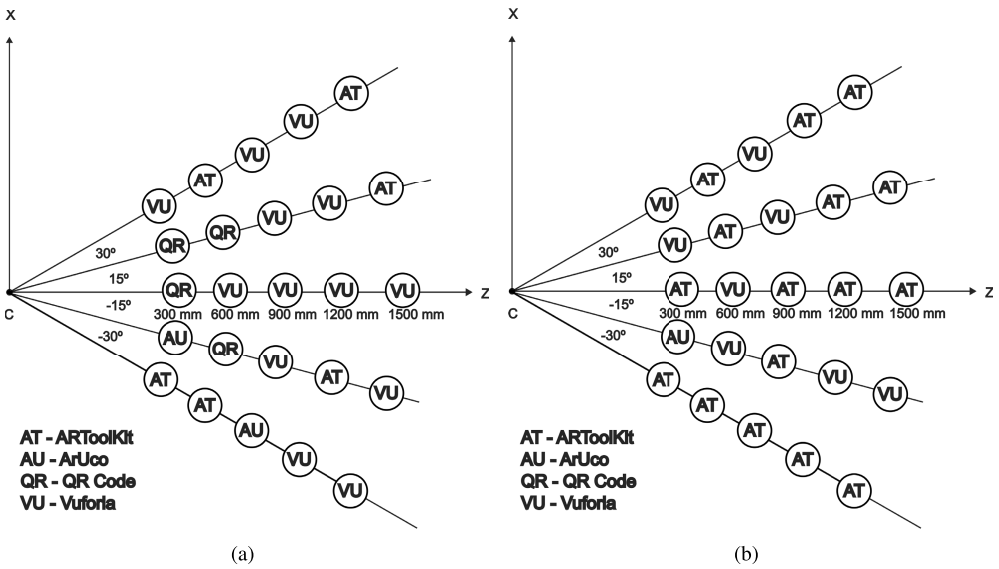


FIGURE 11. Spatial representation of the best algorithms for rotation: (a) Accuracy; (b) Precision.

Doughty et al. [9] obtained a translation error of 1.02 mm on the XY plane at an average human arm length (635 mm). This work estimated a translation error of 3.14 mm at 600 mm, under-performing in comparison to [9].

Gsaxner et al. [13], on their turn, reported a mean translation and rotation root mean square error, 6.09 mm and 6.73°, for distances from 200 to 800 mm using the ArUco algorithm. By calculating the average translation and orientation RMS error between 300 and 900 mm obtained in this paper, 6.90 mm and 0.035°, it is noticeable that this paper's translation estimation under-performed, while the rotation estimation achieved better results when compared to [13].

Thabit et al. [23] evaluated the ArUco tracking algorithm using multiple camera resolutions, marker sizes, and distances by measuring the relative measurement deviation regarding a reference position chosen at a distance of 300 mm between the HL2 and the marker. On the other hand, this paper tries to assess the absolute pose of the tag. To compare this paper's result with [23], similar conditions, resolution, marker size, and the closest possible distances are needed. Thabit's [23] 80 mm marker translation and rotation reported results are used for comparison since the authors only describe the translation error for the 100 mm marker size. Moreover, the comparison is made with the module of the values at the closest possible distance, so this paper's 600 mm result is compared to Thabit's 500 mm result, and so on. With respect to the translation estimation, on the X-axis, the only distance this paper approach performed better was at 600 mm. On the Y-axis, it performed similarly at 900 mm and better at 1200 and 1500 mm. On the Z-axis, this paper

performed better at 900 mm, similarly at 1200 mm, and underperformed at 600 mm and 1500 mm. Regarding the rotation error, this paper performed better at all axes. Overall, both papers performed similarly and accurately.

B. ARToolKit COMPARISON WITH RELATED WORKS

Quian et al., [30], assessed the translation error by computing the average 2D overlay error and not the error on Euclidean space. Nevertheless, comparing this paper's estimated average translation error at 600 mm distance, 4.507 mm, with [30], 4.27 mm, it is noticeable that this paper's result is, in the 3-dimensional space, as accurately as [30]. If comparing only the XY plane, this paper had an average translation error of 2.48 mm, performing better than [30].

Works [1], [33] assess the reprojection error with different calibrations procedures, cameras, and tracking approaches. This paper's results are compared to [1] and [33] results using mono cameras and head-anchored approaches. Since no camera calibration steps were involved in this paper, the results should be compared relative to [1] and [33] the uncalibrated scenario. Regarding the uncalibrated scenario, this paper's results were better than those presented in [1] and [33]. On the other hand, regarding the calibrated scenarios, this paper's results underperformed all approaches regarding translation estimation.

