

RESEARCH ARTICLE

A Multifactor Comparative Assessment of Augmented Reality Frameworks in Diverse Computing Settings

MFUNDO A. MANELI^{ID} AND OMOWUNMI E. ISAFIADI^{ID}

Department of Computer Science, Faculty of Natural Science, University of the Western Cape, Bellville, Cape Town 7535, South Africa

Corresponding author: Omowunmi E. Isafiade (oisafiade@uwc.ac.za)

This work was supported in part by the University of the Western Cape (UWC), and in part by the Open Serve/Telkom Centre of Excellence (CoE).

ABSTRACT Research and development on different augmented reality (AR) frameworks have come a long way when it comes to image tracking, object tracking, plane tracking and light estimation. However, there might be trade-offs and varying results obtained from different AR frameworks, depending on the use cases, and this is critical for consideration during immersive application development. Besides the current literature effort, this research proposes a multifactor comparative analysis of two core AR frameworks, which aims to analyze and evaluate ARKit and ARCore in diverse computing settings. This research developed a structural application which evaluated three major test parameters across ten devices spanning ARKit and ARCore. The first parameter relates to evaluating AR measurements using four different distance criteria. The second parameter evaluated resource utilization, relating to the central processing unit (CPU) and random access memory (RAM), while the last parameter evaluated plane detection based on light estimation. Findings conclude that ARKit is the preferable AR framework for AR measurement accuracy and reliability within the tested distance criteria. ARCore is the most optimized AR framework in terms of RAM utilization. Regarding plane detection based on light estimation, ARCore is the preferable choice under low lighting conditions, however, ARKit is the most suitable AR framework under adequate ambient lighting conditions. The findings of this research could guide future prototyping and immersive mobile application development within the context of the parameters used.

INDEX TERMS Augmented reality, ARCore, ARKit, AR measurements, plane detection, light estimation, resource utilization.

I. INTRODUCTION

The relevance of immersive technologies across different domains is evident in the different use cases that have emerged over the years, which includes marketing [1], education [2], agriculture [3], construction [4] and forensic science [5], to mention a few. It is important to note that the error margins and trade-offs that can be tolerated in the immersive applications developed across these domains may vary. This varying level of tolerance is due to the fact that the impact of an error in an immersive application could be more grave in certain domains than in others, for example, in construction [6] and forensic science [7] as opposed to

advertising [1]. In advertising, the main objective is often to display the features of certain products or services and increase brand awareness [1], while in forensic science there is a higher level of sensitivity with respect to accuracy, and how the information presented to the court will inform the jury's decision as presented in Figure 1.

Accuracy and reliability are the core fundamentals in ensuring that credible crime scene measurement data has been captured. In crime scene data capturing, investigators must capture data without the risk of contaminating the scene [5]. Immersive technologies could enable a crime investigator to accurately capture measurement data without the risk of scene contamination. These technologies encompass virtual reality (VR), augmented reality (AR), mixed reality (MR) and extended reality (XR). AR and VR are the primary

The associate editor coordinating the review of this manuscript and approving it for publication was Andrea F. Abate^{ID}.



FIGURE 1. Schematic of immersive Jury's decision support system.

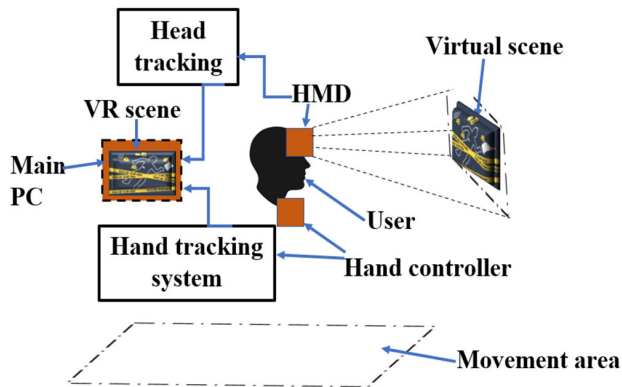


FIGURE 2. VR schematic diagram.

technologies under immersive technologies, whilst MR and XR are a combination of AR and VR. VR in general offers three-dimensional (3D) computer generated environments, immersive, and multisensory experience [8], which relies on 3D, stereoscopic, head tracked displays, hand/body tracking and binaural sound as shown in Figure 2. VR can be broken up into the following categories [9]:

- Non-immersive – A user maintains full awareness of the physical environment outside of the VR
- Semi-immersive – More senses are immersed compared to non-immersive, but not all.
- Fully-immersive – stimulates all the user's senses.

AR is a technology which can superimpose digital perceptual information in the real world [10]. AR can also be categorized into the following as described in Table 1.

Figure 3, provides a schematic diagram example of fully imposed AR [15].

Prior implementations of AR/VR applications required bulky, computationally demanding and costly gear to function [16]. With further iterations in research and development, low-powered hand-held mobile devices (e.g smartphones) and headsets are able to run AR and VR applications [17]. The mobile operating system sector is primarily dominated by Android and iOS as shown in Figure 4. Android is an open-source mobile operating system utilized by a vast majority of smartphone companies, this operating system is developed, supported and owned by Google [18]. iOS is developed, supported and owned by Apple; it is not open source nor is the operating system utilized by any other company besides Apple [18]. Both Google and Apple developed

TABLE 1. AR categories and description.

S/N	Type	Description	Example
1	Marker-based	Relies on an image target to function	QR code marker [11], [12]
2	Markerless	Relies on technology such as global positioning system (GPS) to provide results	GPS [11] Google Maps [10], [13]
3	Partially/Fully superimposed	Enables a user to partially or fully project virtual content onto the real world	Virtually insert furniture into a real home space [11]
4	Projection-based	Relies on different lighting sources such as lasers to produce a hologram-like output	TeleAdvisor [14]

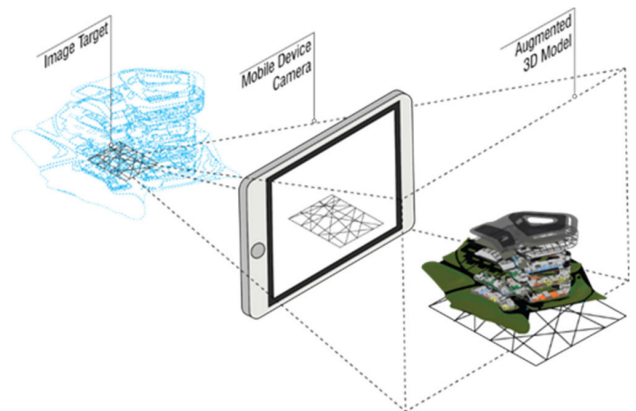


FIGURE 3. AR schematic diagram.

frameworks, which offer AR features [19], [20]. Google offers ARCore while Apple offers ARKit, both AR frameworks offer features which may achieve similar goals as shown in Table 2. However, the same level of accuracy may not necessarily always be achieved by both frameworks in certain applications.

Table 2 provides a summary of the supported features in both ARCore and ARKit software development kit (SDK), which is applicable in this research. Features which are supported by either AR framework are indicated with a tick symbol, while those not supported are not filled (i.e blank). Furthermore, it is important to note that in certain domains that are seeking to integrate AR, accuracy deviations may be more severe, such as in forensic science where minor deviations in the information presented to a jury (see Figure 1) could wrongly inform a hypothesis about a crime scene, possibly rule off an entire case, or even lead to a death sentence.

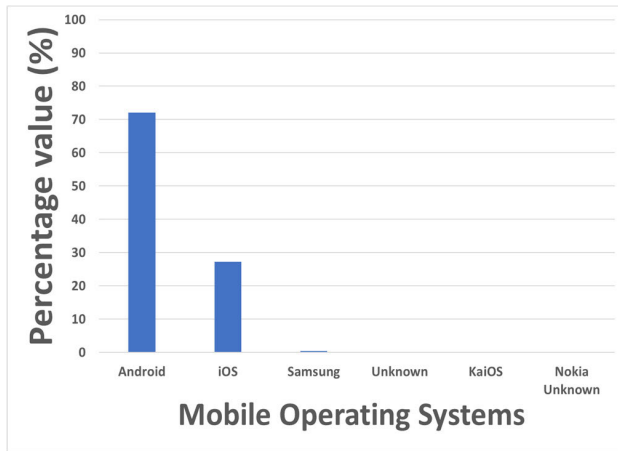


FIGURE 4. Mobile operating systems used globally [17].

TABLE 2. Summary of supported features in both frameworks.

