Article

Benchmarking Built-In Tracking Systems for Indoor AR Applications on Popular Mobile Devices

Emanuele Marino [1,*], Fabio Bruno [1,2], Loris Barbieri [1] and Antonio Lagudi [1,2]

1   Department of Mechanical, Energy and Management Engineering (DIMEG), University of Calabria, P. Bucci, 87036 Rende, Italy; fabio.bruno@unical.it (F.B.); loris.barbieri@unical.it (L.B.); antonio.lagudi@unical.it (A.L.)
2   3D Research s.r.l., 87036 Rende, Italy
*   Correspondence: emanuele.marino@unical.it

Abstract: As one of the most promising technologies for next-generation mobile platforms, Augmented Reality (AR) has the potential to radically change the way users interact with real environments enriched with various digital information. To achieve this potential, it is of fundamental importance to track and maintain accurate registration between real and computer-generated objects. Thus, it is crucially important to assess tracking capabilities. In this paper, we present a benchmark evaluation of the tracking performances of some of the most popular AR handheld devices, which can be regarded as a representative set of devices for sale in the global market. In particular, eight different next-gen devices including smartphones and tablets were considered. Experiments were conducted in a laboratory by adopting an external tracking system. The experimental methodology consisted of three main stages: calibration, data acquisition, and data evaluation. The results of the experimentation showed that the selected devices, in combination with the AR SDKs, have different tracking performances depending on the covered trajectory.

1. Introduction

In recent years, several improvements in computer technology have opened up a wide range of new possibilities in the field of immersive technologies, which can be applied to a plethora of different contexts. Immersive systems, which include Augmented Reality (AR), Virtual Reality (VR), and Mixed Reality (MR), have led to new ways of perceiving the real-world environment by creating exciting experiences for many people around the world.

Among them, AR is a key technology that enriches the real world with contextualized virtual computer-generated information by offering users real-time interaction possibilities [1,2]. Thanks to these capabilities, AR technologies have been adopted in many application areas such as gaming, entertainment, medicine, marketing, the military, and in particular, in the industrial field where AR has been demonstrated to be a valuable tool for improving and accelerating product and process development [3–13].

In general, an augmented scenario needs to achieve high accuracy in regard to the positioning of the virtual content in the real-world environment to create the illusion of seamless integration. To this end, AR solutions require an accurate built-in tracking system to track the position and orientation of the user with respect to the physical world; this is one of the most important AR tasks [14]. Essentially, tracking techniques can be classified according to two main approaches: sensor-based and visual-based [15–17]. The first approach uses data provided by active sensors mounted on the device itself, such as GPS, accelerometers, gyroscopes and magnetometers to track the user's pose. Conversely, the visual-based approach exploits computer vision methods by means of images captured by the device camera that are analyzed at run-time. In particular, this approach can be

roughly divided into marker-based and marker-less techniques [18]. A more modern and promising solution is hybrid tracking, which combines sensor- and visual-based approaches to increase the quality of the pose estimation by reducing tracking errors [19,20]. Recently, high-end mobile devices have been equipped with Laser Imaging Detection and Ranging (LIDAR) sensors since their cost has decreased rapidly and significantly. This has permitted the development of more accurate and robust localization tracking algorithms [21].

At the same time, the rapid development of mobile technologies, e.g., smartphones and tablets, with increasingly powerful performances at a lower cost and application programming interfaces have enabled the development and operation of AR applications on these devices. This is also due to their versatility and ease of use. Among the best-known frameworks for rapid prototyping of AR solutions, ARKit$^{®}$ and ARCore^{TM} represent the most constantly evolving ones for iOS$^{®}$ and Android^{TM} devices, respectively [22,23]. They use a process called simultaneous localization and mapping (SLAM) to localize the device in real-world coordinates. In the literature, several AR applications, which are based on these libraries, have been proposed for different use cases [24–28]. Nevertheless, there are no studies that address the estimation of tracking accuracy of different mobile devices while considering AR libraries and price ranges at the same time.

On the basis of these considerations, this paper contributes to this area of research by presenting an experimental evaluation of the tracking capabilities of some popular mobile devices, which should be a representative set of those currently on sale in the global market. In particular, the price range, platform, i.e., iOS$^{®}$ and Android^{TM}, and hardware specifications were taken into account during the selection process. In line with similar works, an external tracking system was adopted to obtain the ground truth in order to perform the comparison with the estimated tracking data [29–31]. More specifically, the OptiTrack^{TM} motion capture system was used since it uses multiple fixed IR cameras to localize and track retro-reflective markers with high accuracy [32]. Experiments were conducted in a laboratory room and considered multiple paths to assess the tracking accuracy for both open and closed trajectories. The analysis of the tracking data was performed by considering standard metrics and descriptive statistics. It is important to note that this work has been presented in support of previous research conducted in the industrial field where mobile AR technology has been adopted to support operators during industrial activities [33–35].

The rest of this paper is organized as follows: Section 2 describes in detail the methodology adopted for the evaluation of tracking accuracy. In Section 3, the results and a discussion are presented and the conclusions are provided in Section 4.

2. Related Works

Various research studies have included evaluations of tracking performances for different devices and AR applications. Feigl et al. [36] studied the applicability of AR solutions, such as ARKit, ARCore, and Microsoft HoloLens, by measuring their tracking accuracy in an industrial scenario. They considered both small- and long-scale measurements with different paths by acquiring the ground truth by means of an external optical system. Hasler et al. [37] conducted a performance evaluation of an indoor mapping application by adopting ARCore and ARKit frameworks. In [38], the authors tested the tracking accuracy of the Microsoft HoloLens, which is a head-worn mobile AR device. Cortès et al. [39] compared the tracking data captured by some handheld devices including iPhone, Google Pixel, and Google Tango. As can be observed, tracking accuracy is a research topic trend in the literature because it represents a key aspect of AR technologies that ensures they can be used effectively. However, to the best of our knowledge, there are no specific studies that deal with the evaluation of the tracking performance of AR mobile systems, which combine proprietary AR frameworks, i.e., ARKit$^{®}$ and ARCore^{TM}, with different handheld devices.

3. Materials and Methods

In this section, the methodology developed for estimating the tracking accuracy of some popular mobile devices that are equipped with a built-in tracking system is presented. First, a description of the mobile devices selected for the experimentation is provided. Then, the experimental setup is presented. Next, the mathematical model implemented for the calibration and the comparison between the estimated and ground-truth trajectories is described. Finally, the data analysis procedure as well as the metrics adopted for the assessment of tracking accuracy are defined.

3.1. Description of Selected Devices

For the selection of the mobile devices employed in the experiments, several aspects were considered such as their compatibility with AR frameworks, the release date, price range, platform, and other specific parameters related to the hardware components. First, the choice was made by considering devices compatible with the ARKit® and ARCore™ frameworks, which are some of the most common AR libraries, and in turn, they are most used in many research works [40]. In general, devices with iOS® operating system can run AR applications developed with ARKit®, whereas apps based on ARCore™ can be used with Android™ devices. Secondly, popular mid- to high-end smartphones and tablets in terms of prices and hardware performances, released in the first semester of 2021, were considered since experiments were conducted in the fall of the same year. This choice was also supported by the need to assess whether various built-in tracking systems, consisting of inertial and optical sensors, affected the tracking accuracy. In other words, the selected devices should be representative of the mobile global market. The following table (Table 1) provides an overview of the main relevant specifications for each device.

Table 1. Selected mobile devices and their main specifications.

