

Performance Evaluation of ARCore Anchors According to Camera Tracking

Shinhyup Lee¹, Leehwan Hwang², Seunghyun Lee², Taewook Kim³ and Soonchul Kwon^{3*}

¹Department of Plasma Bio Display, Graduate School of Kwangwoon University, South Korea

²Department of Immersive Content Convergence, Graduate School of Kwangwoon University,
South Korea

³Department of Interdisciplinary Information System, Graduate School of Smart Convergence,
Kwangwoon University, South Korea

E-mail: jx007s@nate.com, optics.hwang@kw.ac.kr, shlee@kw.ac.kr, ksc0226@kw.ac.kr

Abstract

Augmented reality (AR), which integrates virtual media into reality, is increasingly utilized across various industrial sectors, thanks to advancements in 3D graphics and mobile device technologies. The IT industry is thus carrying out active R&D activities about AR platforms. Google plays a significant role in the AR landscape, with a focus on ARCore services. An essential aspect of ARCore is the use of anchors, which serve as reference points that help maintain the position and orientation of virtual objects within the physical environment. However, if the accuracy of anchor positioning is suboptimal when running AR content, it can significantly diminish the user's immersive experience. We are to assess the performance of these anchors in this study. To conduct the performance evaluation, virtual 3D objects, matching the shape and size of real-world objects, we strategically positioned ourselves to overlap with their physical counterparts. Images of both real and virtual objects were captured from five distinct camera trajectories, and ARCore's performance was analyzed by examining the difference between these captured images.

Keywords: ARCore, ARCore API, Anchor, Augmented Reality, Difference Images

1. Introduction

Augmented Reality (AR) involves the superimposition of virtual objects onto the real environment. Ronald Azuma's definition emphasizes the real-time merging of reality and virtual images, their coincident location in three-dimensional space, and their capacity for interaction [1-3]. Recent developments in AR technology have been driven by the enhanced capabilities of sensors and cameras in mobile devices, as well as advancements in image processing [4-6]. The growing demand for mobile device-based AR has led to the emergence of platforms like Google's ARCore, Apple's ARKit, and Facebook's Camera Effects Platform [7]. Google embarked on an AR platform project, combining Tango and ARCore, from 2015 until January 2018. Subsequently, research and services in AR technology have predominantly centered on ARCore. ARCore's anchor functionality serves as a reference point, ensuring that virtual objects maintain their precise position and orientation within the real world. The anchor tracking in ARCore encompasses the Instant Placement API, Augmented Images API, Augmented Faces API, and Geospatial API. These APIs leverage the mobile device's camera, sensors, and GPS location data to identify and track anchors necessary for AR services

Manuscript received: September. 26, 2023 / Revised: October. 13, 2023 / Accepted: October. 17, 2023

Corresponding Author: kcs0226@kw.ac.kr

Tel: +82-2-940-8637, Fax: +82-50-4174-3258

Associate professor Graduate School of Smart Convergence, Kwangwoon University, South Korea

[8-10]. However, when a mobile device is in motion, the anchor information derived from sensors and cameras changes, leading to reduced accuracy. Inaccurate anchor positioning can cause AR objects to deviate from their intended real-world locations, diminishing the user's sense of immersion while using AR services. Therefore, there is a need for performance evaluation research on the ARCore API.

2. Experiment

2.1 Experiment Environment

In this experiment, an Android application was developed using JAVA SDK 1.8, and it was integrated with ARCore 1.31.0 API. The visualization of the 3D virtual object was achieved using OpenGL GLES 3.0 [11]. The experimental device employed was the Samsung Galaxy Tab S8, model name SM-X700N. Figure 1 presents a physical object that was meticulously crafted to match a 3D virtual object. Figure 1(a) shows 3D virtual object, and Figure 1(b) shows a physical object. This object is a cube with each side measuring 16 cm. On each side, there is a black and white grid pattern arranged in a 4 x 4 configuration, with each grid measuring 4 cm x 4 cm.

