The 3D virtual drawing mobile application based on augmented reality using AR-Framework

M Safrodin[1], M Zikky[1], S Ghozi[2], M E Wicaksono[1]

[1] Multimedia Creative Department, Politeknik Elektronika Negeri Surabaya, Jalan Raya ITS, Surabaya, Indonesia
[2] The Centre of Research and Community Service, Politeknik Negeri Balikpapan, Jalan Soekarno Hatta, Balikpapan, Indonesia

E-mail: saiful.ghozi@poltekba.ac.id

Abstract. This study discussed the visualization system of 3D sketching model based on mobile Augmented Reality (AR) which supports information needed in real environment. The mobile application was made with adding virtual 3D objects to the 2D drawing by applying technology of AR. The visual system assists those who don't have any specialized skill to be able to draw easily, because 3D model augmented over the drawing through camera of smart phone was added without any marker. This method is the basic combination between dynamic virtual object with the real environment, which nowadays has been developed as the mix reality. As the result on this research, the application can detect the real environment and mark it as a marker for the drawing-place as well. It will become the strategic future research of mix reality or marker less AR, because with this development, virtual world should be more attractive because it has been presented with immersive mixing to the real world.

1. Introduction

A learning media should be provided with creatively and systematically to achieve a great student experience in a study [1, 2]. One of its solutions can be approached by developing learning media with Information and Communication Technology (ICT). One of today's famous ICT learning media which developed continuously is Augmented Reality (AR). Augmented Reality is a technology that can combine the real world with the virtual world, interactive with a real-time, and mostly its virtual objects viewed by 3D animation [3]. AR technology has an ability to display its virtual object into the real world usually using a webcam, smartphones, or even special glasses [4, 5].

In this study, a 3D virtual drawing for basic media learning will be developed using AR framework, namely AR-Kit with swift programming language. The further goal of its application is to assist children in drawing basic objects with three dimensions previews, and it's looks like happened in the real world with a real-time, in other word its scratch will be looked like exist in their surroundings. So, hopefully, by applying Augmented Reality in the learning process of drawing, children can learn easier to describe the shape of the 3-dimensional shape in the real world [6, 7].

2. Related works

2.1. Bare-handed 3D drawing in augmented reality

In this method, there are three types of techniques in drawing shapes, i.e. freehand, tap-line, and GoGo-Tapline. The techniques implemented are using AR interaction. Figure 1 shows the results of pyramid drawing using these three techniques [1].

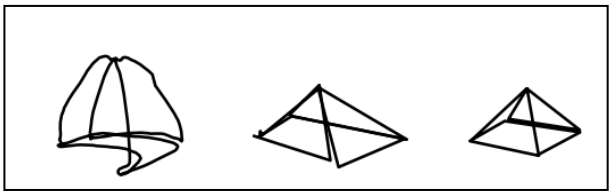


Figure 1. The comparisonresults of AR drawing using three techniques i.e. freehand, tap-line, and GoGo-Tapline.

The study shows that drawing using freehand can be executed very fast, but the result mostly is not accurate. Vice versa, Tapline shows its accuracy and fast execution in drawing, this technique is more interesting than before, and many users preferred tapeline. In this study the freehand use in drawing is very fast though inaccurate. This study highlighted the benefits of the user in switching techniques in a drawing that is not separated from the five principles of design principle, which can form the starting point of the designer to develop other applications [8].

2.2. Augmented reality system drawing visualization

This paper explains the model of 3D visualization system from designing images using Augmented Reality which supports the required information adding 3D virtual objects into the real environment. The visual system contained in AR is very easy to be used because it uses only smartphones without the use of markers [10].

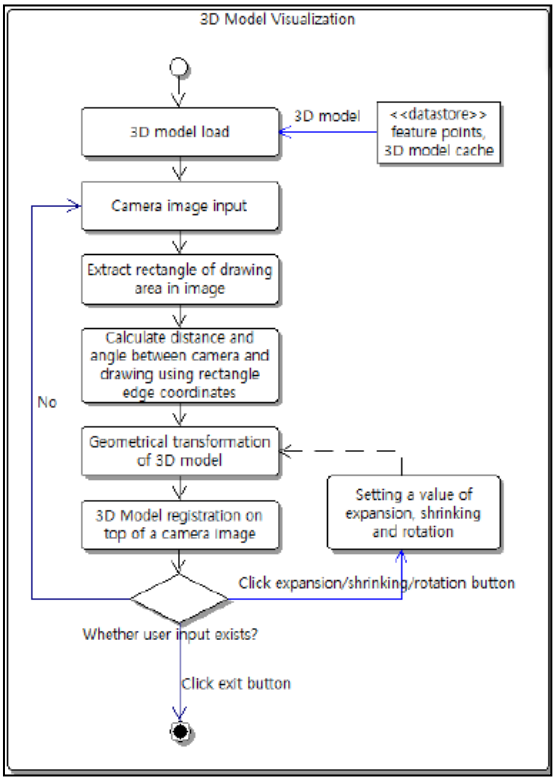


Figure 2. 3D model visualization.