Eom et al. [31] evaluated the hologram translation error by measuring the image registration error with a caliper. This paper's approach performed better on the X and Y-axis translation estimation and underperformed on the Z-axis, at 600 mm, when compared to [31] results.

C. VUFORIA COMPARISON WITH RELATED WORKS

Eom et al. [31] and Frantz et al. [32] used the Vuforia 3D model tracking algorithm, while this paper used the Vuforia marker tracking. Moreover, both papers evaluated the registration accuracy by measuring the hologram perceived drift, the former using a digital caliper and the latter using millimeter paper and a ruler.

Regarding the Eom et al. [31] approach, this paper's results underperformed on the X and Z axes and performed slightly better on the Y-axis. Moreover, this paper supports Eom et al. [31] statement that Vuforia is more stable and sensitive to user distance and orientation with respect to the ARToolKit.

Differently from [32], where the user could walk around the reference object, in this paper's experiment the HL2 is static. Therefore, the angles referred to by Frantz et al. are regarding the user position around the marker, with the scanned object always in the center of the HMD camera field of view, while in this paper, the angles refer to the marker angular offset regarding the camera's field of view. For comparison, since [32] averaged the errors between 400 to 800 mm, this paper's results, between 300 to 900 mm, are also averaged. Frantz et al. approach performed better, 1.41 mm, than this paper's, 6.27 mm, in similar conditions.

D. QR CODE COMPARISON WITH RELATED WORKS

The differences between this paper's approach and Kyaw's et al. [37] are the tracking camera and the measurement technique. While [37] uses the Twinbuild QR Code tracking algorithm, which uses the HL2 front camera, this paper used Microsoft's [39] algorithm, which uses the HL2 stereo cameras for tracking markers. Moreover, [37] measured the registration accuracy on the xy-plane using a digital caliper, and this paper measured the estimation deviation on the 3-dimensional space. When using only one marker, this paper's approach performed better, with an average of 5.49 mm error estimation from 300 to 600 mm distance between the HL and the marker, but performed worse at 900 mm distance. When using the linearization algorithm, Kyaw et al. achieved errors as low as 0.97 mm, performing better than this paper's approach.

VII. CONCLUSION

This paper compared four marker tracking algorithms built for HL2. Overall, for near-sight applications up to 900 mm, the QR Code using the HL2 stereo cameras is rotationally stable and has low battery consumption. On the other hand, it is tight to a single low resolution and field of detection. Regarding the open-source HL2 PV camera tracking algorithms, ArUco is the best option for translation estimation applications, while ARToolKit is better for rotation estimation applications. Although the code is fully accessible in both applications, one without experience may face difficulties adapting the applications to their needs. Both algorithms also allow a large number of markers to be detected. Also, neither the ARToolKit nor the ArUco hinder the runtime access to the HL2 PV camera. Lastly, regarding rotation stability, Vuforia is the best alternative. It allows one to detect more than binary markers, such as any image or 3D models with enough features. It is useful when working with objects and can be easily implemented. However, Vuforia restricts access to the HL2 PV camera during its use, though it can be paused and resumed at will. A last point of attention is that although Vuforia has a free version, it is a proprietary software. Hence, one may encounter hindrances when accessing some protected code, information, or functionalities.

The sources of error in this study include the deviations present in the HL2 PV camera calibration and in the estimation of the transformation matrix between the HL2 PV camera and the retroreflective markers used to define the HL2 rigid body. Moreover, the intrinsic error of the Optitrack cameras used in the evaluation setup is 0.5 mm, which does not allow one to measure quantities much smaller than the obtained ones.

Benchmarking these tracking algorithms allows researchers to select the algorithm that better satisfy their application requirements. Moreover, it reveals areas for future improvement regarding the detection and usage of fiducial markers with HMDs. Projection triggering systems and fine hologram alignment, robotic calibration, map matching, and sensor

fusion are just a few examples of areas where knowing the accuracy and precision of the tracking algorithms are paramount.

Future works may involve moving the marker on the 3D space instead of only moving it on the XZ-plane, applying roll and pitch rotations instead of just yaw, testing different calibration approaches for enhancing tracking accuracy, multiple lighting and occlusion conditions, different marker sizes, and assessing the detection of more than one marker. Further analysis may also involve the deployment and testing of the technology in industrial settings.

REFERENCES

[1] 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*.

