Evaluation of CPU And RAM Performance for Markerless Augmented Reality

Tagred A. Alkasmy †, Rehab K. Qarout*† and Kaouther Laabidi††,

†Department of Computer Science & Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Saudi Arabia
††Department of Computer Engineering and Networks, College of Computer Science and Engineering , University of Jeddah, Saudi Arabia

Summary
Augmented Reality (AR) is an emerging technology and a vibrant field, it has become common in application development, especially in smartphone applications (mobile phones). The AR technology has grown increasingly during the past decade in many fields. Therefore, it is necessary to determine the optimal approach to building the final product by evaluating the performance of each of them separately at a specific task. In this work we evaluated overall CPU and RAM performance for several types of Markerless Augmented Reality applications by using a multiple-objects in mobile development. The results obtained are show that the objects with fewer number of vertices performs steady and not oscillating. Object was superior to the rest of the others is sphere, which is performs better values when processed, its values closer to the minimum CPU and RAM usage.

Keywords:
Augmented Reality, CPU performance, RAM Performance, Markerless AR, Mobile AR.

1. Introduction

Augmented Reality (AR) is an emerging technology and a vibrant field, it has become common in application development, especially in smartphone applications (mobile phones). The AR technology has grown increasingly during the past decade in many fields such as education, human interaction with computers, health, commerce, industry, games, construction, and others.

Augmented Reality (AR) is the technology based on projecting virtual objects and information into the user's actual environment to provide additional information or act as a vector for him. The user can deal with information and virtual objects in Augmented Reality through several devices maybe they are portable such as a smartphone. It can be also wearable devices such as glasses, contact lenses, and all these devices use a tracking system that provides accurate projection and displays the information in the right place such as the global positioning system (GPS), and camera.

There are different types of AR applications, Marker-based AR apps use markers (target images) to indicate things in a given space. These markers determine where the AR application places digital 3D content within the user's visual field or through a camera feed. Markerless AR places virtual 3D objects in the physical environment depending on the environment's real features rather than identifying markers. This differentiation eliminates the need for object tracking systems.

The number of studies on AR gradually increased, most of which proved the positive results and the effectiveness of Augmented Reality use on the human side of the user and on human performance. The development of Augmented Reality applications works to improve human-computer interaction, human performance, speed, and accuracy in learning, produce solutions and complete operations. On the other hand, very few research and empirical studies explore Augmented Reality applications affect the significant parts of the computer architecture and how they may affect the performance of the CPU, GPU, RAM, and other essential components of the computer architecture. The rapid growth of this technology has brought many challenges, including inefficiency and limited resources in a device. Therefore, it is necessary to determine the optimal approach to building the final product by evaluating the performance of each of them separately at a specific task.

To achieve this goal, this paper was studying the effect of Augmented Reality applications on the CPU and RAM usage, analyzing the results and then comparing the performance for each one.

This work is structured as follows. In the Related work section, present the past researches about the topic. In the Methodology section, an overall description of the code implementation and workflow for the overall process is shown. In the Experiment section, the experiment described in details and performance evaluation is shown. In the Results section, the obtained values for the CPU and RAM evaluation are analyzed. Finally, in the Conclusions section, current results are presented.

Manuscript received October 5, 2023
Manuscript revised October 20, 2023
https://doi.org/10.22937/IJCSNS.2023.23.10.6

2. Related Works:

Studies have proven the impact of Augmented Reality systems on users' mental and cognitive performance. In 2014, a paper examined the effect of Augmented Reality on improving cognitive performance, as researchers developed learning tools that allow the user to control a 3D model of fine chemical particles and test them in action. This test resulted in a good effect on the cognitive performance of the users [7].

In another study in 2021 on the impact of augmented reality systems on mental workload and task performance, the results showed that Augmented Reality reduces the cognitive load and frees up mental resources, allowing users to focus more on performing tasks [6].

As for devices, CPU usage data is the primary data used to monitor performance, and the source of that data is the operating system [3]. However, there is no clear and comprehensive evaluation of the performance of the devices during the implementation of Augmented Reality applications [4].

In 2013 researchers introduced a parallel processing model of the CPU and GPU for mobile Augmented Reality devices using intensive image processing algorithms. The proposed scheme distributes the feature extraction unit and feature description unit to CPU and GPU, respectively, and then executes them parallel. The experimental results showed that the parallel scheme outperformed the sequential CPU scheme only without GPU [2].

In 2013, researchers proposed a smart marker-based tracking system that detects QR codes. Then, they analyzed the effects of such a system. The results showed that image processing speed is directly proportional to the power of the processor. The speed of image processing is inversely proportional to the screen's resolution and the camera's resolution [5].