Supported Feature	ARCore SDK (Unity)	ARKit SDK (Unity)
Device tracking	☑	☑
2D Image tracking	☑	☑
3D Object tracking		☑
Occlusion	☑	☑
LiDAR meshing		☑
Raycast	☑	☑
Plane tracking	☑	☑
Anchors	☑	☑
Light estimation	☑	☑

In light of the above, it is instructive to understand how these frameworks may fare under different conditions and across varying mobile devices. This paper introduced a four-distance criteria measure across ten devices spanning ARKit and ARCore. Furthermore, it compared device system utilization, accuracy, runtimes for vertical and horizontal plane detections within a given parameter and the impacts of environmental light estimation on the outcome of the two frameworks. The aim is to explore accurate augmentation capabilities and provide guidance in diverse immersive systems development. There is limited research that has been conducted on comparing these two core AR frameworks especially with respect to AR accuracy measurements using different parameters across ten devices as considered in this research. This has motivated the need for this research, and could assist developers in understanding the potential trade-offs at stake during prototyping and development of mobile immersive applications.

To the best of the researchers’ knowledge, while there are research efforts on comparing these two frameworks [20], [22], [23], there is no study that has considered a multifactor analysis approach as proposed in the current research.

The remainder of this paper is structured as follows: Section II presents the related research and further justifies the novelty of our research. Section III presents the research methodology. Results are presented in Section III-B, while Section IV concludes the paper.

II. RELATED WORK

This section discusses some of the previous research efforts that relate to the comparison of ARKit and ARCore frameworks in diverse computing applications.

Nowacki et al. [22] scrutinize and compare ARCore and ARKit platforms for AR/VR applications. The aim was to evaluate the capabilities of ARKit and ARCore in determining which AR framework is applicable in speeding up prototyping and development of modern AR/VR applications. General performance in terms of time to load models, initialize run applications and cameras was measured. They also compared CPU loads and memory usage. Their work however only utilized obsolete devices and AR framework versions for the testing, thus rendering the results as outdated, compared to the current study. Their research also did not compare the accuracy measurements between ARKit and ARCore as done in this paper.

Chaudhry et al. [23] present an overview of ARFoundation and the ARCore SDK library. The aim of this research was to demonstrate the capabilities of ARCore regarding plane detection, light estimation and efficiency in terms of time taken to detect planes. Even though this research paper caters for plane detection, there is no mention as to how plane detection is affected under different lighting conditions. A point to also note is that this research paper only used a single mobile device in demonstrating these capabilities. However, in the current research ten different mobile devices, with five spanning each of the core frameworks, were utilized for testing the different parameters considered.

Cervenak et al. [20] describe the possibilities of indoor space mapping and user movement tracking using AR as well as incorporating accelerometers, gyroscopes and magnetometer sensors to acquire relative device positioning. The aim was to evaluate the possibilities of ARKit and ARCore with respect to movement in space, without using other navigation technologies. This study however based its findings on the use of ARKit 3 and ARKit 2 which are respectively three and four generations behind from the current version, which is ARKit 6.

Borduas et al. [19] present a study which provides insight and recommendations on which of the four mobile 3D scanning technologies (i.e., ARKit, ARCore, ScandyPro and the 3DSizeMe app) are reliable and sufficient for respiratory face mask customization. The aim is to compare the reliability of ARCore, ARKit, ScandyPro and the 3DSizeMe app using the

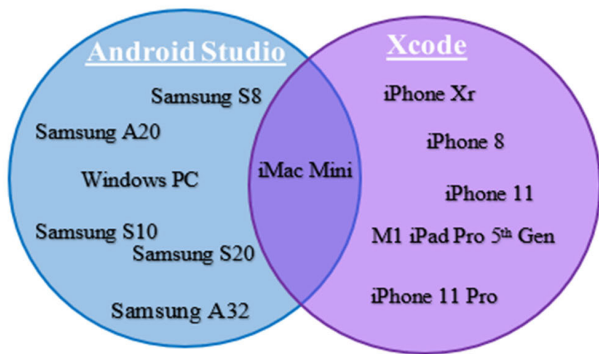


FIGURE 5. Software and device compatibility.

Structure Sensor by Occipital. Findings show high errors with the ScandyPro scanning technology, suggesting that either a noisier scan or greater variability between scans is observed. ARKit, ARCore and 32SizeMe showed acceptable margins of error.

Oufqir et al. [24] present a study which evaluates the functionalities of two AR frameworks, ARKit and ARCore. This study evaluates how the two AR frameworks function, depending on the technology being used to detect and track objects, points, and features in a given scene. The aim of this study was twofold. First, it was to evaluate the capabilities of the two ARKit and ARCore libraries. Second, to evaluate their capabilities in the development of AR applications. This paper, however, did not evaluate the time taken to detect planes nor the comparison of augmentation under different lighting conditions as done in the current research.

Fabricao et al. [25] present a comparative analysis of existing AR frameworks that may allow for educational solutions. The objective of the study was to compare the characteristics of existing frameworks that may allow for the development of educational AR solutions to assist educators in their classrooms. The research focused on which AR framework would be the best at application development, however they never compared how well these AR frameworks fared in terms of accuracy. The versions of ARKit and ARCore used when conducting this research are also now obsolete.

Carranza et al. [26] propose an AR plane detection application which aims to tag detected objects by means of labeling them and relaying said tagged information to visually impaired users. However, this study did not compare plane detection accuracy spanning multiple devices or how plane detections are handled under different lighting conditions as done in the current research.

While the aforementioned research efforts have conducted monumental research on comparing the two different AR frameworks (ARKit and ARCore), none of these research studies has considered the accuracy of AR measurements with four distance criteria, resource utilization as well as run-

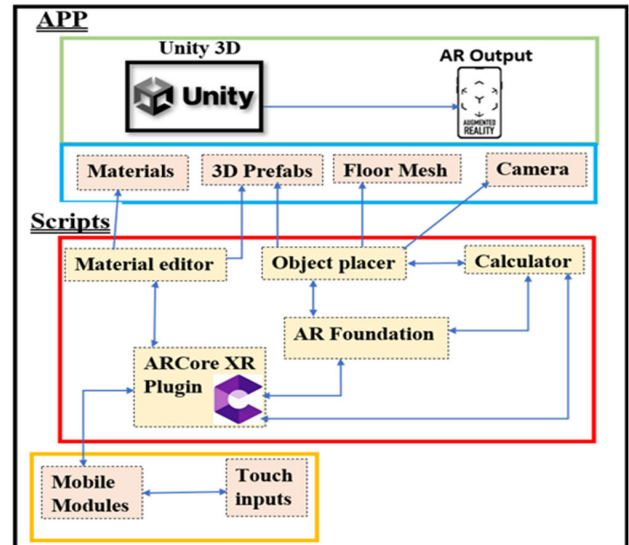


FIGURE 6. ARCore application layout.

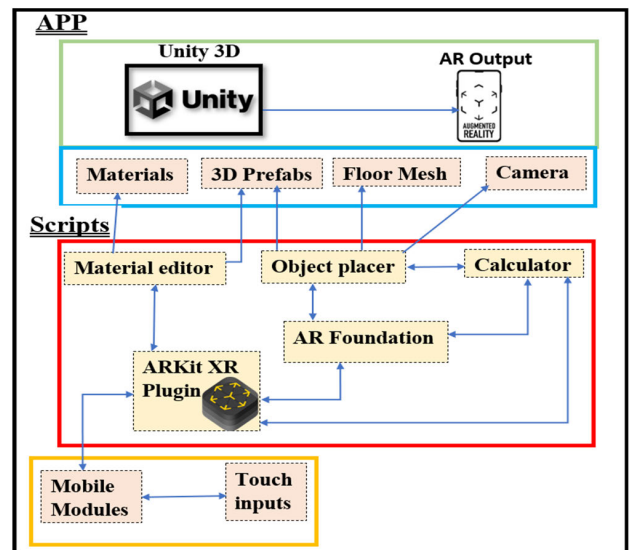


FIGURE 7. ARKit application layout.

time of plane detection and mapping under different ambient lighting conditions, using several different mobile devices as presented in this research. In the previous iteration of this work [27], we looked at the two dominating mobile operating systems and compared the AR frameworks they operate on. ARKit and ARCore were the two AR frameworks that were compared, but the comparison was only based on accuracy of AR measurements. Six mobile devices were used to conduct the prior study, in which five devices represented ARCore and one device was used to represent ARKit. The reason for this offset device selection is due to the fact that the ARKit device houses a LiDAR scanner and has the best configurations for dynamic ranging which all the other ARCore devices do not possess. Results from this study show that ARKit had the highest accuracy measurement average of 99.36% whereas ARCore had an accuracy average of 89.42%.