[2] A. Blaga, C. Militaru, A. D. Mezei, and L. Tamas, "Augmented reality integration into MES for connected workers," *Robot. Comput.-Integr. Manuf.*, vol. 68, Apr. 2021, Art. no. 102057, doi: 10.1016/j.rcim.2020.102057.

[3] D. Ungureanu, F. Bogo, S. Galliani, P. Sama, X. Duan, C. Meekhof, J. Stühmer, T. J. Cashman, B. Tekin, J. L. Schönberger, P. Olszta, and M. Pollefeys, "HoloLens 2 research mode as a tool for computer vision research," 2020, *arXiv:2008.11239*.

[4] 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, Jan. 2020, doi: 10.3390/s20041021.

[5] R. Radkowski and S. Kanunganti, "Augmented reality system calibration for assembly support with the Microsoft Hololens," in *Proc. Int. Manuf. Sci. Eng. Conf.*, College Station, TX, USA, 2018, pp. 1–10, doi: 10.1115/MSEC2018-6660.

[6] R. Vassallo, A. Rankin, E. C. Chen, and T. M. Peters, "Hologram stability evaluation for Microsoft HoloLens," in *SPIE Medical Imaging*, Orlando, FL, USa: SPIE, 2017, pp. 1–6, doi: 10.1117/12.2255831.

[7] X. Hu, F. R. Y. Baena, and F. Cutolo, "Alignment-free offline calibration of commercial optical see-through head-mounted displays with simplified procedures," *IEEE Access*, vol. 8, pp. 223661–223674, 2020, doi: 10.1109/ACCESS.2020.3044184.

[8] 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, doi: 10.1109/ACCESS.2019.2930701.

[9] M. Doughty and N. R. Ghugre, "Head-mounted display-based augmented reality for image-guided media delivery to the Heart: A preliminary investigation of perceptual accuracy," *J. Imag.*, vol. 8, no. 2, p. 33, pp. 1–17, doi: 10.3390/jimaging8020033.

[10] D. Khan, S. Ullah, D.-M. Yan, I. Rabbi, P. Richard, T. Hoang, M. Billinghurst, and X. Zhang, "Robust tracking through the design of high quality fiducial markers: An optimization tool for ARToolKit," *IEEE Access*, vol. 6, pp. 22421–22433, 2018, doi: 10.1109/ACCESS.2018.2801028.

[11] S. Garrido-Jurado, J. Garrido, D. Jurado-Rodríguez, F. Vázquez, and R. Muñoz-Salinas, "Reflection-aware generation and identification of square marker dictionaries," *Sensors*, vol. 22, no. 21, p. 8548, 2022, doi: 10.3390/s22218548.

[12] B. M. Khuong, K. Kiyokawa, A. Miller, J. J. La Viola, T. Mashita, and H. Takemura, "The effectiveness of an AR-based context-aware assembly support system in object assembly," in *Proc. IEEE Virtual Reality (VR)*, Minneapolis, MN, USA, Mar. 2014, pp. 57–62, doi: 10.1109/VR.2014.6802051.

[13] C. Gsaxner, J. Li, A. Pepe, D. Schmalstieg, and J. Egger, "Inside-out instrument tracking for surgical navigation in augmented reality," in *Proc. 27th ACM Symp. Virtual Reality Softw. Technol.*, Osaka, Japan, Dec. 2021, pp. 1–11, doi: 10.1145/3489849.3489863.

[14] B. J. Park, S. J. Hunt, G. J. Nadolski, and T. P. Gade, "Augmented reality improves procedural efficiency and reduces radiation dose for CT-guided lesion targeting: A phantom study using HoloLens 2," *Sci. Rep.*, vol. 10, no. 1, p. 18620, Jun. 2020, doi: 10.1038/s41598-020-75676-4.

[15] G. M. Costa, M. R. Petry, and A. P. Moreira, "Augmented reality for human–robot collaboration and cooperation in industrial applications: A systematic literature review," *Sensors*, vol. 22, no. 7, p. 2725, Apr. 2022, doi: 10.3390/s22072725.

[16] Ó. Blanco-Novoa, T. M. FernáNdez-CaraméS, P. Fraga-Lamas, and M. A. Vilar-Montesinos, "A practical evaluation of commercial industrial augmented reality systems in an industry 4.0 shipyard," *IEEE Access*, vol. 6, pp. 8201–8218, 2018, doi: 10.1109/ACCESS.2018.2802699.

[17] P. Q. Brito and J. Stoyanova, "Marker versus markerless augmented reality. Which has more impact on users?" *Int. J. Hum.–Comput. Interact.*, vol. 34, no. 9, pp. 819–833, Nov. 2017, doi: 10.1080/10447318.2017.1393974.