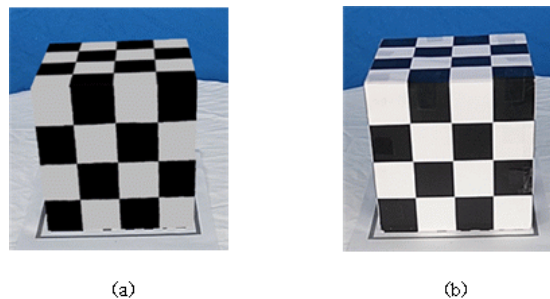


Figure 1. Object created for performance evaluation

(a) Virtual object (b) Real object

To ensure precise measurements, the camera position was defined using a camera tracking system in this experiment. Figure 2 illustrates the setup of the experimental camera tracking system. The cameras were positioned at five different locations: 1) directly in front of the object (0 degrees), 2) at a 35-degree angle to the left, 3) at a 75-degree angle to the left, 4) at a 35-degree angle to the right, and 5) at a 75-degree angle to the right. This setup allowed for precise positioning adjustments using a motorized track.



Figure 2. Camera track system

2.2 Performance Evaluation Method

The difference between two images obtained by performing a subtraction operation is referred to as the difference image (DI) [12]. In this experiment, a difference calculation is conducted on images captured at various locations, where DI is computed based on pixel values. Equation (1) illustrates the process for calculating the difference image at each location. In this equation, R represents the real object image, while V represents the virtual object image. DI corresponds to the absolute pixel value difference resulting from subtracting R from V.

$$DI(x,y) = |V(z,y) - R(k,y)| \quad (1)$$

This experiment is susceptible to minor variations in lighting and environmental conditions during image capture, which can impact the image calculation process. To address this, the acquired images are subjected to binary processing. This procedure enhances accuracy by reducing the MAE [13-15].

As part of the experiment's preprocessing phase, extraneous background elements are removed from both the real object and virtual 3D object images. Subsequently, the background-removed image is converted into a 256-level grayscale image. This converted image is then binarized using a specified threshold. The MAE value is subsequently computed. Figure 3 provides an overview of the Mean Absolute Error's progress.

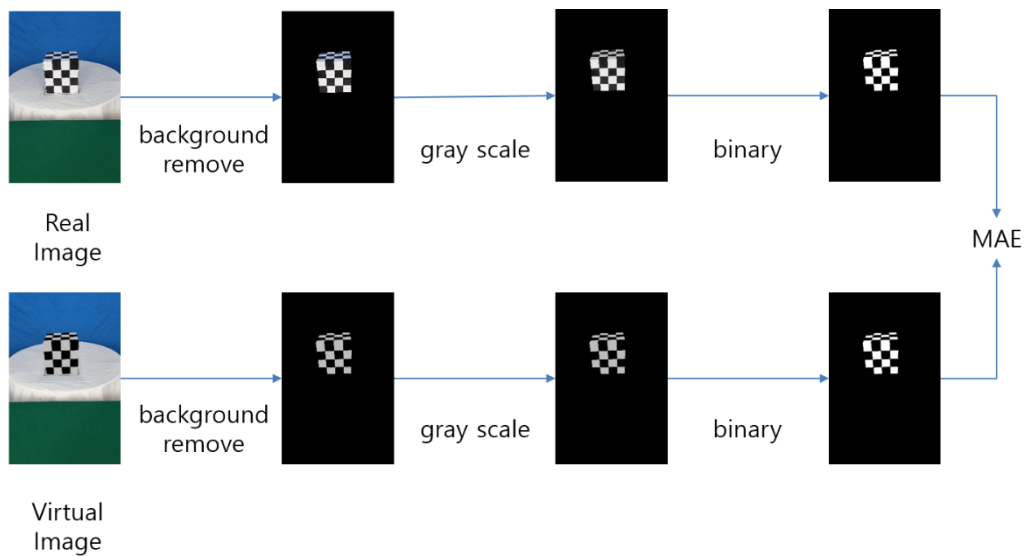


Figure 3. Mean absolute error process

Equation (2) outlines the procedure for background removal from each image. Here, T represents the captured image, and B is the image with the background included. $T(x, y)$ corresponds to the pixel value at coordinates X, Y within the captured image, while $B(x, y)$ denotes the pixel value at the X, Y coordinates in the background-captured image. If the pixel values at the corresponding coordinates in the captured image and background-captured image match, the pixel from the captured image is retained. In cases where they do not match, a value of 0 is assigned. The resulting value is used to calculate the pixel value $BS(x, y)$ at the X, Y coordinates in the new image file.