TABLE 3. Device system specifications.

Device	Processor ID	Core count	Processor architecture	RAM capacity (GB)
iPhone Xr	A12 Bionic	6	x64	3GB
iPhone 11 Pro	A13Bionic	6	x64	4GB
iPhone 8	A11Bionic	6	x64	2GB
iPhone 11	A13Bionic	6	x64	4GB
M1 iPad Pro 5 th Gen	M1	8	x64	8GB
Samsung S8	Exynos 8895	8	x64	4GB
Samsung S20	Exynos 990	8	x64	8GB
Samsung S10	Exynos 9820	8	x64	8GB
Samsung A20	Exynos 7884	8	x64	3GB
Samsung A32	Helio G80	8	x64	4GB

In this new study, we added on more ARKit devices to balance out the ratio between ARKit and ARCore devices and also used multiple parameters to present a rigorous assessment of these AR frameworks. We added more test parameters such as acquiring AR measurements using four distance criteria across more devices, run time, system utilization checks, horizontal and vertical plane detection and mapping, so as to guide future immersive development.

III. DESIGN AND TESTING METHODOLOGY

A. TEST SETUP

For this research, two mobile applications inspired by the AR Ruler application found on the google play store were developed, based on the latest versions of ARCore and ARKit, to determine plane detection accuracy and reliability in augmented reality solutions. Ten mobile devices were used in the testing, with five representing ARCore and the other five working with ARKit. The devices representing ARKit are the iPhones 8, Xr, 11, 11 Pro and the M1 iPad Pro 5th Gen. The devices representing ARCore are the following: Samsung A20, A32, S8, S10 and S20. Both applications were developed in Unity version 2022.1.22 using the latest ARKit and ARCore SDK plugins versions 5.0.2. AR Foundation version 5.0.2 was also utilized for the plane detection section. Visual studio 2021 was utilized for the backend C# coding, for the functionality such as the calculations for both applications. The application for the devices running ARCore was developed on a 6 core, 12 thread Ryzen 5 1600, 32GB of RAM running Windows 10 Pro version 22H2. The ARKit variant of the application was developed using a 6 core, 6 thread Intel (R) Core i5, 8GB of RAM Apple mac Mini running mac OS Monterey. Xcode was also used to deploy and monitor all ARKit devices for RAM and CPU usage. Android studio was

TABLE 4. Camera configuration of each device.

Device	Camera specifications
Samsung S8	12 MP, f/1.7, 26mm (wide), 1/2.55", 1.4µm, dual pixel PDAF, OIS
Samsung S10	12 MP, f/1.5-2.4, 26mm (wide), 1/2.55", 1.4µm, Dual Pixel PDAF, OIS 12 MP, f/2.4, 52mm (telephoto), 1/3.6", 1.0µm, AF, OIS, 2x optical zoom 16 MP, f/2.2, 12mm (ultrawide), 1/3.1", 1.0µm, Super Steady video
Samsung S20	12 MP, f/1.8, 26mm (wide), 1/1.76", 1.8µm, Dual Pixel PDAF, OIS 64 MP, f/2.0, 29mm (telephoto), 1/1.72", 0.8µm, PDAF, OIS, 1.1x optical zoom, 3x hybrid zoom 12 MP, f/2.2, 13mm, 120° (ultrawide), 1/2.55" 1.4µm, Super Steady video
Samsung A20	13 MP, f/1.9, 28mm (wide), AF 5 MP, f/2.2, 12mm (ultrawide)
Samsung A32	64 MP, f/1.8, 26mm (wide), PDAF 8 MP, f/2.2, 123°, (ultrawide), 1/4.0", 1.12µm 5 MP, f/2.4, (macro) 5 MP, f/2.4, (depth)
M1 iPad Pro 5 th Gen	12 MP, f/1.8, (wide), 1/3", 1.22µm, dual pixel PDAF 10 MP, f/2.4, 125° (ultrawide) TOF 3D LiDAR scanner (depth)
iPhone Xr	12 MP, f/1.8, 26mm (wide), 1/2.55", 1.4µm, PDAF, OIS
iPhone 11 pro	12 MP, f/1.8, 26mm (wide), 1/2.55", 1.4µm, dual pixel PDAF, OIS 12 MP, f/2.0, 52mm (telephoto), 1/3.4", 1.0µm, PDAF, OIS, 2x optical zoom 12 MP, f/2.4, 120°, 13mm (ultrawide), 1/3.6"
iPhone 8	12 MP, f/1.8, 28mm (wide), PDAF, OIS
iPhone 11	12 MP, f/1.8, 26mm (wide), 1/2.55", 1.4µm, dual pixel PDAF, OIS 12 MP, f/2.4, 120°, 13mm (ultrawide), 1/3.6"

used to monitor all the ARCore running devices. The reason why we had to use two different applications (Xcode and Android studio) to monitor the different ARKit and ARCore devices is due to the fact that Xcode is only available for MacOS running machines. Figure 5 provides a compatibility illustration stating which devices are compatible with which software.

Figure 8 depicts how AR frameworks work with mobile cameras in determining and detecting geometric planes using the parallax concept to determine depth. The parallax concept measures the angle of inclination between two lines by means of calculating the difference or displacement of an apparent object viewed along two unrelated lines of sight [28]. Figure 8 presents a single mobile device camera, a user must move the camera around to acquire a "left eye" and "right eye" view (i.e. different lines of sight) to record surfaces. Through the utilization of the mobile camera the AR frameworks are able to record and acquire the different inclination angles between

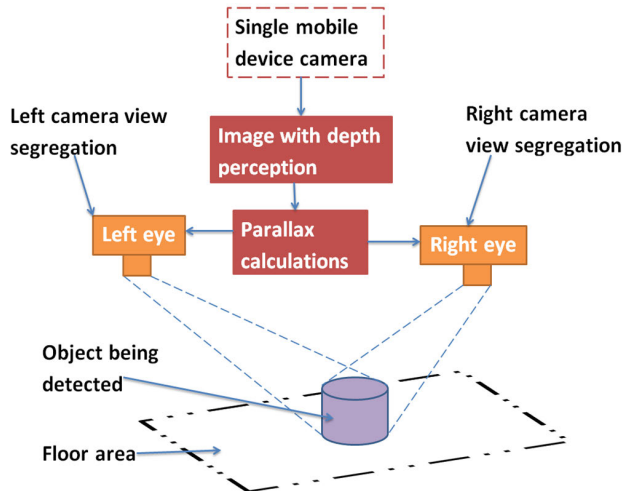


FIGURE 8. Schematic diagram of AR depth plane detection.

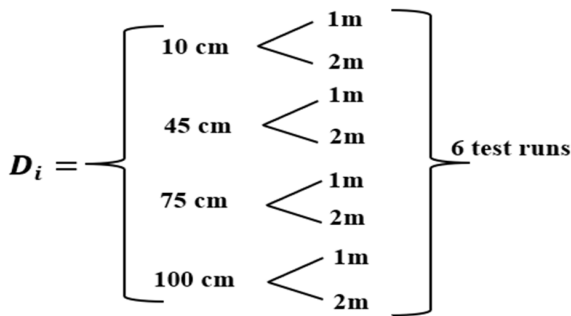


FIGURE 9. Depiction of the distance criteria used with 6 test runs for each device.

the two eye views through the parallax concept, thus enabling plane detection.

B. DATA COLLECTION AND EVALUATION METRIC

1) AR MEASUREMENTS

The nature of data required in the experiment are AR measurements. AR measurements acquired from these tests were compared to measurements captured with a control tape measure. For consistency, the test environment was controlled whereby multiple Ip20 40-watt Dali dimmer strips were used throughout the tests, ensuring adequate and consistent lighting for AR measurements.

A crime scene-related sample scenario was used, where the distance between two items within a crime scene represents either point A or B. The distance between these items were originally set to 100 cm for Figure 10 and 45cm for Figure 11 with a tape measure acting as the control. The same approach was used to set the “75 cm” and “10 cm” distance criteria. A crime scene use case was considered to see how practical AR measurements can assist at measuring crime scene evidence without the risk of contamination by means of physically touching the items. Figure 9 and 10 depict the practical use case of how the 100cm and 45cm criteria worked



FIGURE 10. Crime scene 100 cm measurement taken from 1 meter away.