Finally, in 2021 the researchers performed a quantitative assessment of overall CPU and RAM usage when applying different types of markers. They used an Android device with a specific feature to carry out the experiment. They suggested several scenarios based on markers (fiducial, natural) to evaluate CPU and RAM usage from the runtime perspective. The results showed that markers with fewer heads, such as a ball, perform better than those with more heads. Therefore, developers should consider those results to improve performance [1]. Existing research lacked an assessment of Markerless, location-based, and their effect on CPU, GPU, and RAM performance.

3. Methodology:

We designed and built multiple scenarios to evaluate CPU and RAM performance from an execution time perspective; these scenarios use different Markerless applications.

First, in the implementation stage we have developed four different applications. Each of which contains a three-dimensional geometric shape that automatically appears in the user's actual environment through the camera when the program is run. In this paper, we chose geometric shapes with different specifications, such as (cube, capsule, cylinder, sphere), each with its own characteristics, provided that each shape is implemented separately in a separate application. As shown in figure1, applications were run in these steps starting from running application, open phone camera, tacking object and provide CPU and RAM usage. After the different applications were implemented, we set several fixed distances between the cell phone camera and the flat ground for the practical application, so that the effect could be measured in several different conditions. We also standardized the lighting in the place of application of the experiments to avoid the effect of changes in light intensity on the results to be obtained.

Experiments were conducted with different time periods for recording the results.

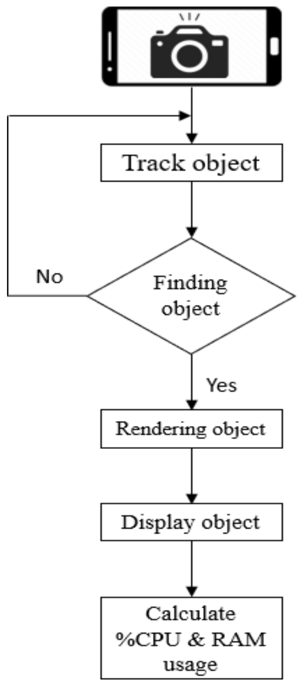

Figure1: Application structure.

Finally, we collected and analyzed the results, and then calculated each CPU and RAM usage percentage. Using Microsoft Excel, we compared the results to find out the other differences in the effect of Markerless Augmented Reality on the performance of CPU and RAM. As shown in figure2, the workflow present steps of running application in details. After application is start, it will start to checking number of executions to continue finding new CPU and RAM usage or stopping application and calculate average of CPU and RAM usage.

4. Experiment:

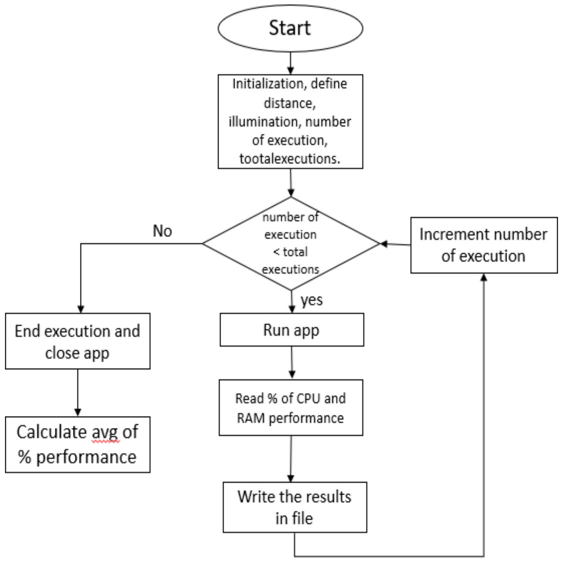

Figure2: Methodology workflow.

To implement the experiments, four different scenarios were proposed (cube, cylinder, sphere, capsule). An application was developed to evaluate each scenario individually. The experiment environment consists of a white LED illumination focused on the target surface. Three different distances between the surface and the phone are defined as (20, 50, 70, 100) centimeters. The results are read after 20, 40, 60 seconds of execution. The number of executions (total executions= 10) is specified for each application. Figure3 shows an example set up for the experiment scenarios. The implementation process is done by running the application and after the predetermined period has elapsed, the percentages are read and then restarted until the maximum number of execution times is reached.