[18] A. Blaga and L. Tamas, "Augmented reality for digital manufacturing," in *Proc. 26th Medit. Conf. Control Autom. (MED)*, Zadar, Croatia, Jun. 2018, pp. 173–178, doi: 10.1109/MED.2018.8443028.

[19] I. Soares, M. R. Petry, and A. P. Moreira, "Programming robots by demonstration using augmented reality," *Sensors*, vol. 21, no. 17, p. 5976, Sep. 2021, doi: 10.3390/s21175976.

[20] S. Blankemeyer, R. Wiemann, L. Posniak, C. Pregizer, and A. Raatz, "Intuitive robot programming using augmented reality," in *Proc. 7th CIRP CATS*, Tianjin, China, 2018, pp. 155–160, doi: 10.1016/j.procir.2018.02.028.

[21] R. Kalpagam Ganesan, Y. K. Rathore, H. M. Ross, and H. Ben Amor, "Better teaming through visual cues: How projecting imagery in a workspace can improve human–robot collaboration," *IEEE Robot. Autom. Mag.*, vol. 25, no. 2, pp. 59–71, Jun. 2018, doi: 10.1109/MRA.2018.2815655.

[22] I. Soares, R. B. Sousa, M. R. Petry, and A. P. Moreira, "Accuracy and repeatability tests on HoloLens 2 and HTC Vive," *Multimodal Technol. Interact.*, vol. 5, no. 8, p. 47, Aug. 2021, doi: 10.3390/mti5080047.

[23] A. Thabit, W. J. Niessen, E. B. Wolvius, and T. van Walsum, "Evaluation of marker tracking using mono and stereo vision in Microsoft HoloLens for surgical navigation," in *SPIE Medical Imaging*. San Diego, CA, USA, SPIE, 2022, pp. 253–262. Apr. 2022, doi: 10.1117/12.2607262.

[24] E. Marino, F. Bruno, L. Barbieri, and A. Lagudi, "Benchmarking built-in tracking systems for indoor AR applications on popular mobile devices," *Sensors*, vol. 22, no. 14, p. 5382, Jul. 2022, doi: 10.3390/s22145382.

[25] S. Park, S. Bokijonov, and Y. Choi, "Review of Microsoft hololens applications over the past five years," *Appl. Sci.*, vol. 11, no. 16, p. 7259, 2021, doi: 10.3390/app11167259.

[26] A. Sadhu, J. E. Peplinski, A. Mohammadkhorasani, and F. Moreu, "A review of data management and visualization techniques for structural health monitoring using BIM and virtual or augmented reality," *J. Struct. Eng.*, vol. 149, no. 1, 2022, Art. no. 03122006, doi: 10.1061/(ASCE)ST.1943-541X.0003498.

[27] A. Mohammadkhorasani, K. Malek, R. Mojidra, J. Li, C. Bennett, W. Collins, and F. Moreu, "Augmented reality-computer vision combination for automatic fatigue crack detection and localization," *Comput. Ind.*, vol. 149, Aug. 2023, Art. no. 103936, doi: 10.1016/j.compind.2023.103936.

[28] A. Hietanen, R. Pieters, M. Lanz, J. Latokartano, and J. K. Kämäräinen, "AR-based interaction for human–robot collaborative manufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 63, Jun. 2020, Art. no. 101891, doi: 10.1016/j.rcim.2019.101891.

[29] D. Puljiz, F. Krebs, F. Bosing, and B. Hein, "What the HoloLens maps is your workspace: Fast mapping and set-up of robot cells via head mounted displays and augmented reality," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, Oct. 2020, pp. 11445–11451, doi: 10.1109/IROS45743.2020.9340879.

[30] L. Qian, A. Deguet, and P. Kazanzides, "ARssist: Augmented reality on a head-mounted display for the first assistant in robotic surgery," *Healthcare Technol. Lett.*, vol. 5, no. 5, pp. 194–200, Sep. 2018, doi: 10.1049/htl.2018.5065.

[31] S. Eom, S. Kim, S. Rahimpour, and M. Gorlatova, "AR-assisted surgical guidance system for ventriculostomy," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces Abstr. Workshops (VRW)*, Christchurch, New Zealand, Mar. 2022, pp. 402–405, doi: 10.1109/VRW55335.2022.00087.

[32] T. Frantz, B. Jansen, J. Duerinck, and J. Vandemeulebroucke, "Augmenting Microsoft's HoloLens with vuforia tracking for neuronavigation," *Healthcare Technol. Lett.*, vol. 5, no. 5, pp. 221–225, Sep. 2018, doi: 10.1049/htl.2018.5079.