Manufacturer	Device	Type	Price Range	Platform	CPU (GHz)	RAM (GB)	Main Cameras	Camera Resolution	Release Date
Apple	iPad Pro 11	Tablet	High	iOS	Octa-core ($4 \times 2.5 + 4 \times 1.6$)	8	3	12 MP, 10 MP	May 2021
Apple	iPhone 11	Smartphone	High	iOS	Hexa-core ($2 \times 2.65 + 4 \times 1.8$)	4	2	12 MP	May 2021
Samsung	Tab S6	Tablet	Medium	Android	Octa-core ($1 \times 2.84 + 3 \times 2.42 + 4 \times 1.78$)	4	2	13 MP, 5 MP	July 2019
Samsung	S20	Smartphone	High	Android	Octa-core ($2 \times 2.73 + 2 \times 2.50 + 4 \times 2.0$)	8	3	12 MP, 64 MP	March 2020
Samsung	S10	Smartphone	Medium	Android	Octa-core ($2 \times 2.73 + 2 \times 2.31 + 4 \times 1.95$)	8	3	12 MP, 16 MP	March 2019
Oppo	Find X3 Lite	Smartphone	Low	Android	Octa-core ($1 \times 2.4 + 1 \times 2.2 + 6 \times 1.8$)	8	4	64 MP, 8MP, 2 MP	March 2021
Xiaomi	MI 11 Lite	Smartphone	Low	Android	Octa-core ($2 \times 2.3 + 6 \times 1.8$)	6	3	64 MP, 8 MP, 4 MP	March 2021
Xiaomi	Redmi Note 10	Smartphone	Low	Android	Octa-core ($2 \times 2.2 + 6 \times 1.7$)	4	4	48 MP, 8 MP, 2 MP	March 2021

As for price ranges, devices were categorized into three different groups: low (up to 350 euros), medium (from 350 euros to 450 euros), and high (from 450 euros to 600 euros). Furthermore, the number and resolution of the main cameras shown in Table 1 refers to the rear-facing ones in the mobile device.

3.2. Experimental Setup

In order to track the movements, in terms of translation and rotation, of the selected devices while moving within the laboratory room, a specific experimental setup was prepared as shown in Figure 1.
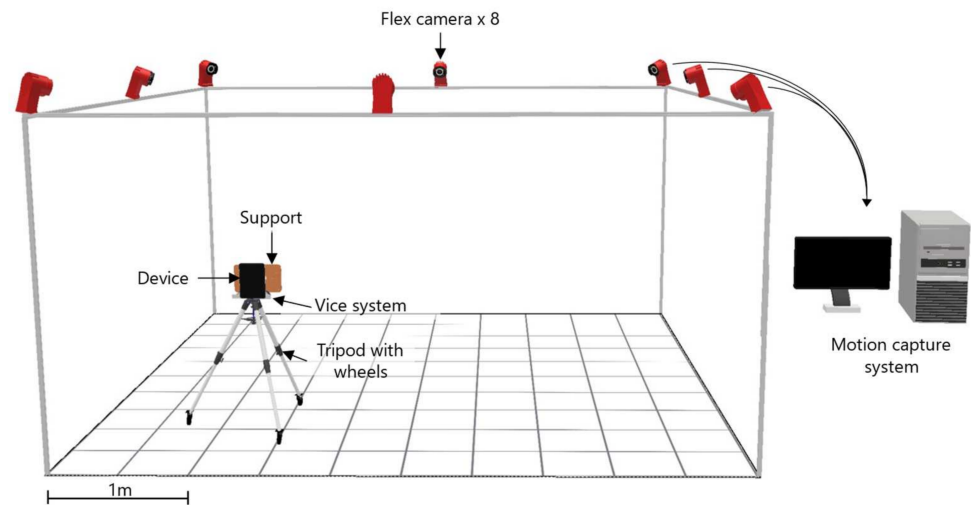
Figure 1. The experimental setup used during the experimentation phase.

The experimental setup consisted of several components including the mobile device, an external tracking system connected to a desktop PC, wooden support, a vice system, and a tripod with wheels. Since the accuracy of the built-in tracking system of the selected devices is expected to be of the order of centimeters [36], an optical tracking system was adopted to capture the movements of the mobile devices and obtain the ground truth for data comparison and analysis. To this end, the OptiTrack$^{\text{TM}}$ motion capture system was used because it features high-quality tracking with sub-millimeter accuracy [41,42]. It consisted of eight Flex13 InfraRed cameras located in the laboratory room, which was about 25 square meters, and connected to a desktop PC with USB cables. The PC runs the OptiTrack$^{\text{TM}}$ software that captures and processes the tracking data. The cameras were placed 3 m from the floor, covering an effective capture volume of about 2.5 m × 2.5 m × 2 m to prevent object occlusions during the acquisition stage. The experimental setup also comprised a wooden support, specifically designed and prototyped for each device, and instrumented with eleven retro-reflective markers. The wooden supports were designed in order to meet the following requirements. First, the reflecting markers should be asymmetrical to avoid recognition issues during the tracking process performed by the OptiTrack$^{\text{TM}}$ motion capture system as suggested by the producer. Secondly, eleven different markers were considered to ensure high-stability tracking. A vice system was used to connect the device to the support. Finally, the device and the support were mounted on a tripod with three wheels at the bottom.

3.3. Evaluation Method

This section describes the evaluation method that have been used to assess the tracking accuracy. In this regard, the tracking data gathered from the selected devices is compared with the ground truth provided by the external tracking system, which is capable of estimating the pose of a target with higher-order accuracy. Figure 2 depicts the mathematical model adopted to transform the device tracking data from a local to a world reference system, and as a consequence, allowing the comparison with the ground truth.

In particular, three main reference systems can be identified in Figure 2, i.e., {W}, {S}, {L}. The reference frame {W} represents a fixed-world coordinate system defined by the OptiTrack$^{\text{TM}}$ motion capture system, and it is usually located on the floor of the room. The reference frames {S} and {L} correspond to local coordinate systems identified by the support and the device, respectively. Specifically, {S} and {L} are placed in correspondence to the center of gravity of retro-reflective markers and the center of the camera, respectively. The external tracking system is able to track the support while moving in the real world. The homogeneous matrix $_S^W T(t)$, which is time-dependent, contains its pose in terms of positions and orientations with respect to the {W} reference frame. On the other hand, the

device can track its pose in the 3D space by means of the built-in tracking system, which combines data provided by inertial and optical sensors. In this case, the device pose $^L P(t)$ is estimated with respect to *{L}*, which is set when the AR application is started. During the experimentation, the mobile device is joined to its support so that they move together. The homogeneous matrix $^S_L T$ represents the spatial offset between the reference systems of the support and the device. In order to compare the estimated tracking data of the device with the ground truth gathered by capturing the motion of the support, it is necessary to express the poses of the device $^L P(t)$ with respect to the global reference system *{W}* by means of the transformation matrix $^W_L T(t)$. To this end, $^W P(t)$ can be computed through the following formulation (Equation (1)):

$$^W P(t) = {}^W_S T(t) \bullet {}^S_L T \bullet {}^L P(t) \tag{1}$$

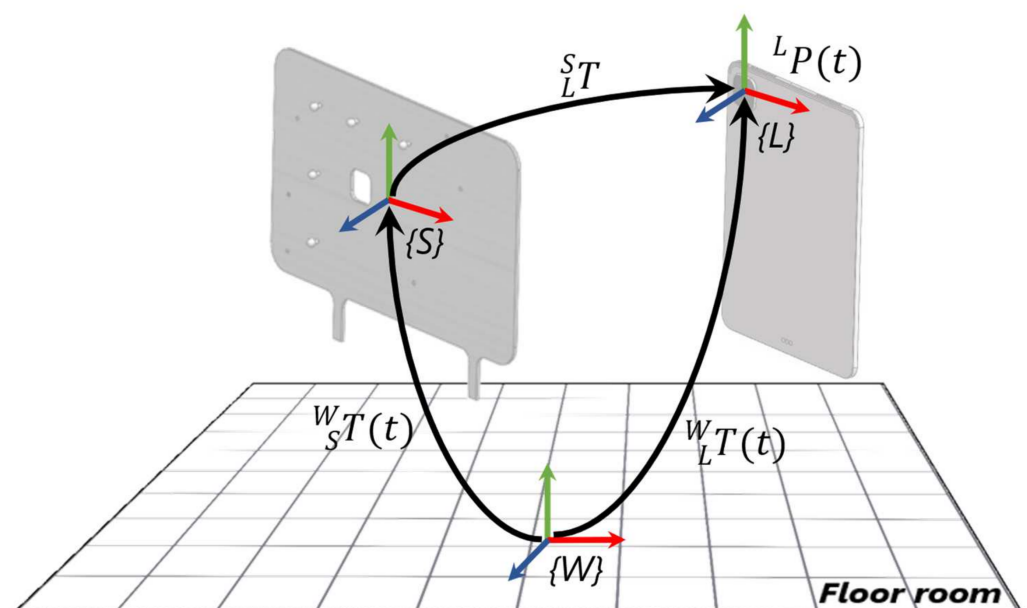A schematic overview of the evaluation framework is illustrated in Figure 3.