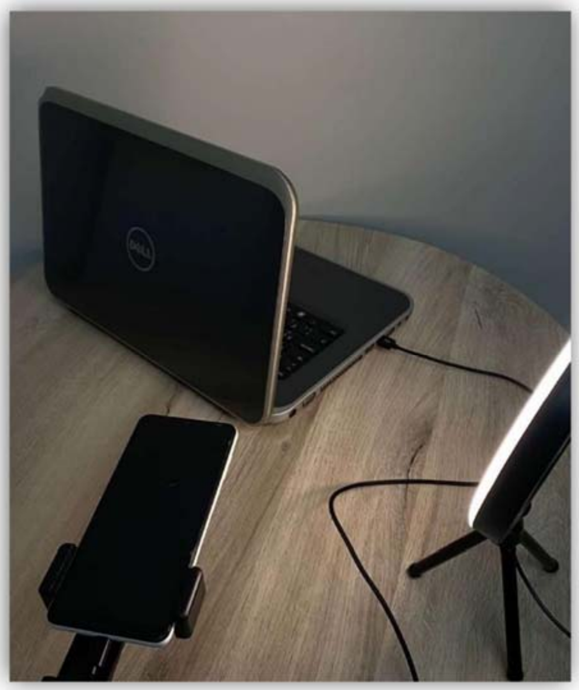

Figure3: Example of an experiment scenario.

The different applications were made according to the scenario to appear 3D object every time, the application opens the Android phone's camera and then recognizes the surface. After the surface is recognized, 3D object is displayed on the surface. These applications were developed using (Unity3D [13] and Vuforia [14]) where 3D object was created in each application with a uniform size for all applications, where the 3D object appears after opening the phone's camera, then the results were read and recorded. After recording the results were compared through tables and graphs. Experiments were performed on an Android phone (Samsung galaxy A12). The device has a CPU with an octa-core processor, with 4GB of RAM and 64GB of ROM. The device has quad rear camera with 5, 2, 2 MP and 48MP for main camera. Figure4 shows a simulation of running different applications.

Figure4: Simulation of running applications.

5. Result:

After carrying out all the experiments and recording all the results, the results were presented in Tables 1 and 2. Table 1 presents the CPU usage percentage results for the various experiments mentioned previously according to the specified variables. As for Table 2, it presents the RAM usage percentage results. In both tables, the rows represent the selected variables (the distance between the phone and the surface in centimeters, and the time for recording results in seconds), while the columns display 3D objects details. Each number recorded in the table represents the average percentage of total use obtained after 10 times. The maximum value of the CPU
usage can be seen which is 25.22%, while the minimum value is 16.7%. Also, for RAM usage, the maximum value is 9.1% and the minimum value is 5.4%.

Table 1: Means for overall CPU. Colour reference: green =low, red = high.

Time	Cube	Sphere	Capsule	Cylinder	Distance
20	21.24	18.73	17.55	19.73	20cm
40	18.62	17.59	16.8	17.9	
60	18.76	17.99	18.21	17.91	
20	18.76	17.3	18.4	18.0	50cm
40	18.32	16.8	19.0	19.3	
60	17.21	17.8	19.0	19.8	
20	25.22	17.8	18.8	18.6	70cm
40	23.26	18.5	18.9	18.8	
60	17.73	18.9	17.8	18.6	
20	16.98	17.9	19.7	18.3	100cm
40	17.26	17.8	19.8	18.2	
60	17.29	16.7	20.2	18.8	

By looking at Table 1, it can be seen that the maximum value of CPU usage appeared when processing cube at 70cm, while the minimum value appeared when processing sphere at the longest distance between the phone and the surface. This means that increasing the distance had a negative effect on 3D objects except for the sphere, the effect was good.

In Table 2, the good effect of increasing the distance between the phone and the surface can be seen on RAM usage. Where the maximum value of RAM usage appeared when processing sphere at 20cm, while it was the minimum value a cylinder at a larger distance.

When comparing the values in Tables 1,2, we found the sphere was superior to the rest of the object, as the value of sphere when processed was closer to the minimum CPU and RAM usage, while the higher values differed from one object to another.

Table 2: Means for overall RAM. Colour reference: green =low, red = high.

Time	Cube	Sphere	Capsule	Cylinder	Distance
20	6.0	8.9	8.5	8.3	20cm
40	5.9	9.1	8.0	8.7	
60	6.0	8.8	7.8	8.9	
20	6.0	8.6	8.6	8.3	50cm
40	6.1	8.6	8.7	7.9	
60	6.0	8.3	8.6	8.1	
20	5.9	5.9	6.0	6.1	70cm
40	5.8	6.3	5.7	5.9	
60	6.2	5.8	6.0	6.0	
20	6.0	5.9	6.0	6.1	100 cm
40	6.1	5.9	6.2	6.2	
60	5.9	5.8	5.8	5.4	

A multiple plot with the summary performance measures for the evaluated running times within each application is shown in both Figures 5 and 6.

It can be seen that the means calculated for each of the experiment describe a inconsistent behavior, where the maximum CPU usage was measured at 70cm between the surface and the mobile phone. The maximum RAM usage was measured at shortest distance between the surface and the mobile phone.