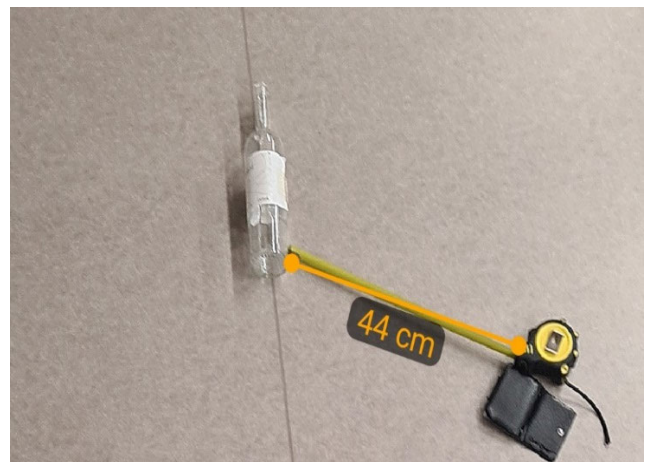


FIGURE 11. Crime scene 45 cm measurement taken from 1 meter away.

when compared to the respective AR measurements observed (i.e., 99cm and 44 cm).

The accuracy of both AR frameworks were assessed based on how close they could get to the control criteria. Six test runs were conducted per device per measurement criteria as seen in Figure 9. Four measurement criteria were used, which are “10cm”, “45cm”, “75cm” and “100cm” at a distance of one meter (1m) and two meters (2m) proximities across all tests. The average score (D) of each device per measurement criteria after six test runs was then computed using Equation (1).

$$\bar{D} = \frac{1}{N} \sum_{j=1}^N X_j. \tag{1}$$

where N represents the total number of test runs that was conducted for each device (D) per distance criterion. Parameter X represents the measured distance between two points (A and B) per time per test for each D .

2) CPU AND RAM UTILIZATION

The next phase of this work required device processing and memory utilization readings whilst using the test application.

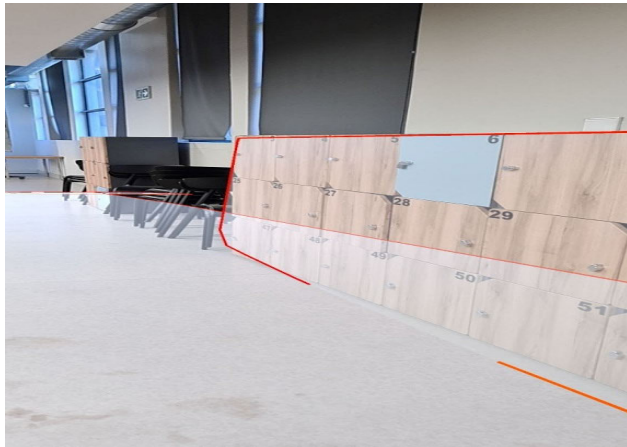


FIGURE 12. Horizontal and vertical test field plane detection.

Data acquired from these tests are then compared and calculated to determine the relative percentage change. End data calculations acquired from Equation (2), which computes the relative change, is used to help determine which AR framework is the most efficient and well optimized. Equation (2) was applied to both CPU and RAM utilization tests. These tests do not take the different operating systems into account, RAM management, CPU architectural builds nor the nanometer fabrication processes across these ten devices.

$$X = \frac{J_2 - J_1}{J_1} \tag{2}$$

(X) represents the relative change, j_2 represents the final value, while j_1 represents the initial value.

3) PLANE DETECTION VARIATION

The final phase of this research evaluated the different AR frameworks in terms of detecting horizontal and vertical planes in different lighting conditions as well as relative runtime in acquiring the relevant planes. Two lighting modes were used from the multiple Ip20 40-watt Dali dimmer strips. The first mode was where all the lights were set at the maximum of 40-Watts of lighting and the second was where the ambient lighting was kept at 14-Watts. For this test, the different AR frameworks had to map out the vertical and horizontal planes of the designated test area as shown in Figure 12.

IV. RESULT AND DISCUSSION

A. AR MEASUREMENTS

Tables 5 and 6 show the raw data captured from the 75cm test criteria taken from 1 meter and 2 meters away respectively. The first row under the table headings represents the control tape measure. N/A has been designated for the control value since its value never changes. Figures 12 and 13 represent visual data extrapolated from the raw test data.

The comparison between 10cm taken at 1 meter (Figure 13) and at 2 meters (Figure 14) depicts a trend whereby 60% of devices running either ARCore or ARKit seem to improve

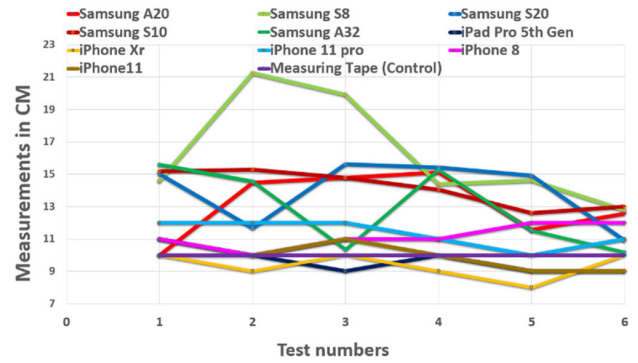


FIGURE 13. 10cm AR measurements taken from 1 meter away.

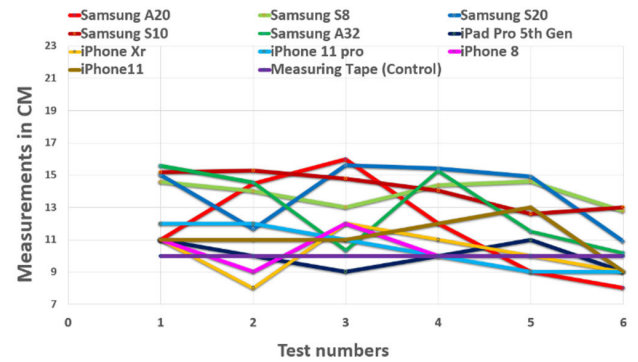


FIGURE 14. 10cm AR measurements taken from 2 meters away.

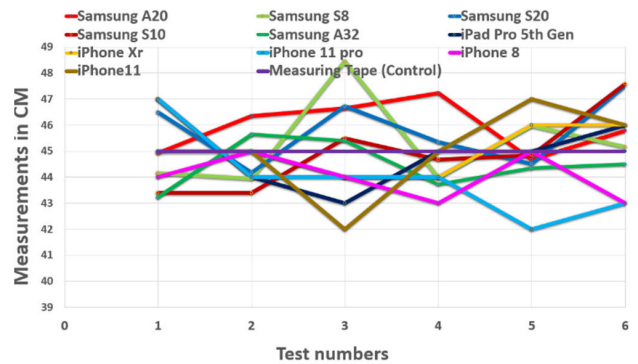


FIGURE 15. 45cm AR measurements taken from 1 meter away.

in terms of accuracy at the 2-meter mark compared to the 1-meter mark. 30% of the devices retained the same accuracy across both measurement criteria, those being the Samsung S10, Samsung S20 and Samsung A32 as seen in Figure 13.

The comparison between 45cm taken at 1 meter (Figure 15) and at 2 meters (Figure 16) shows a different trend compared to the 10cm measurements. In this case, 50% of the devices improved at the 2-meter mark while 40% of the devices decreased in accuracy at the 2-meter mark and only 10% of the devices retained the same accuracy at both 1 meter and 2-meter mark as seen in Figure 15 and 16

The comparison between 75cm taken at 1-meter (Figure 17) and at 2 meters (Figure 18) depicts a trend whereby

TABLE 5. AR 75cm measurements taken from 1 meter away.

Device Name	Result 1 (CM)	Result 2 (CM)	Result 3 (CM)	Result 4 (CM)	Result 5 (CM)	Result 6 (CM)	Average (CM)	Best Result (CM)	Worst Result (CM)
Tape measure	N/A	N/A	N/A	N/A	N/A	N/A	75	N/A	N/A
Samsung S8	73	70	76	75	74	74	73.67	75	76
Samsung S10	73	74	73	76	74	73	73.83	74	76
Samsung S20	75	77	73	77	76	77	75.83	75	77
Samsung A20	74	76	74	73	74	75	74.33	75	76
Samsung A32	106	100	107	105	108	97	103.86	100	108
M1 iPad Pro 5th Gen	72	76	74	74	75	76	74.50	75	76
iPhone Xr	69	73	72	71	74	74	72.17	74	69
iPhone 11 pro	81	73	75	73	72	65	73.17	75	65
iPhone 8	76	75	76	76	77	75	75.83	75	77
iPhone11	73	73	72	75	73	72	73.0	75	72

TABLE 6. AR 75cm measurements taken from 2 meters away.