Figure 2. Mathematical model representation for comparing tracking data.

In particular, it consists of three main stages: calibration, data acquisition, and data analysis, which enables the comparison of the tracking data from the device with the ground truth.

The first stage, i.e., the calibration, aims to estimate the spatial offset that exists between the reference frame of the support and the device when they are integral to each other. More specifically, this is represented by the constant transformation $^S_L T$, which has to be computed before starting the testing phase.

The second stage, i.e., the data acquisition, consists of recording the trajectories for both the device and the support. The first is captured by a mobile app based on AR SDK that runs on the device itself, whereas the ground truth is recorded by means of a desktop application connected to the OptiTrack^{TM} software, namely, Motive.

In the third stage, the device tracking data are transformed and time-synchronized in order to be processed and evaluated. Then, a comparison of the tracking of the device and the external tracking system is performed by means of a script implemented in MATLAB$^{\circledR}$.
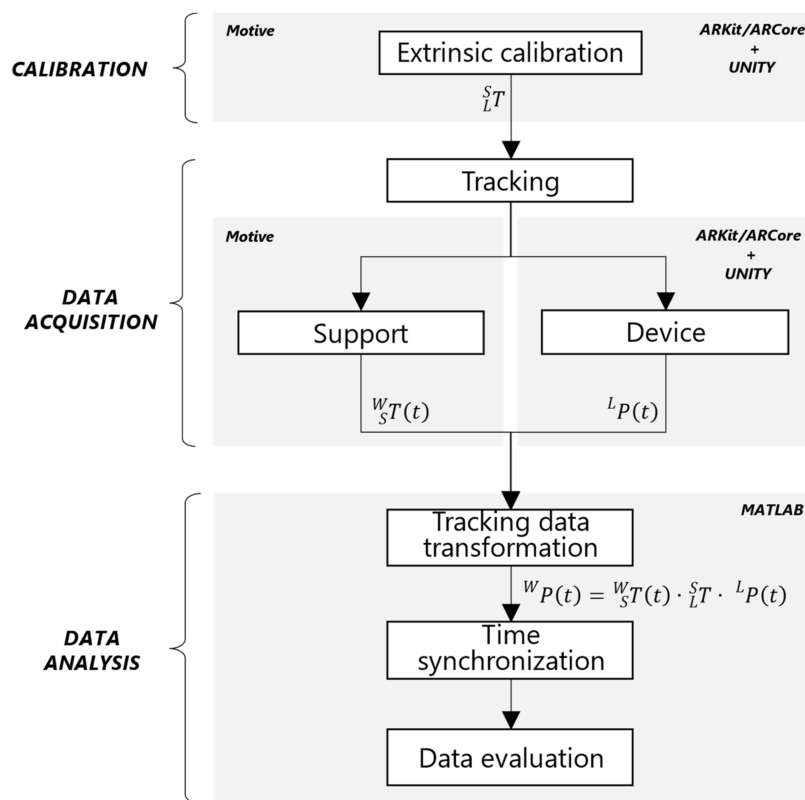
Figure 3. A schematic overview of the evaluation framework adopted for the acquisition and comparison of the tracking data.

3.3.1. Calibration

The spatial offset $_L^S T$ between the local reference system identified by the rigid body markers attached to the support *{S}* and that relative to the device *{L}* has to be experimentally determined by following a calibration procedure. Figure 4 shows the mathematical model that was used to measure it.
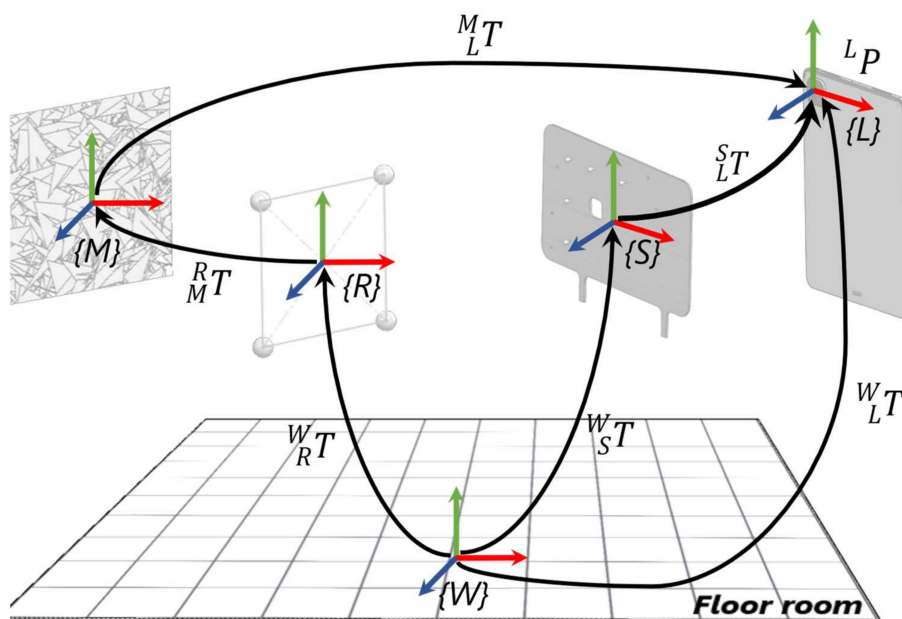

Figure 4. Mathematical model used to estimate the spatial offset $_L^S T$.

On the basis of the previous schema (Figure 4), the pose of the device $^W P$ expressed in the global reference system *{W}* can be calculated by means of two different equations as follows (Equations (2) and (3)):

$$^W P = {}^W_S T \bullet {}^S_L T \bullet {}^L P \tag{2}$$

$$^W P = {}^W_R T \bullet {}^R_M T \bullet {}^M_L T \bullet {}^L P \tag{3}$$

By considering Equations (2) and (3), it follows (Equation (4)):

$$^W_S T \bullet {}^S_L T \bullet {}^L P = {}^W_R T \bullet {}^R_M T \bullet {}^M_D T \bullet {}^L P \tag{4}$$

Then finally, the unknown rigid transformation matrix ${}^S_L T$ is obtained as follows (Equation (5)):

$$^S_L T = ({}^W_S T)^{-1} \bullet {}^W_R T \bullet {}^R_M T \bullet {}^M_L T \tag{5}$$

The pose ${}^M_L T$ of the device relative to the AR marker can be determined by means of the AR SDK whereas the ${}^R_M T$ transformation matrix consists of a combination of elementary rotation matrices to align the correspondent reference systems. In addition, this transformation matrix can be expressed through the following equation (Equation (6)):