$$BS(x,y) = \begin{cases} T(x,y) & (T(x,y) \neq B(x,y)) \\ 0 & (T(x,y) = B(x,y)) \end{cases} \quad (2)$$

Equation (3) outlines the process of converting an image with the background removed into a grayscale image. In this equation, (x, y) represents the X, Y coordinates of the image, $R(x, y)$ corresponds to the red channel value of the pixel, $G(x, y)$ corresponds to the green channel value, and $B(x, y)$ represents the blue channel value. The grayscale value of the pixel is computed by taking the arithmetic mean of these three values.

$$Gray(x,y) = (R(x,y) + G(x,y) + B(x,y))/3 \quad (3)$$









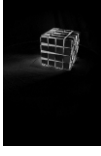
Equation (4) describes the binary processing applied to a grayscale image. The variable 't' in this equation signifies the threshold that distinguishes between white and black. $Gray(x, y)$ denotes the grayscale value of the image pixel at coordinates X, Y. If this value is greater than the threshold, 255 is assigned; otherwise, if it is less than or equal to the threshold, 0 is assigned. The real image and virtual object image are consecutively processed using Equation (2), Equation (3), and Equation (4), culminating in the extraction of the difference image through the computation defined in Equation (1).

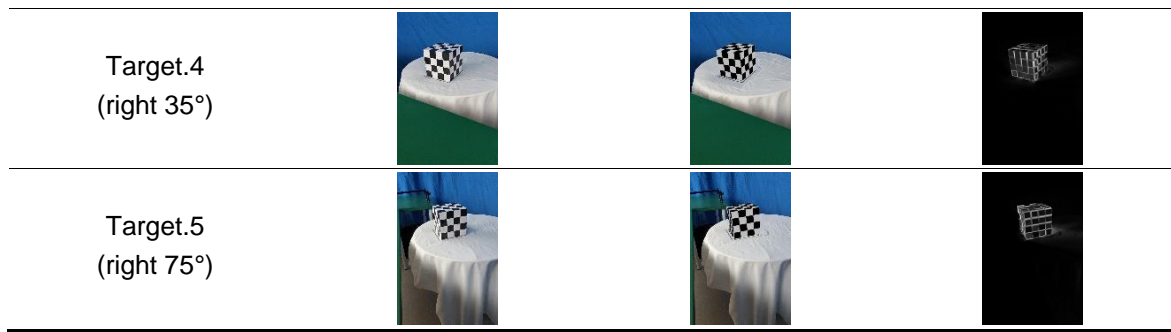
$$Binary(x,y) = \begin{cases} 255 & (Gray(x,y) > t) \\ 0 & (Gray(x,y) \leq t) \end{cases} \quad (4)$$

3. Results

Table 1 displays the difference images between the real object and virtual 3D object images captured at each location. All these images share the same resolution, measuring 1600 x 2458, and contain a total of 3,932,800 pixels. Difference images are computed on a per-pixel basis, where a 100% match is represented as black, and decreasing concordance rates are depicted in shades of white.






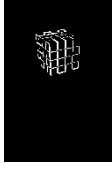








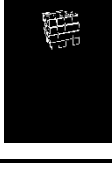
Table 1. Calculation of difference image between real object and virtual 3D object

Position	Real Object	Virtual 3D Object	Difference image
Target.1 (center 0°)			
Target.2 (left 35°)			
Target.3 (left 75°)			



To minimize the impact of lighting and external factors, the creation of difference images was meticulously executed. The ARCore error was quantified using the MAE metric. This experiment involved removing background elements from the captured images, leaving only the object, and then comparing these with the real object and virtual 3D object images. Table 2 presents the MAE calculated based on the binary images from each location. These are the difference images computed using binary representations of the real object and virtual 3D object for each respective location.

Table 2. Mean absolute error image

position	Real Object	Virtual 3D Object	Difference image
Target.1 (center 0°)			
Target.2 (left 35°)			
Target.3 (left 75°)			
Target.4 (right 35°)			
Target.5 (right 75°)			

4. Discussion

In Table 3, we present the percentage of matching pixels across all pixels. The matching criteria were determined by tallying only the black pixels (0,0,0) in the difference images. Concerning image concordance rates, Target 1, which corresponds to the frontal view, exhibited the highest concordance rate at 71.61%. In contrast, Target 3, located at 75 degrees to the left, had the lowest matching rate at 36.59%.