Device Name	Result 1 (CM)	Result 2 (CM)	Result 3 (CM)	Result 4 (CM)	Result 5 (CM)	Result 6 (CM)	Average (CM)	Best Result (CM)	Worst Result (CM)
Tape measure	N/A	N/A	N/A	N/A	N/A	N/A	75	N/A	N/A
Samsung S8	77	73	74	76	72	73	74.17	76	72
Samsung S10	74	77	76	74	77	76	75.67	74	77
Samsung S20	73	74	72	77	77	74	74.50	74	77
Samsung A20	71	76	77	73	74	74	74.17	74	77
Samsung A32	77	71	75	74	74	71	73.67	75	71
M1 iPad Pro 5th Gen	76	75	75	74	76	75	75.17	75	76
iPhone Xr	75	79	74	73	70	74	74.17	75	79
iPhone 11 pro	77	73	72	76	75	75	74.67	75	72
iPhone 8	72	77	79	76	77	75	76.0	75	79
iPhone11	83	75	73	80	68	70	74.83	75	68

60% of devices running either ARCore or ARKit seem to improve in terms of accuracy the farther away they move from the targets being observed. 30% of devices in general worsened in accuracy the further away they got, and only 10% of the devices retained the same accuracy at both the 1-meter mark and 2-meter mark as seen in Figures 17 and 18.

However, the least accurate device in Figure 18 was the Samsung S20, which shows vast deviations of up to 81cm in the 75cm criteria. The most accurate device in Figure 17 was the iPad Pro. In Figure 18, the worst performing device was the iPhone 11 which had deviations of up to 83cm and the best performing device in Figure 18 was the iPad Pro.

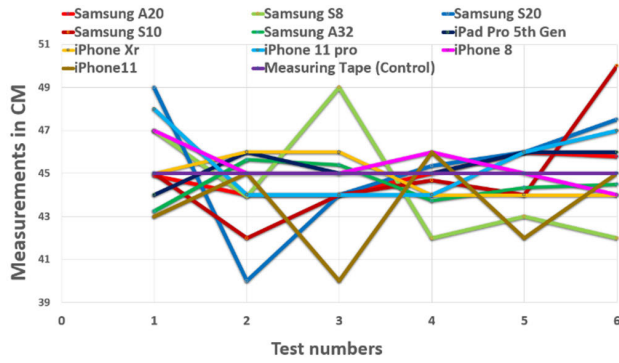


FIGURE 16. 45cm AR measurements taken from 2 meters away.

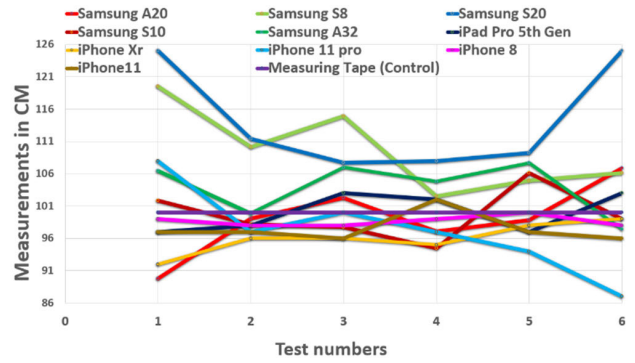


FIGURE 19. 100cm AR measurements taken from 1 meter away.

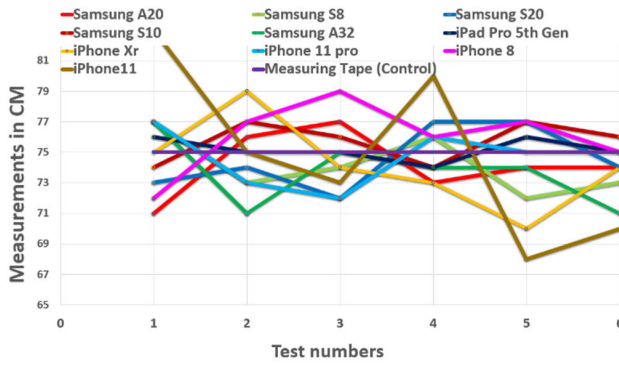


FIGURE 17. 75cm AR measurements taken from 1 meter away.

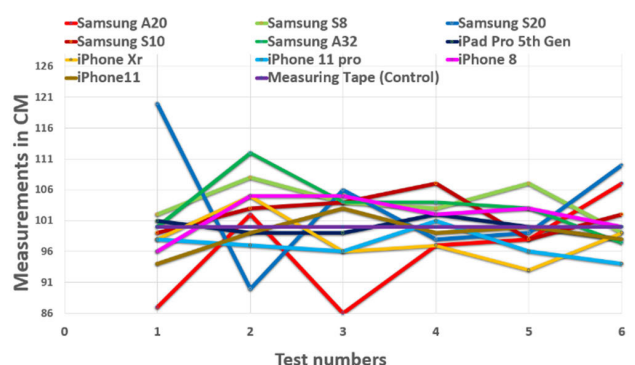


FIGURE 20. 100cm AR measurements taken from 2 meters away.

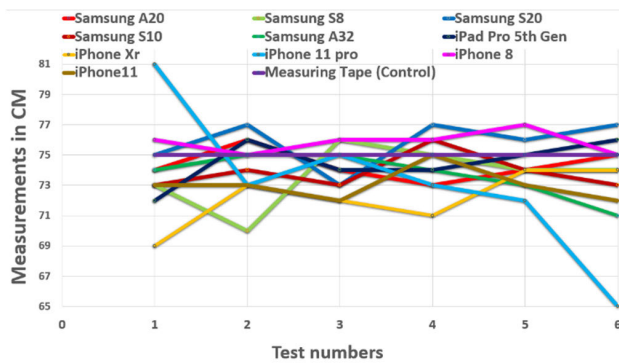


FIGURE 18. 75cm AR measurements taken from 2 meters away.

The comparison between 100cm taken at 1 meter and at 2 meters depicts a great divide between devices that improved in accuracy at the 2-meter mark in comparison to devices that did not improve. 50% of devices running either ARCore or ARKit seemed to improve in terms of accuracy the farther away they were from the targets being measured. 50% of devices running under ARCore or ARKit did not improve and recorded a drop in accuracy at the 2-meter mark, compared to the 1-meter mark as seen in Figures 19 and 20.

The average accuracy percentage scores for the two AR frameworks are measured by applying Equation (1) and comparing them to the control value per criteria. The following can be noted, in Figure 13 ARCore scored an average

TABLE 7. Performance comparison of ARCore and ARKit.

Framework	Average accuracy score	Deviation
ARCore	89.42%	10.58%
ARKit	97.52%	2.48%

accuracy of 59.5% and ARKit scored an average accuracy of 92.67%. In Figure 14 ARCore scored 71.78% and ARKit scored 95.67%. In Figure 15 ARCore scored 99.38% and ARKit scored 99.11%. In Figure 16 ARCore scored 99.63% and ARKit scored 99.26. In Figure 17 ARCore scored 99.02% and ARKit scored 97.87%. In Figure 18 ARCore scored 99.24% and ARKit scored 99.33%. In Figure 19 ARCore scored 94.70% and ARKit scored 97.87%, while in Figure 20, ARCore scored 92.12% and ARKit scored 98.37%.

Table 7 provides a summary and average score of how the two frameworks performed across all devices, spanning ARCore and ARKit in all given tests. These results were obtained by adding all the average percentages values obtained for each device across all tests, as shown in Figures 13, 13, 14, 15, 16, 17, 18 and 20. Then those total results were each divided by 5 to get the average accuracy score for each framework. The final result is shown in Table 7. Table 7 illustrates that ARKit proves to be far superior compared to ARCore regarding accuracy and reliability in AR related applications within the context of the criteria used in this

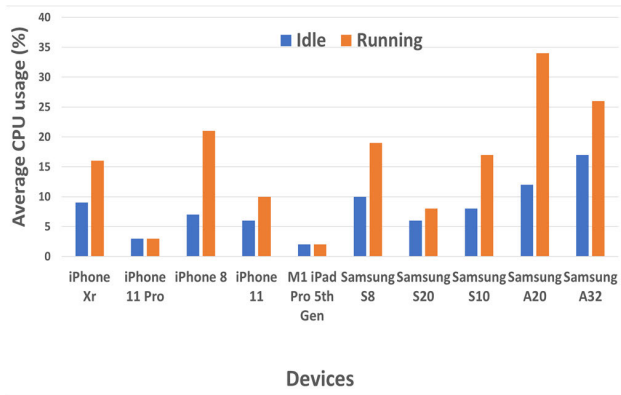


FIGURE 21. Average CPU usage across all devices.