$$^R_M T = Rz(180°) * Rx(90°) \tag{6}$$

Figure 5 shows the setup prepared for the calibration stage.
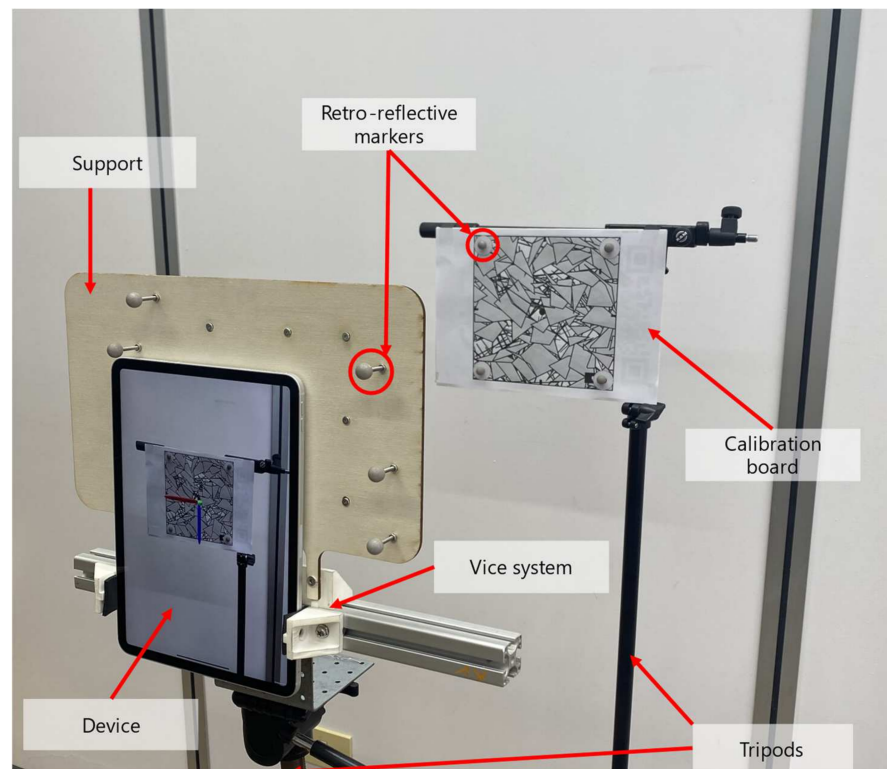
Figure 5. Calibration setup.

A calibration board was fixed on a tripod placed in the laboratory room. It consisted of a marker detectable by the AR device with four retro-reflective markers placed at its outer corners in correspondence to predefined locations. The device equipped with the support trackable by the external tracking systems was placed in a static setting in front of the calibration board. In addition, the pose relative to the calibration board was acquired

ten times for each device and averaged to provide more representative data and achieve more reliability.

3.3.2. Data Acquisition

The second stage consisted of the data acquisition process. To record the trajectories of the device and support, two applications were programmed in Unity, which is a cross-platform game engine for the rapid prototyping of software solutions [43]. Regarding the desktop application, this was able to connect to the OptiTrack^{TM} motion capture system Motive (the optical motion capture software by OptiTrack^{TM}) through the OptiTrack^{TM} Unity plugin. In the case of the mobile application, tracking data was acquired by using the specific AR SDK of each device platform, and exported into a CSV file for offline data analysis.

Experiments were performed on multiple paths with different shapes to assess their influence on tracking accuracy (Table 2) for both open and closed trajectories. Each test was repeated ten times in a laboratory room by using all the selected devices.

Table 2. The three paths considered during the experimentation field.

| Path (#)	Shape	Length (m)	Mean Duration (S)
1	∞	13	160
2	□	8	230
3	⊐	6	30

3.3.3. Preliminary Tests

With regard to the paths characterized by a closed trajectory, it is necessary to define the number of laps that should be performed for each test, to obtain valuable results from the experimentation. Generally, this is not known a priori, and without any solid criteria it can lead to long-lasting experiments, especially in the case of more devices. At the same time, it is reasonable to suppose that this quantity can be obtained in correspondence with a stabilized tracking process. As mentioned above, AR technology adopts visual-inertial SLAM in order to achieve a more precise local trajectory estimation through an iterative optimization procedure. In particular, a map of the environment is produced simultaneously with the tracking, which is used to reduce drift error accumulated during the device movement [44,45]. This is possible thanks to the estimation of the position of previously initialized visual features and performing a loop-closure process [46,47].

On the basis of these considerations, a series of experiments were conducted by using a mid-end device and capturing its pose while moving along a closed trajectory over a square-shaped path. The test consisted of 20 repetitions of 20 laps. Then, the resultant values were averaged, and the Euclidian distance between the estimated trajectory and ground truth was computed. Figure 6 shows the trend in the error over time, where the error bars represent a 95% confidence interval.

The results showed that the averaged error levelled off after lap 5, and then it remains almost constant with a mean value of 0.187 m and a standard deviation of 0.005 m. According to these results, it is possible to conclude that 5 laps, for each test, should be enough to achieve high tracking stability. Therefore, a total of 10 laps were taken into consideration for more test reliability.
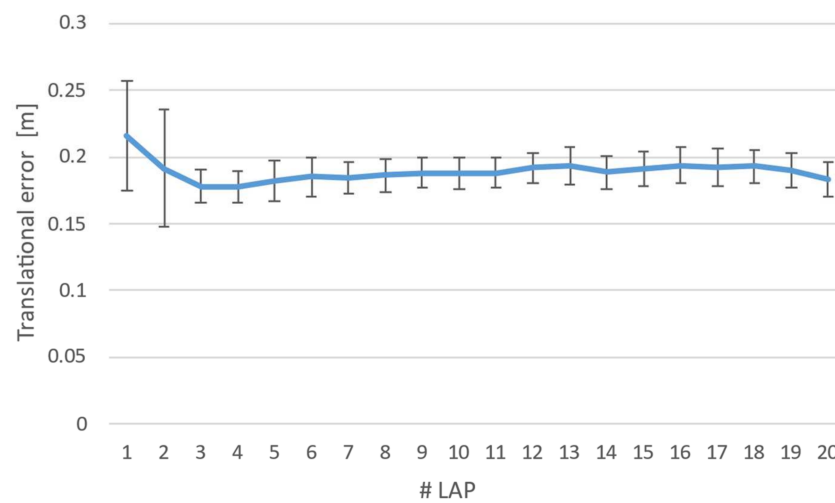
Figure 6. Translational error estimated during the preliminary experimentation for a total of 20 laps.

3.3.4. Data Analysis

The analysis of the tracking data was executed offline using a script implemented in MATLAB®, which performs two main operations: tracking data transformation, and time synchronization. Both operations are mandatory for performing a tracking data comparison since they usually refer to different reference frames and are acquired from different systems, with specific timestamps. In particular, the tracking data transformation operation consists of a coordinate transformation process that transforms tracking data from the device in the global reference frame {W} according to Equation (1).

Furthermore, it is important to note that the comparison between the tracking data should be achieved by labelling the corresponding poses with a common timestamp. Thus, the time synchronization is mandatory because the rates of sampling were different. In this case, the OptiTrack^{TM} captured tracking data at 4 Hz, whereas the device collected poses at 10 Hz. To perform this temporal alignment, a specific script called *FilterData* has been developed in MATLAB® to best-fit the tracking data gathered from both systems. In particular, the synchronization methodology consists of selecting poses recorded by the device that are the nearest to OptiTrack^{TM} ones in terms of timestamp. This is performed by calculating the percent error for each combination and selecting that with the minimum error value according to the following equation (Equation (7)):

$$\text{PE} = \frac{\left| {}_S^W T(t) - {}^W P(t) \right|}{{}^W P(t)} \cdot 100 \tag{7}$$

Once the tracking data refer to the same reference frame and time-synchronized, it is possible to perform a quantitative comparison in order to assess the tracking accuracy for each device. As evaluation metrics, the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE), as well as descriptive statistics, were adopted since they are the most commonly used metrics to compare trajectories with ground truth data, especially in the case of SLAM tracking algorithms [48–50].

In particular, the ATE standard metric evaluates the global consistency of tracking trajectories by comparing the absolute distance between the estimated tracking data, gathered from the built-in tracking system of the device, and the ground truth data provided by the external motion capture system. Once both trajectories are registered in the same coordinate system, the ATE is calculated as the root mean square error over all time indices of the translational components F_i as shown in the following equations (Equations (8) and (9)):

$$F_i = {}^{GT} T_i^{-1} * {}^{Device} T_i \tag{8}$$

and as a consequence:

$$\text{ATE} = \sqrt{\frac{1}{n} \sum_{1}^{n} ||trans(F_i)||^2} \tag{9}$$

Similarly, the RPE standard metric estimates the local accuracy of tracking in terms of relative drift error between an estimated trajectory and its ground truth over a fixed time interval. RPE is calculated as the root mean square error of translational or rotational components as shown in the following equation (Equation (10)):

$$\text{RPE} = \sqrt{\frac{1}{m} \sum_{1}^{m} ||trans(E_i)||^2} \tag{10}$$

where E_i is determined as follows (Equation (11)):

$$E_i = \left({}^{GT}T_i^{-1\,GT}T_{i+\Delta} \right)^{-1} * \left({}^{Device}T_i^{-1\,Device}T_{i+\Delta} \right) \tag{11}$$

In this experiment, only the translational components were considered as suggested by Sturm et al. [46] since rotational errors show up as translational errors during the movement of the device.