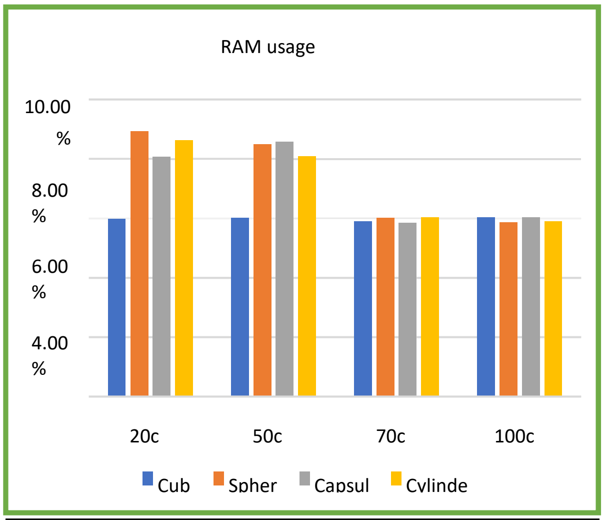

Figure5: Bar plot for CPU values.

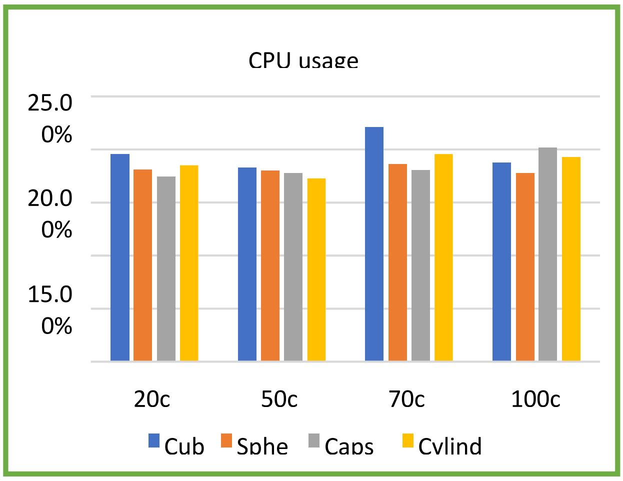

Figure6: Bar plot for RAM values.

6. Conclusion:

In this work we evaluated CPU and RAM performance for Markerless Augmented Reality applications, we used different objects such as (cube, capsule, cylinder, sphere). Four applications were developed, one for each object, and experiments were carried out with a number of times the application was run at different distances. The results were read at different times. Cylinder object had the best performance for the CPU at the shortest distance, while the capsule object was the worst at the longest distance. As for the RAM performance, capsule object was the best, and sphere was the worst at shortest distances. The CPU usage results obtained are show that the objects with fewer number of vertices performs steady and not oscillating. Object was superior to the rest of the others is sphere, which is performs better values. The RAM usage results obtained are show that the cube object performs steady and not oscillating.

Declaration of interests
The authos declare that they have no known competing financial nterests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Ciro A., Franco M., Damián A, Guillermo L. and David C., 2021. CPU and RAM Performance Assessment for Different Marker Types in Augmented Reality Applications.
[2] Baek, A. R., Lee, K., & Choi, H., 2013. CPU and GPU parallel processing for mobile augmented reality.
[3] Konstantin S. S., Alexey A. G., 2016. Analysis of CPU Usage Data Properties and their possible impact on Performance Monitoring.
[4] Qiao, X., Ren, P., Dustdar, S., & Chen, J., 2018. A new era for web AR with mobile edge computing.
[5] Alexandru G., Alexandru O., and Nicolae T.,2013. A Marker-Based Augmented Reality System for Mobile Devices.
[6] Nor F. S., Dayang R. A., 2021. A review of augmented reality systems and their effects on mental workload and task performance.
[7] Su C., Xu W., Feng-Kuang C., 2014. A case study of Augmented Reality simulation system application in a chemistry course.
[8] Jack C.P., Keyu C., Weiwei C., 2017. Comparison Of Marker-Based AR And Markerless AR: A Case Study on Indoor Decoration System.
[9] D.W.F. van Krevelen and R. Poelman, 2010. A Survey of Augmented Reality Technologies, Applications and Limitations.
[10] Mehdi M., Andr´e L.,2014. Augmented Reality: Applications, Challenges and Future Trends.]
[11] Huixiang C., Yuting D., Hao M., Yilun C., Tao L., 2018. Understanding the Characteristics of Mobile Augmented Reality Applications.
[12] Dimitris C., Carlos B., Zhanpeng H., Pan H., 2016. Mobile Augmented Reality Survey: From Where We Are to Where We Go.
[13] Technologies, U. Unity. Unity. Retrieved from https://unity.com/.
[14] Vuforia Developer Portal |. Developer.vuforia.com. Retrieved from https://developer.vuforia.com/.