[33] M. Brand, L. A. Wulff, Y. Hamdani, and T. Schüppstuhl, "Accuracy of marker tracking on an optical see-through head mounted display," in *Annals of Scientific Society for Assembly, Handling and Industrial Robotics*. Berlin, Germany: Springer Vieweg, 2020, pp. 21–31, doi: 10.1007/978-3-662-61755-7_3.
[34] C. Kunz, P. Maurer, F. Kees, P. Henrich, C. Marzi, M. Hlaváč, M. Schneider, and F. Mathis-Ullrich, "Infrared marker tracking with the HoloLens for neurosurgical interventions," *Current Directions Biomed. Eng.*, vol. 6, no. 1, 2020, Art. no. 20200027, doi: 10.1515/cdbme-2020-0027.
[35] J. L. van Smoorenburg, "Using mixed-reality in biopsy of lesions inside deformable organs," M.S. thesis, Robot. Mechatronics, Univ. Twente, Twente, Enschede, 2021.
[36] P. Hübner, M. Weinmann, and S. Wursthorn, "Marker-based localization of the Microsoft Hololens in building models," in *Proc. ISPRS TC*, Karlsruhe, Germany, 2018, pp. 195–202, doi: 10.5194/isprs-archives-XLII-1-195-2018.
[37] A. H. Kyaw, A. H. C. Xu, G. Jahn, N. van den Berg, C. Newnham, and S. Zivkovic, "Augmented Reality for high precision fabrication of glued laminated timber beams," *Automat. Construct.*, vol. 152, Aug. 2023, Art. no. 104912, doi: 10.1016/j.autcon.2023.104912.
[38] *Vuforia Engine | Developer Portal*. Accessed: Nov. 10, 2023. [Online]. Available: https://developer.vuforia.com/
[39] *Microsoft | QR Code Tracking Overview*. Accessed: Nov. 10, 2023. [Online]. Available: https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/qr-code-tracking-overview
[40] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, "Fiducial markers for pose estimation: Overview, applications and experimental comparison of the ARTag, apriltag, aruco and stag markers," *J. Intell. Robotic Syst.*, vol. 101, p. 71, Mar. 2021, doi: 10.1007/s10846-020-01307-9.
[41] K. Shabalina, A. Sagitov, L. Sabirova, H. Li, and E. Magid, "ARTag, AprilTag and CALTag fiducial systems comparison in a presence of partial rotation: Manual and automated approaches," in *Proc. 14th Int. Conf. Inform. Control, Automat. Robot. (ICINCO)*, Madrid, Spain, Jul. 2017, pp. 536–558, doi: 10.1007/978-3-030-11292-9_27.

GABRIEL M. COSTA received the bachelor's degree in control and automation engineering–mechatronics from Centro Universitário de Maringá, and the master's degree in chemical engineering from Universidade Estadual de Maringá. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Faculdade de Engenharia da Universidade do Porto. He received the Postgraduate Certificate in mechatronics engineering from Centro Universitário de Maringá. His research field focus on augmented reality, robotics, automatic control, automation, and control systems.

MARCELO R. PETRY received the Ph.D. degree in informatics engineering from the University of Porto, in 2013. He is a robotics and computer vision researcher and an educator that works on robot perception, human–robot interaction, and the application of robots to logistics, inspection, and human assistance. Since 2019, he has been a Senior Researcher with INESC TEC's Centre for Robotics in Industry and Intelligent Systems. Previously, he was an Assistant Professor with the Federal University of Santa Catarina and a Researcher with INESC P&D Brazil.

JOÃO G. MARTINS received the M.Sc. degree in electrical and computer engineering, specialization in automation from Faculdade de Engenharia da Universidade do Porto, Portugal, in 2023, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. As a Researcher Trainee, he joined the Centre for Robotics in Industry and Intelligent Systems, INESC TEC, in 2021, and has been a Researcher, since 2022, working on multiple projects. His research interests comprise perception, robotics, automation, and autonomous systems.

ANTÓNIO PAULO G. M. MOREIRA received the degree in electrical engineering from the University of Oporto, in 1986, and the M.Sc. degree in electrical engineering–systems and the Ph.D. degree in electrical engineering from the University of Porto, in 1991 and 1998, respectively. Currently, he is a Full Professor (tenure) with Faculdade de Engenharia da Universidade do Porto, and a Researcher and the Head of the Robotics and Intelligent Systems Centre, INESC TEC. His main research interests include process control and robotics.