Table 3. Difference image accuracy

position	Total pixel	Similar pixel	Difference pixel	matching rate
Target.1 (center 0°)	3,932,800	2,816,132	1,116,668	71.61%
Target.2 (left 35°)	3,932,800	2,319,015	1,613,785	58.97%
Target.3 (left 75°)	3,932,800	1,438,867	2,493,933	36.59%
Target.4 (right 35°)	3,932,800	2,097,202	1,835,598	53.33%
Target.5 (right 75°)	3,932,800	1,978,348	1,954,452	50.30%

Table 4 showcases the percentage of matching pixels in relation to all pixels, using the pixels from the MAE difference images. The criteria for pixel matching focused on black pixels (0,0,0) within the difference images. Remarkably, the MAE concordance rate demonstrated superior results compared to the difference images derived from the actual images, as indicated in Table 2. The average absolute error agreement rate was found to be 95.24%. In descending order, Target 1 (front) achieved the highest matching rate at 99.39%, followed by Target 4 (35° right) at 98.29%, Target 5 (75° right) at 98.16%, Target 2 (35° left) at 97.83%, and Target 3 (75° left) at 97.48%.

Table 4. Average absolute error Percentage

position	total	similar	difference	matching rate
Target.1 (center 0°)	3,932,800	3,908,825	23,975	99.39%
Target.2 (left 35°)	3,932,800	3,847,410	85,390	97.83%
Target.3 (left 75°)	3,932,800	3,833,830	98,970	97.48%
Target.4 (right 35°)	3,932,800	3,865,419	67,381	98.29%
Target.5 (right 75°)	3,932,800	3,860,321	72,479	98.16%

5. Conclusion

The AR platform, such as Google's ARCore, is a powerful technology that alleviates the challenges of developing AR technology from scratch. It not only provides a clear development direction but also offers efficient methods for implementation. The Google ARCore API offers tremendous convenience for implementing AR technology, thanks to its features like motion tracking, environmental understanding, instant placement, augmented images, and geospatial capabilities. However, with the expanding application areas of AR technology, there is a growing need for platform performance evaluation and enhancement. When utilizing the Google ARCore API, it becomes crucial to assess the technology's accuracy and performance.

In this study, we conducted a performance evaluation of the ARCore API, measuring the matching rate

when implementing virtual objects based on image anchors. Accuracy was assessed through the use of difference images and MAE. The results of our experiment indicated that the highest accuracy was achieved in the front view based on the image anchor, while the accuracy on the sides was found to be comparatively lower. We confirmed that the accuracy of the anchor varies depending on the location.

Additionally, we conducted a performance evaluation of the augmented image API. During the measurement of difference images, we observed varying performance in ARCore due to real-world environmental factors, including external interferences and lighting conditions. This suggests that ARCore's performance can be influenced by these external factors. It's important to note that our study was limited to evaluating the performance of the augmented image API, and we did not delve into anchor research related to instant placement, geospatial features, or augmented faces. Consequently, there are limitations in our evaluation of the overall functionality of ARCore.

The primary objective of this study was to validate the results of the ARCore API performance evaluation. We anticipate that this experimental approach will contribute to ongoing assessments of AR technology performance. In the future, when working with the ARCore API, further research should explore how ARCore's performance is affected by interference factors and should aim to enhance ARCore's overall performance.

6. Acknowledgement

This research was supported by the Ministry of Science and ICT(MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2023-RS-2023-00258639), supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP). And, this research is supported by the Ministry of Culture, Sports and Tourism and Korea Creative Content Agency(Project Number: R2021040083).