4. Results and Discussion

In this section, the results of the experimentation and are provided and discussed. In particular, the outcomes of the error metrics ATE and RPE are reported for each device. Descriptive statistics as well as one-way ANOVA analysis were used to assess the effects of different paths on the devices' tracking accuracy. In addition, an estimation of the reference points detected during the experiment was conducted for each path and device. All analyses were conducted by using the statistical package Microsoft Excel and IBM® SPSS. The statistical significance level was set at $p < 0.05$.

The following figure shows an example of trajectories covered by the iPad Pro considering a single lap over the eight-shaped (Figure 7a) and square-shaped (Figure 7b) paths.

Figures 8 and 9 show the averaged ATE and RPE values of each device for the three different paths. The vertical error bars represent the 95% confidence interval.

The results show different trends among the devices, and some of them showed different behavior when the path was changed. It is evident from both graphs that the Find X3 Lite device shows the highest errors for both metrics and for all the paths investigated. A one-way ANOVA analysis was carried out in order to investigate if each individual device behaved differently when the path was varied. The results, summarized in the following table (Table 3), confirmed that there are statistically significant differences, and thus some devices perform differently as the path changes.

Regarding the ATE metric, the analysis of variances revealed that six out of eight devices showed statistically significant differences among paths, and in particular, as confirmed by post hoc tests, between the eight-shaped trajectory and the other two paths, i.e., square- and U-shaped. A similar result was also obtained for the RPE metric for which it was found that there are four statistically significant differences between the performance obtained from the eight-shaped path and the other two ones. Furthermore, with regard to the square- and U-shaped paths, the results of the ANOVA post hoc tests show that there are no statistically significant differences in eight out of eight cases for the ATE metric, and in six out of eight cases for the RPE metric.

As a consequence, it is possible to assert that, on the whole, the mobile devices behave in the same way on the square- and U-shaped paths, and in addition, that overall, for both metrics, the majority of devices perform differently on the eight-shaped path compared to the other two ones. In particular, the results of a one-way ANOVA analysis carried out among the mobile devices given the same path, show that the path for which there are more statistically significant differences, and which therefore better highlights the differences

between devices, is the eight-shaped one. From these analyses, it emerged that only one device, i.e., iPad Pro, does not show statistically significant differences and it performs in the same manner on the different pathways.

Based on these considerations, the data analyses therefore focused on the eight-shaped path as it is the one that best highlights and compares the performance of the different mobile devices.

The following figure (Figure 10) shows the ATE values computed for the mobile devices in the case of the eight-shaped path.

(**a**)

(**b**)
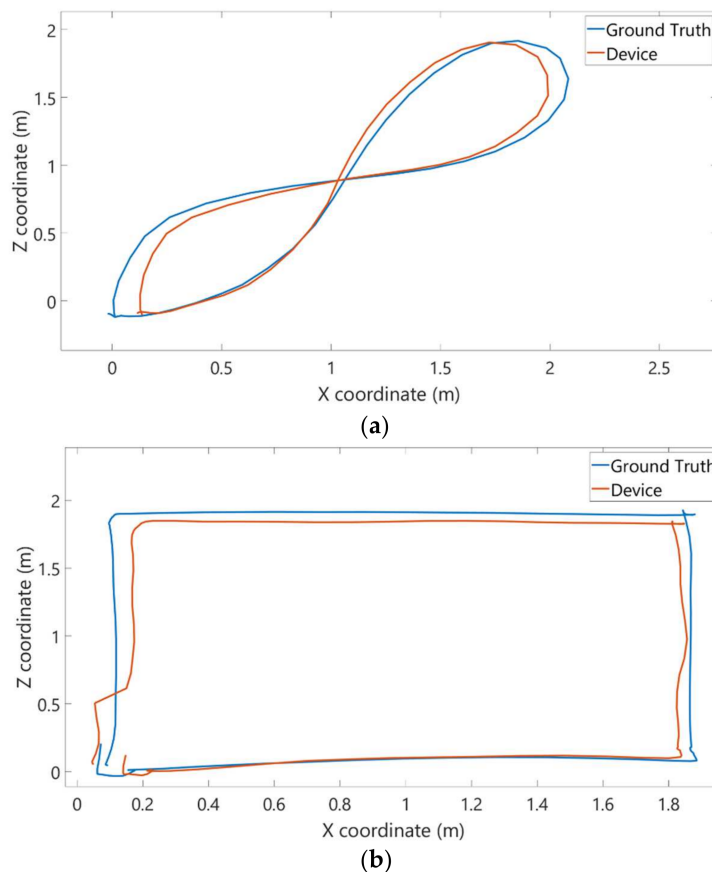
Figure 7. Examples of eight-shaped (**a**) and square-shaped (**b**) paths tracked by the motion capture system (blue line) and the corresponding aligned paths tracked by the selected device (red line).
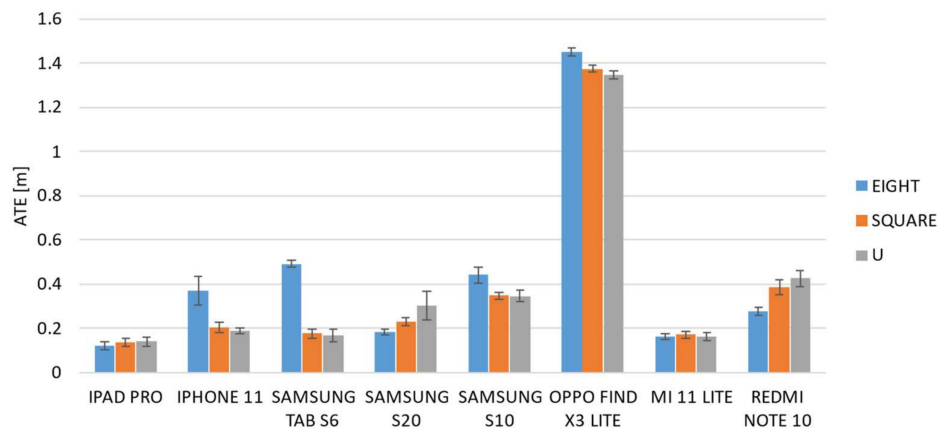

Figure 8. Averaged ATE for the selected devices over the three paths (eight-, square-, and U-shaped trajectories).
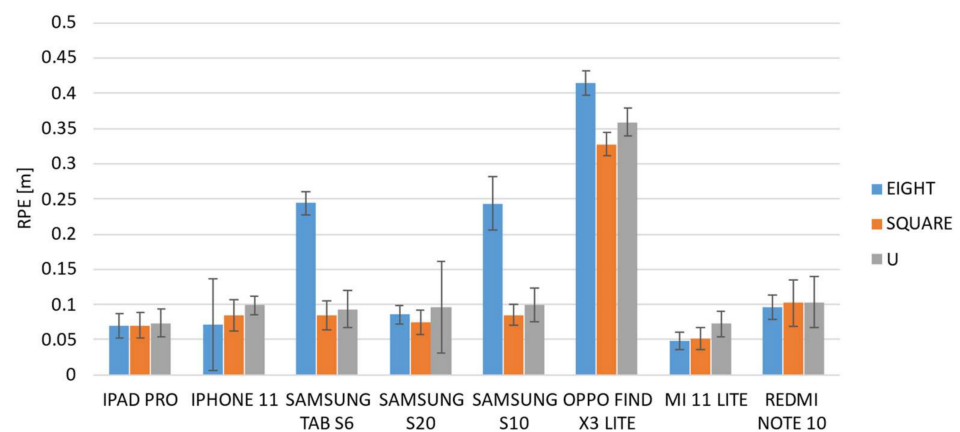
Figure 9. Averaged RPE for the selected devices over the three paths (eight-, square-, and U-shaped trajectories).