TABLE 8. Relative CPU change.

Framework	Average CPU usage
ARCore	44.92%
ARKit	30.08%

TABLE 9. RAM utilization comparison between ARCore and ARKit.

Framework	RAM CPU usage
ARCore	51.8MB
ARKit	54.2MB

study. The performance ratio of ARKit to ARCore is 5:3 in the eight tests conducted.

B. CPU AND RAM UTILIZATION

Figure 21 shows the average CPU usage per device, which draws the comparison between which AR framework is the most optimized and efficient for diverse computing AR applications. Each CPU check was conducted after a clean reboot with no background applications running, besides the operating system and the important service applications. The mobile devices were first checked 10 minutes after rebooting for idle CPU loads. Later, the devices were then evaluated 10 minutes into running the application for how much CPU load the developed applications were using. Xcode was used to verify this data for all the ARKit devices and CPU profiler was used to verify this data for Samsung devices running ARCore.

Applying Equation (2) to these results helped to determine the relative change amongst the devices. Figure 21 also depicts an expected performance trend between the devices with the weakest computational CPU consumption, and the strongest devices. The lower the CPU utilization the more capable the device is. The device with the best CPU utilization is M1 iPad Pro 5th Gen. The worst performing device in this lineup is the Samsung A20.

Table 8 illustrates a performance summary of how the two frameworks performed across all devices, spanning ARCore and ARKit. ARKit showed a 14.84% lower CPU utilization

TABLE 10. Plane detection summary.

Framework	40-Watts ambient lighting	14-Watts ambient lighting	Relative percentage change
ARCore	7.40s	11.35s	51.08%
ARKit	6.10s	11.45s	90.54%

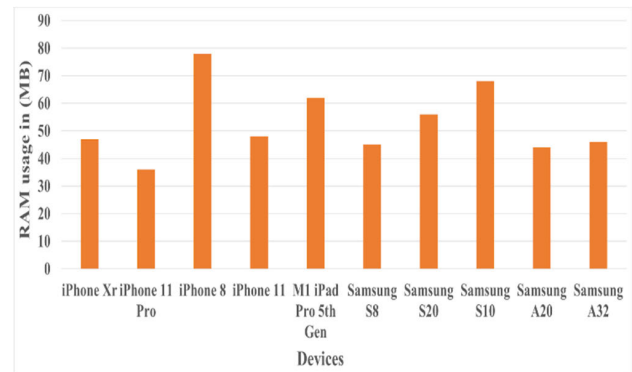


FIGURE 22. RAM usage spanning ten devices.

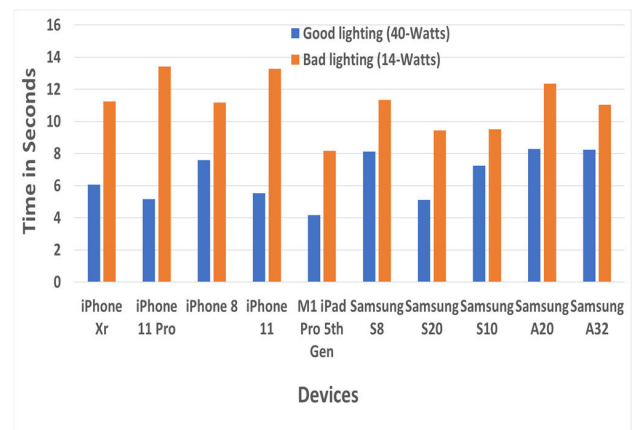


FIGURE 23. Run time of plane detection results under different lighting conditions.

average compared to ARCore. ARKit in this test proves to be far superior compared to ARCore in terms of overall efficiency and optimization regarding CPU management in AR related applications within the context of the criteria used in this study. A point to note is that the CPUs utilized in all ARKit devices are greatly more powerful compared to most CPUs used in some of the ARCore devices, hence the large disparity in the overall CPU utilization.

Figure 22 shows the RAM utilization comparison spanning the ten test devices used in this research. In this test, we noted that the iPhone 8 running ARKit seemed to be the worst optimized device, while the iPhone 11 Pro running ARKit seemed to be the most optimized for the test. On the other hand, the worst optimized device for ARCore was the Samsung S10 and the most optimized was the Samsung A20.

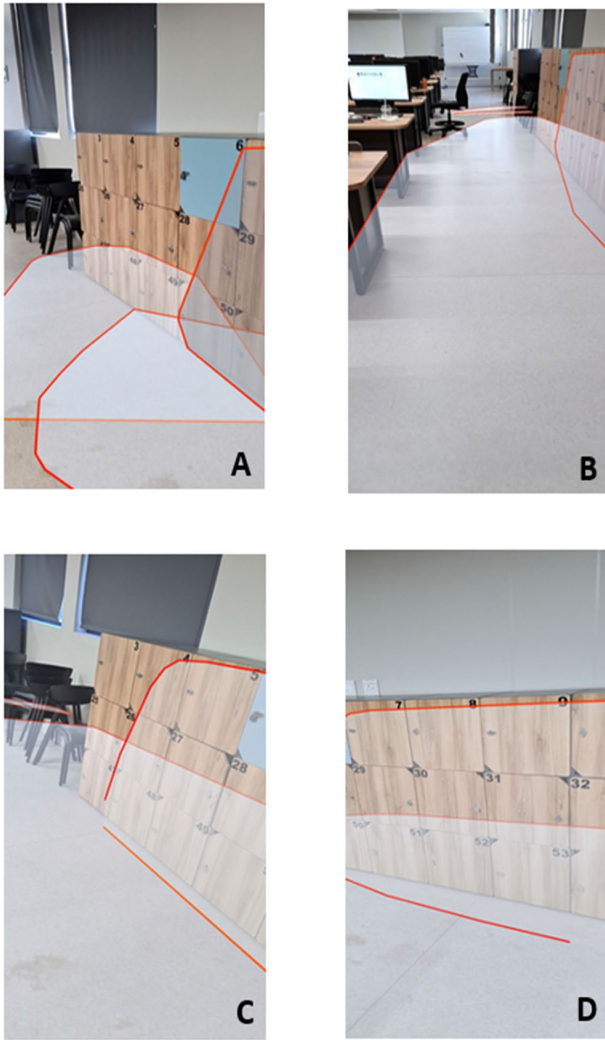


FIGURE 24. Sample plane detection images under 14-Watts.

Table 9 presents the average comparison utilization between ARCore running devices and ARKit running devices. Based on Table 9, one can see that ARCore running devices on average used less RAM than ARKit running devices.

C. PLANE DETECTION VARIATION

Figure 23 shows the comparison between the time taken for planes to be detected under good lighting, which is 40-Watts of ambient lighting, and under bad lighting which is 14-Watts of ambient lighting. This test checks to see how the different AR frameworks function under different lighting conditions since both AR frameworks cater for light estimation and plane tracking.

Table 10 depicts the summary of the average run time taken for planes to be detected within the test area, spanning ARKit and ARCore devices with two different lighting conditions. ARKit on average had a marginal lead over ARCore under 40-Watts of ambient lighting.