7. Reference

- [1] Jason Tham, Ann Hill Duin, Laura Gee, Nathan Ernst, Bilal Abdelqader and Megan McGrath, "Understanding Virtual Reality: Presence, Embodiment, and Professional Practice," *IEEE Transactions on Professional Communication*, Vol. 61, Issue 2, pp. 178-195, 15 March 2018
DOI: <https://doi.org/10.1109/TPC.2018.2804238>
- [2] Se-Woong Kim and Yun-Sung Cho, "A Case Study of Realistic Media Service Using Virtual Reality Technology," *Korea Institute of Design Research Society*, vol. 4. no. 2, pp. 29-38, 2019.
- [3] Hee Kyung Cho, Sung Hoon Kim, "A Study on Affordance Design Characteristics in Augmented Reality(AR) Digital Signage Advertisement," *JOURNAL OF THE KOREAN SOCIETY DESIGN CULTURE*, Vol. 24, pp. 611-623, 30 Sep 2018
DOI: <https://doi.org/10.18208/ksdc.2018.24.3>.
- [4] Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage and Mika Ylianttila, "A Survey on Mobile Augmented Reality With 5G Mobile Edge Computing: Architectures, Applications, and Technical Aspects," *IEEE Communications Surveys & Tutorials*, Vol. 23, Issue: 2, pp. 1160-1192, 25 February 2021
DOI: <https://doi.org/10.1109/COMST.2021.3061981>
- [5] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier and B. MacIntyre, "Recent advances in augmented reality," *IEEE Computer Graphics and Applications IEEE Comput. Grap. Appl. Computer Graphics and Applications*, *IEEE.*, Vol. 21, Issue 6, pp. 34-47, Jan 2001.
DOI: <https://doi.org/10.1109/38.963459>

- [6] Yong-Hwan Lee, Yukyong Lee, Je-ho Park, Kyoungro Yoon, Cheong Ghil Kim and Youngseop Kim, "Trend of Technologies and Standardizations for Mobile Augmented Reality," *The Korea Society of Satellite Technology*, Vol. 8, No. 1, pp.83-88, March 2013.
- [7] Zainab Oufqir, Abdellatif El Abderrahmani and Khalid Satori, "ARKit and ARCore in serve to augmented reality," *2020 International Conference on Intelligent Systems and Computer Vision (ISCV) Intelligent Systems and Computer Vision (ISCV)*, pp. 1-7, 2020.
DOI: <https://doi.org/10.1109/iscv49265.2020.9204243>
- [8] Nayfah Mohsen Almutairi, Abdulrahman Alkandari, Wasmya Alhayan and Abeer Essa Alomairi, "Google Project Tango and ARCore Under the View of Augmented Reality," *Journal of Computational and Theoretical Nanoscience*, Vol. 16, No. 3, pp. 1127-1133, March 2019.
DOI: <https://doi.org/10.1166/jctn.2019.8007>
- [9] Ju-Eun Song and Joong-jin Kook, "Building a Mobile AR System Based on Visual SLAM," *Journal of the Semiconductor & Display Technology*, Vol.20, No. 4 pp. 96-101, 2021.
DOI: <https://doi.org/10.3390/asi5010011>
- [10] Florent Duguet and George Drettakis, "Flexible point-based rendering on mobile devices," *IEEE Comput. Graph. Appl.*, Vol.24, Issue 4, pp. 57-63, July-Aug. 2004
DOI: <https://doi.org/10.1109/mcg.2004.5>
- [11] K. Pulli, T. Aarnio, K. Roimela and J. Vaarala, "Designing graphics programming interfaces for mobile devices," *IEEE Computer Graphics and Applications*, Vol. 25, Issue 6, pp. 66 - 75, 07 November 2005.
DOI: <https://doi.org/10.1109/MCG.2005.129>
- [12] Seok-Woo Jang and Jin-Uk Kim, "Analysis of Moving Object Trajectories Using Difference Images," *Journal of Korean Institute of Information Technology*, Vol. 3, No. 3, pp. 19-28, July 2005.
- [13] Soon H. Kwon, "Advanced Image Quality Index-based Binalization of Gray Images," *Journal of Korean Institute of Intelligent Systems*, Vol. 30, No. 3, pp. 236-241, July 2020.
DOI: <https://doi.org/10.5391/jkiis.2020.30.3.236>
- [14] Taegeun Oh and Sanghoon Lee, "Blind Sharpness Prediction Based on Image-Based Motion Blur Analysis," *IEEE Transactions on Broadcasting*, Vol. 61, Issue 1, pp. 1-15, 11 February 2015.
DOI: <https://doi.org/10.1109/TBC.2015.2395092>
- [15] Donggu Lee, Young-Ghyu Sun, Soo-Hyun Kim, Issac Sim, Kye-San Lee and Myoung-Nam Song, "CNN-based Image Rotation Correction Algorithm to Improve Image Recognition Rate," *The Journal of the Institute of Internet, Broadcasting and Communication(JIIBC)*, Vol. 20, No. 1, pp.225-229, 2020.
DOI: <https://doi.org/10.1109/TBC.2015.2395092>