Table 3. Analysis of variances for ATE and RPE metrics over all paths for each device.

| Device	ATE		RPE	
	F-Value	**p-Value**	**F-Value**	**p-Value**
iPad Pro	0.810	0.456	0.111	0.896
iPhone 11	14.288	<0.001	21.126	<0.001
Tab S6	110.819	<0.001	304.782	<0.001
S20	13.880	<0.001	7.322	0.007
S10	9.672	0.002	180.569	<0.001
Find X3 Lite	35.436	<0.001	117.87	<0.001
Mi 11 Lite	0.425	0.658	16.328	<0.001
Redmi Note 10	27.101	<0.001	0.809	0.456
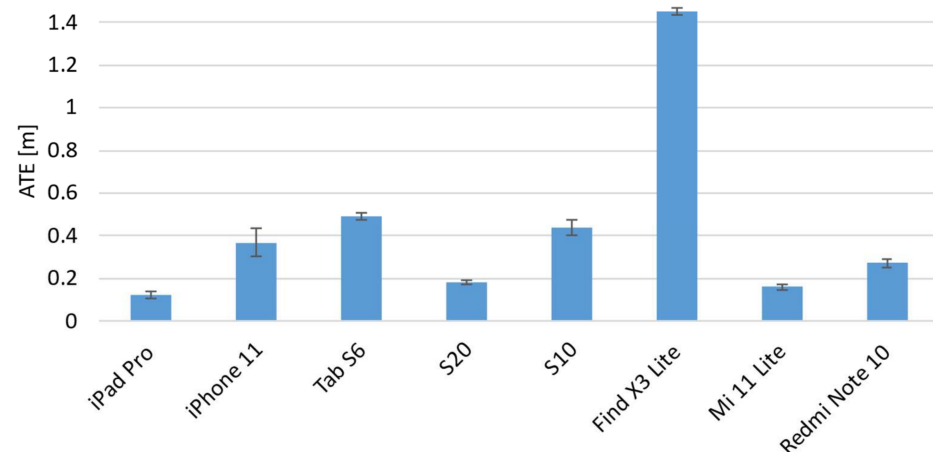
Figure 10. Absolute Trajectory Error (ATE) evaluation results related to the eight-shaped path.

The graph shows that the smallest ATE value was obtained by the iPad Pro with an average error of M = 0.121 m and SD of 0.027 m. This difference is statistically significant as confirmed by the one-way ANOVA test ($F_{(7)} = 815.38$) with respect to the other devices. However, the homogeneity test of variances was highly significant with a significance value of less than 0.05, and as a consequence, the null hypothesis was rejected. For this reason, the Welch version of the F-ratio was adopted, and it returned a value of $F_{(7, 30.62)} = 2476.94$ and $p < 0.001$. Following this, the group of devices that performed best in terms of the ATE metric, and showed no statistically significant differences between them were identified as the Xiaomi Mi 11 Lite and Samsung S20 with M = 0.162 m (SD = 0.019 m) and M = 0.182 m (SD = 0.020 m), respectively. Then, the third group was composed of the iPhone 11, S6, S10

and Note 10. Finally, as abovementioned, the OPPO Find X3 Lite showed the highest ATE error with M = 1.451 m and SD = 0.027 m.

Figure 11 shows the results obtained for the RPE metric related to the eight-shaped path.
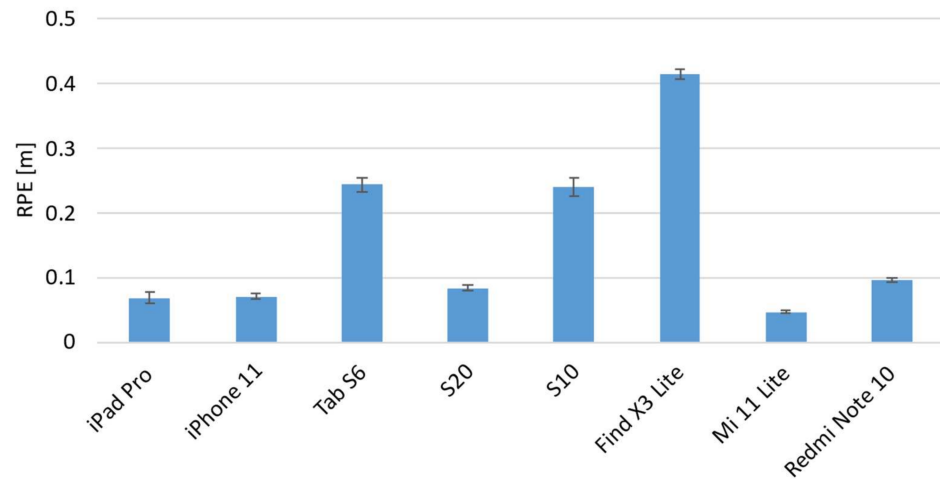
Figure 11. Relative Pose Error (RPE) evaluation results related to the eight-shaped path.

In this case, the Mi 11 Lite shows the lowest value of RPE with M = 0.048 m and SD = 0.004 m. In addition, a one-way ANOVA analysis indicated that there is a statistically significant difference with the RPE obtained from the other devices ($p < 0.001$). Additionally, the mobile devices that performed better in terms of the RPE metric, without any statistically significant difference among them, were the iPad Pro, iPhone 11 and S20 with mean values of M = 0.069 m (SD = 0.013 m), M = 0.072 m (SD = 0.007 m), and M = 0.085 m (SD = 0.008 m), respectively. The Tab S6 and S10 showed similar performances, without any statistically significant difference between them. In this case, the Find X3 Lite also provided the highest RPE error value with M = 0.414 m and SD = 0.012 m.

The following table (Table 4) depicts the mean, standard deviation, maximum, and minimum values for both ATE and RPE.

Table 4. Mean, standard deviation (SD), maximum (Max), and minimum (Min) of ATE and RPE values in the case of the eight-shaped path for all devices.

Device	ATE (m)				RPE (m)			
	Mean	SD	Max	Min	Mean	SD	Max	Min
iPad Pro	0.121	0.027	0.175	0.089	0.069	0.013	0.097	0.052
iPhone 11	0.369	0.105	0.514	0.221	0.072	0.007	0.085	0.064
Tab S6	0.491	0.026	0.547	0.453	0.244	0.018	0.266	0.208
S20	0.182	0.020	0.225	0.160	0.085	0.008	0.095	0.073
S10	0.437	0.059	0.562	0.376	0.241	0.022	0.272	0.204
Find X3 Lite	1.451	0.027	1.494	1.411	0.414	0.012	0.438	0.391
Mi 11 Lite	0.162	0.0196	0.202	0.138	0.048	0.004	0.053	0.041
Redmi Note 10	0.271	0.029	0.331	0.234	0.096	0.005	0.102	0.087

Other interesting outcomes that emerged from the experimentation are related to the feature points detected by the selected devices for the pose estimation. In fact, as stated in the introduction, SLAM algorithms combine inertial and visual-based tracking data in order to ensure the tracking is more robust and then they achieve high accuracy in pose estimation. In particular, in addition to the inertial data, visual natural feature points in the environment are recognized and extracted from the captured images, and then used to estimate the position and orientation of a device in the real world [51]. As a consequence, the device's ability to recognize and acquire the natural features of the environment is a key

aspect that can affect its tracking performance [52]. Therefore, the following table (Table 5) shows the average number of feature points detected by the selected devices on the three paths; specifically, the number of reference points detected while covering the same path ten times, in the same experimental environment and under the same lighting conditions.

Table 5. Averaged number of feature points detected by the selected devices in relation to the three paths.

Device	Detected Reference Points (#)		
	∞	▢	⊏
iPad Pro 11	33.233	31.061	2.069
iPhone 11	31.697	29.913	2.798
Tab S6	25.363	29.315	2.188
S20	31.865	31.043	3.559
S10	31.132	29.639	2.935
Find X3 Lite	29.662	26.812	3.865
MI 11 Lite	31.231	30.254	3.869
Redmi Note 10	31.671	30.672	3.740

The results show that the selected devices that run the AR application based on the two AR libraries, i.e., ARKit and ARCore, perform in a similar way. In fact, the reference points detected and extracted from the environment were comparable in relation to the path for the different devices. This means that the AR libraries perform in a similar way and no differences could be observed during their functioning.