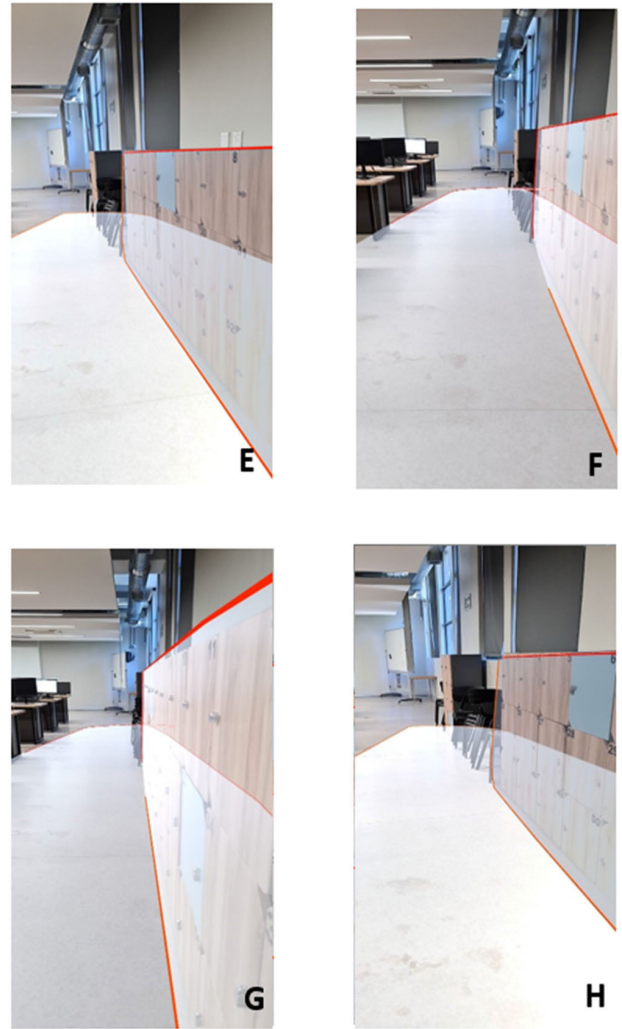


FIGURE 25. Sample plane detection images under 40-Watts.

In the tests conducted, we noted that ARCore running devices on average took 7.40 seconds to completely map out the test area. However, under the same lighting conditions, ARKit on average took 6.10 seconds to map out the same test area, which is significantly better than ARCore. Under 14-Watts of lighting, however, ARCore performs better than ARKit. We noted that the same test area took ARKit running devices on average 11.45 seconds to detect a plane, while it took ARCore running devices on average 11.35 seconds. The following relative percentage change can be noted from the two test cases conducted after applying Equation (2):

- ARCore manages to handle lighting estimation and ranging better than ARKit.
- ARKit had a 90.54% relative percentage change under low light in comparison to good lighting.
- ARCore had a 51.08% relative percentage change under low light in comparison to good lighting.

Figures 24 and 25 present the results obtained from both ARKit and ARCore running devices under different lighting

conditions while detecting planes in the designated test area. In Figure 24, A and B represent two ARCore devices under 14-Watts of ambient lighting, C and D represent two ARKit running devices. In the plane detection section, under bad lighting conditions, in some instances ARCore struggled to map out the test area correctly and at one stage had overlapping planes shown in A. In Figure 25, E and F represent two ARCore devices under 40-Watts of ambient lighting, G and H represent two ARKit running devices. In this test case scenario, both AR frameworks excellently mapped out the entire test area. Minor alignment issues were noted for both ARKit and ARCore running devices as seen in Figures 25(E and G).

An important point to note from all the test parameters conducted in this study is that, the chosen AR frameworks, ARKit and ARCore are not interchangeable between the different operating systems, Android and iOS. ARKit is only accessible to iOS running devices, irrespective of the different hardware components, a device running Android cannot run ARKit, irrespective of whether or not hardware limitations exist. The same can be said about ARCore, which is only offered for Android-running devices. The vendors, Google and Apple, do not cater for cross-platform functionality. Apple distinctively makes ARKit for only Apple devices [29] and Google makes ARCore for only Android running devices.

Tests conducted in this paper utilized two AR frameworks, ARKit and ARCore spanning different devices with different hardware configurations. This testing method ensured that there was no bias, hence the randomized device selection relevant to each framework. Figures 12-19 show that even a high-end device can perform poorly in comparison to a relatively low-end device under certain constraints or parameters. The results show that the accuracy and reliability determination all boil down to how the AR framework functions in each test, and depending on the device. It is expected that this type of exploration will continue to evolve as there are continuous efforts by the vendors on improving these frameworks.

V. CONCLUSION

This work has provided a multifactor comparative evaluation of AR frameworks, ARKit and ARCore, aimed at diverse computing settings, using a crime scene related scenario as a use case. Three major evaluation parameters were considered, which are (i) AR measurements under four distance criteria; (ii) resource (CPU and RAM) utilization capacity; and (iii) plane detection under varying lighting conditions. The first evaluation involves eight test measurement criteria with six test runs per criteria to gauge the accuracy and reliability of the AR frameworks. The second evaluation involves comparing AR framework optimization and efficiency in diverse computing settings in terms of overall CPU and RAM utilization. The third evaluation comprises plane detection and mapping of a given test area under different ambient lighting conditions as well as the relative runtime.

Evaluation one consisted of ten devices split evenly between ARKit and ARCore, the four measurement criteria used are 10cm, 45cm, 75cm and 100cm at a distance of 1 meter and 2 meters' proximity across all tests. For the distance criteria, ARKit proved to be the most accurate and superior between the two frameworks by scoring an average accuracy of 97,52%, as opposed to the 89,42% scored by ARCore. M1 iPad Pro 5th Gen running ARKit was the most accurate and reliable device out of all the devices used based on seven out of the eight criteria used. A point to note is that the M1 iPad Pro 5th Gen houses a LiDAR scanner and has the best configurations for dynamic ranging, which all the other devices do not possess.

Evaluation two consisted of gauging the overall system (CPU and RAM) utilization between the two AR frameworks for efficiency and optimization. Results show that ARKit on average was the most efficient regarding CPU utilization. However, in the RAM management test, ARCore was the most optimized for this work.

Evaluation three consisted of gauging plane detection accuracy and light estimation under different lighting conditions. Results illustrated that ARKit was the preferable choice under 40-Watts of ambient lighting, however, under 14-Watts of ambient lighting ARCore was the preferable choice. Regarding plane detection accuracy across both tests, ARKit was the most accurate in mapping out planes for the designated test area. However, when it came to light estimation under the two different lighting conditions, ARCore was the most preferable AR framework in low lighting conditions, in comparison to ARKit. The results obtained have proven that both frameworks have their strengths within certain parameters, and further development is possible. The findings in this study could also guide future prototyping of immersive applications. It is expected that this type of exploration and analysis will continue to evolve based on future developments and technological advancement.

REFERENCES

- [1] P. A. Rauschnabel, R. Felix, and C. Hinsch, "Augmented reality marketing: How mobile AR-apps can improve brands through inspiration," *J. Retailing Consum. Services*, vol. 49, pp. 43–53, Jul. 2019, doi: [10.1016/j.jretconser.2019.03.004](https://doi.org/10.1016/j.jretconser.2019.03.004).
- [2] A. Boomgaard, K. A. Fritz, O. E. Isafiade, R. C. M. Kotze, O. Ekpo, M. Smith, T. Gessler, K. J. Filton, C. C. Cupido, B. Aden, N. Yokwe, L. Mayekiso, S. Gxowa, A. Levitt, L. Dlodlo, N. Madushana, and D. L. de Laroche Souvestre, "A novel immersive anatomy education system (Anat_Hub): Redefining blended learning for the musculoskeletal system," *Appl. Sci.*, vol. 12, no. 11, p. 5694, Jun. 2022, doi: [10.3390/app12115694](https://doi.org/10.3390/app12115694).
- [3] O. Mabiletsa, I. Omowunmi, S. Viljoen, J. Farrell, and L. Ngqwemla, "Immersive interactive technology: A case study of a wine farm," in *Proc. ITU Kaleidoscope, Ind.-Driven Digit. Transformation (ITU K)*, Dec. 2020, pp. 1–7, doi: [10.23919/ITUK50268.2020.9303195](https://doi.org/10.23919/ITUK50268.2020.9303195).
- [4] X. Li, W. Yi, H.-L. Chi, X. Wang, and A. P. C. Chan, "A critical review of virtual and augmented reality (VR/AR) applications in construction safety," *Autom. Construction*, vol. 86, pp. 150–162, Feb. 2018, doi: [10.1016/j.autcon.2017.11.003](https://doi.org/10.1016/j.autcon.2017.11.003).
- [5] M. A. Maneli and O. E. Isafiade, "3D forensic crime scene reconstruction involving immersive technology: A systematic literature review," *IEEE Access*, vol. 10, pp. 88821–88857, 2022, doi: [10.1109/ACCESS.2022.3199437](https://doi.org/10.1109/ACCESS.2022.3199437).

- [6] P. Wang, P. Wu, J. Wang, H.-L. Chi, and X. Wang, "A critical review of the use of virtual reality in construction engineering education and training," *Int. J. Environ. Res. Public Health*, vol. 15, no. 6, p. 1204, Jun. 2018, doi: [10.3390/ijerph15061204](https://doi.org/10.3390/ijerph15061204).
- [7] S. Formosa, E. Sciberras, and G. Charles, "Taking LiDAR to court: Mapping vapour evidence through spatial forensics," in *Proc. Appl. Geomatics Approaches Maltese Islands SpatialTrain II*, pp. 67–87, 2022.
- [8] H. Kharoub, M. Lataifeh, and N. Ahmed, "3D user interface design and usability for immersive VR," *Appl. Sci.*, vol. 9, no. 22, p. 4861, Nov. 2019, doi: [10.3390/app9224861](https://doi.org/10.3390/app9224861).
- [9] *An Introduction to Immersive Technologies—Vista Equity Partners*. Accessed: Oct. 15, 2022. [Online]. Available: <https://www.vistaequitypartners.com/insights/an-introduction-to-immersive-technologies/>
- [10] R. Aggarwal and A. Singhal, "Augmented reality and its effect on our life," in *Proc. 9th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, 2019, pp. 510–515, doi: [10.1109/CONFLUENCE.2019.8776989](https://doi.org/10.1109/CONFLUENCE.2019.8776989).
- [11] J. Cheng, K. Chen, and W. Chen, "Comparison of marker-based AR and markerless AR: A case study on indoor decoration system," in *Proc. Joint Conf. Comput. Construct.*, 2017, pp. 483–490, doi: [10.24928/JC3-2017/0231](https://doi.org/10.24928/JC3-2017/0231).
- [12] T. Scargill, J. Chen, and M. Gorlatova, "Here to stay: Measuring hologram stability in markerless smartphone augmented reality," 2021, *arXiv:2109.14757*.
- [13] M. T. Abhishek, P. S. Aswin, N. C. Akhil, A. Souban, S. K. Muhammedali, and A. Vial, "Virtual lab using markerless augmented reality," in *Proc. IEEE Int. Conf. Teaching, Assessment, Learn. Eng. (TALE)*, Wollongong, NSW, Australia, Dec. 2018, pp. 1150–1153, doi: [10.1109/TALE.2018.8615245](https://doi.org/10.1109/TALE.2018.8615245).
- [14] P. Gurevich, J. Lanir, and B. Cohen, "Design and implementation of teledvisor: A projection-based augmented reality system for remote collaboration," *Comput. Supported Cooperat. Work*, vol. 24, no. 6, pp. 527–562, Dec. 2015, doi: [10.1007/s10606-015-9232-7](https://doi.org/10.1007/s10606-015-9232-7).
- [15] *Augmented Reality và ứng Dụng Trong Kiến trúc—Point Group*. Accessed: Oct. 15, 2022. [Online]. Available: <https://pointgroup.vn/augmented-reality-va-ung-dung-trong-kien-truc/>
- [16] A. Basu, "A brief chronology of virtual reality," 2019, *arXiv:1911.09605*.
- [17] R. Schroeder, "Virtual reality in the real world: History, applications and projections," *Futures*, vol. 25, no. 9, pp. 963–973, 1993, doi: [10.1016/0016-3287\(93\)90062-X](https://doi.org/10.1016/0016-3287(93)90062-X).
- [18] J. Higginbotham and S. Jacobs, "The future of the Android operating system for augmentative and alternative communication," *Perspect. Augmentative Alternative Commun.*, vol. 20, no. 2, pp. 52–56, Jun. 2011, doi: [10.1044/aac20.2.52](https://doi.org/10.1044/aac20.2.52).
- [19] J. Borduas, A. Castonguay, P. Laurin, and D. Beland, "Reliability of mobile 3D scanning technologies for the customization of respiratory face masks," in *Proc. 11th Int. Conf. Exhib. 3D Body Scanning Process. Technol.*, Oct. 2020, pp. 1–8, doi: [10.15221/20.34](https://doi.org/10.15221/20.34).
- [20] R. Cervenak and P. Masek, "ARKit as indoor positioning system," in *Proc. 11th Int. Congr. Ultra-Modern Telecommun. Control Syst. Workshops (ICUMT)*, 2019, pp. 1–5, doi: [10.1109/ICUMT48472.2019.8970761](https://doi.org/10.1109/ICUMT48472.2019.8970761).
- [21] *Mobile Operating System Market Share Worldwide | Statcounter Global Stats*. Accessed: Nov. 9, 2022. [Online]. Available: <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- [22] P. Nowacki and M. Woda, "Capabilities of ARCore and ARKit platforms for AR/VR applications," in *Proc. Int. Conf. Dependability Complex Syst.*, 2020, pp. 358–370, doi: [10.1007/978-3-030-19501-4_36](https://doi.org/10.1007/978-3-030-19501-4_36).
- [23] T. Chaudhry, A. Juneja, and S. Rastogi, "AR foundation for augmented reality in unity," *Int. J. Adv. Eng. Manage.*, vol. 3, no. 1, pp. 1–7, 2021.
- [24] Z. Oufqir, A. El Abderrahmani, and K. Satori, "ARKit and ARCore in serve to augmented reality," in *Proc. Int. Conf. Intell. Syst. Comput. Vis. (ISCV)*, Jun. 2020, pp. 1–7.
- [25] F. Herpich, R. L. M. Guarese, and L. M. R. Tarouco, "A comparative analysis of augmented reality frameworks aimed at the development of educational applications," *Creative Educ.*, vol. 8, no. 9, pp. 1433–1451, 2017, doi: [10.4236/ce.2017.89101](https://doi.org/10.4236/ce.2017.89101).
- [26] A. Carranza, J. Estrella, R. Zaidi, and H. Carranza, "Plane detection based object recognition for augmented reality," in *Proc. 8th Int. Conf. Control Syst., Robot. (CDSR)*, 2021, pp. 1–8, doi: [10.11159/cdsr21.305](https://doi.org/10.11159/cdsr21.305).
- [27] M. A. Maneli and O. E. Isafiade, "A comparative analysis of augmented reality frameworks aimed at diverse computing applications," in *Proc. ITU Kaleidoscope Extended Reality Boost Quality Exper. Interoperability*, Accra, Ghana, 2022, pp. 1–8, doi: [10.23919/ITUK56368.2022.10003046](https://doi.org/10.23919/ITUK56368.2022.10003046).
- [28] B. Triggs, "Plane + parallax, tensors and factorization," in *Proc. Eur. Conf. Comput. Vis.*, 2000, pp. 522–538.
- [29] L. M. Reimer, S. Weigel, F. Ehrenstorfer, M. Adikari, W. Birkle, and S. M. Jonas, "Mobile motion tracking for disease prevention and rehabilitation using apple ARKit," in *Proc. dHealth*, May 2021, pp. 78–86.



MFUNDO A. MANELI was born in South Africa. He received the National Diploma and Advanced Diploma degrees in software development from Nelson Mandela University, South Africa, and the P.G.Dip. degree in e-skills development with immersive technology from the University of the Western Cape, South Africa, where he is currently pursuing the M.Sc. degree in computer science. He is also a Research Assistant with the University of the Western Cape and also works as a Teaching

Assistant for the PGDip programme. He has published in an accredited high impact journal platform. His research interests include software development, LiDAR 3D reconstruction solutions, computer vision, and immersive technology in diverse computing settings. He was a recipient of the Golden Key International Honor Society Award and the ITU Kaleidoscope Young Author Recognition Certificate.



OMOWUNMI E. ISAFIADI received the M.Sc. and Ph.D. degrees in computer science from the University of Cape Town, South Africa, and the P.G.Dip. degree in mathematical sciences from the African Institute for Mathematical Sciences (AIMS), South Africa.

Prior to joining the University of the Western Cape as a full-time Academician, she was a Post-doctoral Researcher under the National Research Foundation and the Department of Science and Technology (NRF-DST) Innovation Fellowship. Her research interests include data-driven decision support, immersive technology, machine learning applications, and public safety in (emerging) smart cities. She has received various honors and awards, such as the Prestigious L'Oréal-UNESCO for Women in Science, an AIMS-DAAD Fellowship, the Paul G. Allen Family Price, the Golden Key International Honor Society Award, the Google Anita Borg (Women Techmakers) Scholarship (Zürich, Switzerland), the Grace Hopper Scholarship (USA), and the Heidelberg Laureate Forum and German Scholars Organization (HLF-GSO) Grant.

...