In summary, some interesting outcomes emerged from the analysis of the results shown above. With regard to the examined paths, which were both open and closed paths, the eight-shaped path was the one that better emphasized the differences in terms of the tracking performance of the mobile devices. This outcome made it possible to focus the analysis of the results on this specific pathway. This also provides guidelines on the choice of pathways to be used in experiments for tracking performance evaluation. The only device that performed the same, regardless of the path, was the iPad Pro. This could be due to the fact that it is equipped with a LIDAR sensor that improves the tracking accuracy. In this regard, it is also important to consider its price, which is in the high-cost range.

With regard to square-shaped and U-shaped paths, the selected mobile devices showed similar results on the whole. This was also confirmed by analysis of variances, which did not reveal any statistically relevant difference. Thus, this suggests that it might be sufficient to conduct tests using short U-shaped paths to evaluate device pose tracking quality without wasting time covering a long square-shaped path.

In general, considering both the ATE and RPE metrics, the iPad Pro and Xiaomi Mi 11 Lite performed the best. They were immediately followed by the S20 while the Find X3 Lite device showed the highest errors for both metrics and for all the paths investigated.

When the SDKs were taken into consideration, both ARKit and ARCore seem to perform in a similar way since the tracking data provided by different devices that implement these frameworks did not show significant relevant differences.

While from a merely economic point of view, both mid- and high-cost devices such as the iPhone 11 and the S20, show similar performances it may not be necessary to use expensive devices for applications that have tracking capabilities.

It is important to note that the presented outcomes are limited to the systems and pathways examined, and relate to tests undertaken in a laboratory. Therefore, future developments will involve testing such mobile devices on different paths, both in the laboratory and in open air, and they will also involve users to simulate different conditions of use.

5. Conclusions

In this work, a benchmark evaluation of the tracking performance of mid- to high-end popular devices has been presented. Eight different smartphones and tablets were selected from the market with the aim of ensuring a representative set of mobile devices. For this reason, both iOS® and Android™ devices were taken into account by considering several aspects such as the release date, price range, platform, and other specific parameters related to the hardware components. A methodology was developed for the estimation of their tracking capabilities; this consisted of three main stages: calibration, data acquisition, and data analysis, which allowed a comparison of the estimated tracking data from the device with the ground truth. An external motion capture system was adopted to capture the movements of mobile devices and to obtain the ground truth for data comparison and analysis. A series of experiments were conducted in a laboratory by investigating the mobile devices' performances on open and closed paths. These performances were evaluated in terms of two standard error metrics, and specifically, ATE and RPE metrics. The results demonstrated that the devices performed in different ways over the three different paths, except for the iPad Pro, which showed similar results for all trajectories. In fact, the selected devices, in combination with the AR SDKs, showed different tracking performances, which does not allow for a definitive classification in terms of tracking performance, but still provide suggestions and guidelines for conducting similar experiments in the future. Furthermore, future work will investigate more paths in different environmental conditions in order to assess how external parameters can affect the tracking accuracy of mobile devices. In particular, a series of experiments will be carried out with different illumination settings to assess how the lighting of the scene can affect the tracking accuracy of the selected devices.

Nomenclature

AR	Augmented Reality
SDK	Software Development Kit
SLAM	Simultaneous localization and mapping
LIDAR	Laser Imaging Detection and Ranging
ATE	Absolute Trajectory Error
RPE	Relative Pose Error

References

1. Liberatore, M.J.; Wagner, W.P. Virtual, mixed, and augmented reality: A systematic review for immersive systems research. *Virtual Real.* **2021**, *25*, 773–799. [CrossRef]
2. Kim, K.; Billinghurst, M.; Bruder, G.; Duh, H.B.L.; Welch, G.F. Revisiting trends in augmented reality research: A review of the 2nd decade of ISMAR (2008–2017). *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 2947–2962. [CrossRef] [PubMed]

3. Sirohi, P.; Agarwal, A.; Maheshwari, P. A survey on Augmented Virtual Reality: Applications and Future Directions. In Proceedings of the 2020 Seventh International Conference on Information Technology Trends (ITT), Abu Dhabi, United Arab Emirates, 25–26 November 2020; pp. 99–106.

4. Minaee, S.; Liang, X.; Yan, S. Modern Augmented Reality: Applications, Trends, and Future Directions. *arXiv* **2022**, arXiv:2202.09450.

5. Parekh, P.; Patel, S.; Patel, N.; Shah, M. Systematic review and meta-analysis of augmented reality in medicine, retail, and games. *Vis. Comput. Ind. Biomed. Art* **2022**, 3, 21. [CrossRef]

6. Chatzopoulos, D.; Bermejo, C.; Huang, Z.; Hui, P. Mobile augmented reality survey: From where we are to where we go. *IEEE Access* **2017**, 5, 6917–6950. [CrossRef]

7. Masood, T.; Egger, J. Adopting augmented reality in the age of industrial digitalization. *Comput. Ind.* **2020**, 115, 103112. [CrossRef]

8. Bekele, M.K.; Pierdicca, R.; Frontoni, E.; Malinverni, E.S.; Gain, J. A survey of augmented, virtual, and mixed reality for cultural heritage. *J. Comput. Cult. Herit.* **2018**, 11, 1–36. [CrossRef]

9. Costa, G.D.M.; Petry, M.R.; Moreira, A.P. Augmented Reality for Human–Robot Collaboration and Cooperation in Industrial Applications: A Systematic Literature Review. *Sensors* **2022**, 22, 2725. [CrossRef]

10. Garzón, J.; Pavón, J.; Baldiris, S. Systematic review and meta-analysis of augmented reality in educational settings. *Virtual Real.* **2019**, 23, 447–459. [CrossRef]

11. De Pace, F.; Manuri, F.; Sanna, A. Augmented reality in industry 4.0. *Am. J. Comput. Sci. Inf. Technol.* **2018**, 6, 17. [CrossRef]

12. Fraga-Lamas, P.; Fernández-Caramés, T.M.; Blanco-Novoa, O.; Vilar-Montesinos, M.A. A review on industrial augmented reality systems for the industry 4.0 shipyard. *IEEE Access* **2018**, 6, 13358–13375. [CrossRef]

13. Reljić, V.; Milenković, I.; Dudić, S.; Šulc, J.; Bajči, B. Augmented reality applications in industry 4.0 environment. *Appl. Sci.* **2021**, 11, 5592. [CrossRef]

14. Erdem, A.T.; Ercan, A.Ö. Ercan. Fusing Inertial Sensor Data in an Extended Kalman Filter for 3D Camera Tracking. *IEEE Trans. Image Process.* **2015**, 24, 538–548. [CrossRef] [PubMed]

15. Palmarini, R.; Erkoyuncu, J.A.; Roy, R.; Torabmostaedi, H. A systematic review of augmented reality applications in maintenance. *Robot. Comput. -Integr. Manuf.* **2018**, 49, 215–228. [CrossRef]

16. Danielsson, O.; Holm, M.; Syberfeldt, A. Augmented reality smart glasses in industrial assembly: Current status and future challenges. *J. Ind. Inf. Integr.* **2020**, 20, 100175. [CrossRef]

17. Roberto, R.; Lima, J.P.; Teichrieb, V. Tracking for mobile devices: A systematic mapping study. *Comput. Graph.* **2016**, 56, 20–30. [CrossRef]

18. de Souza Cardoso, L.F.; Mariano, F.C.M.Q.; Zorzal, E.R. A survey of industrial augmented reality. *Comput. Ind. Eng.* **2020**, 139, 106159. [CrossRef]

19. Sadeghi-Niaraki, A.; Choi, S.M. A survey of marker-less tracking and registration techniques for health & environmental applications to augmented reality and ubiquitous geospatial information Systems. *Sensors* **2020**, 20, 2997.

20. Gupta, S.; Chaudhary, R.; Kaur, A.; Mantri, A. A survey on tracking techniques in augmented reality-based application. In Proceedings of the 2019 Fifth International Conference on Image Information Processing (ICIIP), Shimla, India, 15–17 November 2019; pp. 215–220.

21. Debeunne, C.; Vivet, D. A review of visual-LiDAR fusion based simultaneous localization and mapping. *Sensors* **2020**, 20, 2068. [CrossRef]

22. Nowacki, P.; Woda, M. Capabilities of arcore and arkit platforms for ar/vr applications. In Proceedings of the International Conference on Dependability and Complex Systems, Brunów, Poland, 1–5 July 2019; Springer: Cham, Switzerland; pp. 358–370.

23. Cao, J.; Lam, K.Y.; Lee, L.H.; Liu, X.; Hui, P.; Su, X. Mobile Augmented Reality: User Interfaces, Frameworks, and Intelligence. *arXiv* **2021**, arXiv:2106.08710.

24. Diao, P.H.; Shih, N.J. MARINS: A mobile smartphone AR system for pathfinding in a dark environment. *Sensors* **2018**, 18, 3442. [CrossRef]

25. Shih, N.J.; Diao, P.H.; Chen, Y. ARTS, an AR tourism system, for the integration of 3D scanning and smartphone AR in cultural heritage tourism and pedagogy. *Sensors* **2019**, 19, 3725. [CrossRef] [PubMed]

26. Wang, J.; Qi, Y. A Multi-User Collaborative AR System for Industrial Applications. *Sensors* **2022**, 22, 1319. [CrossRef]

27. Li, X.; Tian, Y.; Zhang, F.; Quan, S.; Xu, Y. Object detection in the context of mobile augmented reality. In Proceedings of the 2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Porto de Galinhas, Brazil, 9–13 November 2020; pp. 156–163.

28. Ye, H.; Kwan, K.C.; Su, W.; Fu, H. ARAnimator: In-situ character animation in mobile AR with user-defined motion gestures. *ACM Trans. Graph.* **2020**, 39, 83. [CrossRef]

29. Monica, R.; Aleotti, J. Evaluation of the Oculus Rift S tracking system in room scale virtual reality. *Virtual Real.* **2022**, 1–11. [CrossRef]

30. Soares, I.; BSousa, R.; Petry, M.; Moreira, A.P. Accuracy and Repeatability Tests on HoloLens 2 and HTC Vive. *Multimodal Technol. Interact.* **2021**, 5, 47. [CrossRef]

31. Groves, L.A.; Carnahan, P.; Allen, D.R.; Adam, R.; Peters, T.M.; Chen, E. Accuracy assessment for the co-registration between optical and VIVE head-mounted display tracking. *Int. J. Comput. Assist. Radiol. Surg.* **2019**, 14, 1207–1215. [CrossRef]

32. Li, J.; Slembrouck, M.; Deboeverie, F.; Bernardos, A.M.; Besada, J.A.; Veelaert, P.; Casar, J.R. A hybrid pose tracking approach for handheld augmented reality. In Proceedings of the 9th International Conference on Distributed Smart Cameras, Seville, Spain, 8–11 September 2015; pp. 7–12.

33. Marino, E.; Barbieri, L.; Colacino, B.; Fleri, A.K.; Bruno, F. An Augmented Reality inspection tool to support workers in Industry 4.0 environments. *Comput. Ind.* **2021**, *127*, 103412. [CrossRef]

34. Bruno, F.; Barbieri, L.; Marino, E.; Muzzupappa, M.; D'Oriano, L.; Colacino, B. An augmented reality tool to detect and annotate design variations in an Industry 4.0 approach. *Int. J. Adv. Manuf. Technol.* **2019**, *105*, 875–887. [CrossRef]

35. Bruno, F.; Barbieri, L.; Marino, E.; Muzzupappa, M.; Colacino, B. A Handheld Mobile Augmented Reality Tool for On-Site Piping Assembly Inspection. In *International Conference of the Italian Association of Design Methods and Tools for Industrial Engineering*; Springer: Cham, Switzerland, 2019; pp. 129–139.

36. Feigl, T.; Porada, A.; Steiner, S.; Löffler, C.; Mutschler, C.; Philippsen, M. Localization Limitations of Arcore, Arkit, and Hololens in Dynamic Large-scale Industry Environments. In Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (GRAPP), Valletta, Malta, 27–29 February 2020; pp. 307–318.

37. Hasler, O.; Blaser, S.; Nebiker, S. Implementation and first evaluation of an indoor mapping application using smartphones and AR frameworks. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *XLII-2/W17*, 135–141. [CrossRef]

38. Hübner, P.; Clintworth, K.; Liu, Q.; Weinmann, M.; Wursthorn, S. Evaluation of hololens tracking and depth sensing for indoor mapping applications. *Sensors* **2020**, *20*, 1021. [CrossRef] [PubMed]

39. Cortés, S.; Solin, A.; Rahtu, R.; Kannala, J. Advio: An authentic dataset for visual-inertial odometry. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 419–434.

40. Aurand, A.M.; Dufour, J.S.; Marras, W.S. Accuracy map of an optical motion capture system with 42 or 21 cameras in a large measurement volume. *J. Biomech.* **2017**, *58*, 237–240. [CrossRef] [PubMed]

41. Chen, Q.; Zhou, Y.; Wang, Y.; Zhu, M.M.; Guo, L.; He, C.X. Research on stability and accuracy of the OptiTrack system based on mean error. In Proceedings of the International Symposium on Artificial Intelligence and Robotics 2021, Fukuoka, Japan, 28 October 2021; Volume 11884, pp. 438–450.

42. Nagymáté, G.; Kiss, R.M. Application of OptiTrack motion capture systems in human movement analysis: A systematic literature review. *Recent Innov. Mechatron.* **2018**, *5*, 1–9.

43. Nezami, F.N.; Wächter, M.A.; Maleki, N.; Spaniol, P.; Kühne, L.M.; Haas, A.; Pingel, J.; Tiemann, L.; Nienhaus, F.; Keller, L.; et al. Westdrive X LoopAR: An Open-Access Virtual Reality Project in Unity for Evaluating User Interaction Methods during Takeover Requests. *Sensors* **2021**, *21*, 1879. [CrossRef] [PubMed]

44. Ortiz-Fernandez, L.E.; Cabrera-Avila, E.V.; da Silva, B.M.; Gonçalves, L.M. Smart artificial markers for accurate visual mapping and localization. *Sensors* **2021**, *21*, 625. [CrossRef]

45. Chai, W.; Li, C.; Zhang, M.; Sun, Z.; Yuan, H.; Lin, F.; Li, Q. An enhanced pedestrian visual-inertial SLAM system aided with vanishing point in indoor environments. *Sensors* **2021**, *21*, 7428. [CrossRef]

46. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 573–580.

47. Seiskari, O.; Rantalankila, P.; Kannala, J.; Ylilammi, J.; Rahtu, E.; Solin, A. HybVIO: Pushing the Limits of Real-time Visual-inertial Odometry. In Proceedings of the IEEE/CVF Winter onference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 701–710.

48. Di, K.; Zhao, Q.; Wan, W.; Wang, Y.; Gao, Y. RGB-D SLAM based on extended bundle adjustment with 2D and 3D information. *Sensors* **2016**, *16*, 1285. [CrossRef]

49. Newman, P.; Ho, K. SLAM-loop closing with visually salient features. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 635–642.

50. Mahon, I.; Williams, S.B.; Pizarro, O.; Johnson-Roberson, M. Efficient view-based SLAM using visual loop closures. *IEEE Trans. Robot.* **2008**, *24*, 1002–1014. [CrossRef]

51. Oufqir, Z.; El Abderrahmani, A.; Satori, K. ARKit and ARCore in serve to augmented reality. In Proceedings of the 2020 International Conference on Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 9–11 June 2020; pp. 1–7.

52. Torresani, A.; Rigon, S.; Farella, E.M.; Menna, F.; Remondino, F. Unveiling large-scale historical contents with V-SLAM and markerless mobile AR solutions. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *46*, 761–768. [CrossRef]