2.3. AR framework

The AR Framework is a framework in creating ARCore (Google) and ARKit (Apple) to provide more AR applications on their platforms. This research experiment will be implemented on ARKit, which is Apple's new framework for app developers in iOS 11 to make AR content much easier and faster. An example of ARKit apps and games that use Augmented Reality are the real-time mixes of camera recordings and animations that come into real-world view.

- Orientation Tracking
  All the settings in AR combine the real world contained in the Device and the Virtual 3D coordinate space where the user can set it up. When an app displays a Virtual image that can converge with the real world it can provide an exciting virtual illusion experience. To create and manage this issue is required Tracking the Device Motion. In AR Orientation Tracking tracks the movement of devices with Three Degrees of Freedom (3DOF) especially in the three-axis rotation of roll, pitch, and yaw.
- World Tracking
  In all of AR Experiences, ARKit uses a real-world coordinate system and cameras that follow the right-hand rules: the y-axis is pointing upwards, the z-axis is pointing forward and the x-axis leads sideways. To create a merger between the real and virtual spaces, ARKit uses a technique called Virtual-Inertial Odometry. This process combines information from a Motion Sensing Hardware iOS device with the Computer Vision Analysis seen on the camera. This technique can produce a high precision model position of the device and movement.

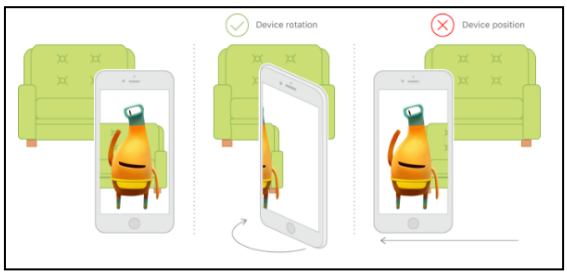


Figure 3. Orientation tracking.

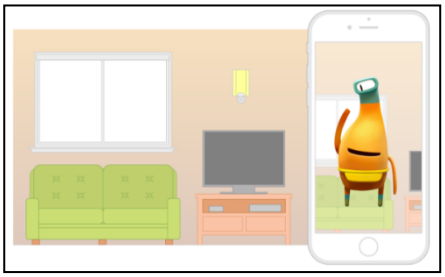


Figure 4. World tracking.

Yaw, Pitch and Roll is a 3D object rotation that can be rotated around three orthogonal axes i.e. x, y, and Z.

- Yaw is the rotation opposite the α of the z axis. Matrix rotation is obtained from:

$$R_z(\alpha) = \begin{bmatrix} cos\ \alpha & -sin\ \alpha & 0 \\ \sin\alpha & cos\ \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

- Pitch is the rotation opposite to the β of the y-axis. Matrix rotation is obtained from:

$$R_y(\beta) = \begin{bmatrix} cos\ \alpha & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix} \quad (2)$$

- Roll is the rotation opposite of γ of the x-axis. Matrix rotation is obtained from:

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\gamma & -sin\gamma \\ 0 & sin\gamma & cos\gamma \end{bmatrix} \quad (3)$$

Yaw, Pitch and Roll rotation can be used to place 3D objects with multiple orientations.

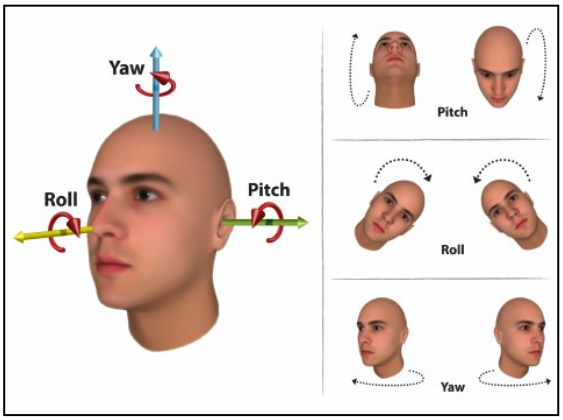


Figure 5. Illustration on roll, pitch and yaw rotation and perspective.

3. System design

3.1. Environment detection

The application focused on AR Kit development. The diagram of system design can be seen in the Figure 6.

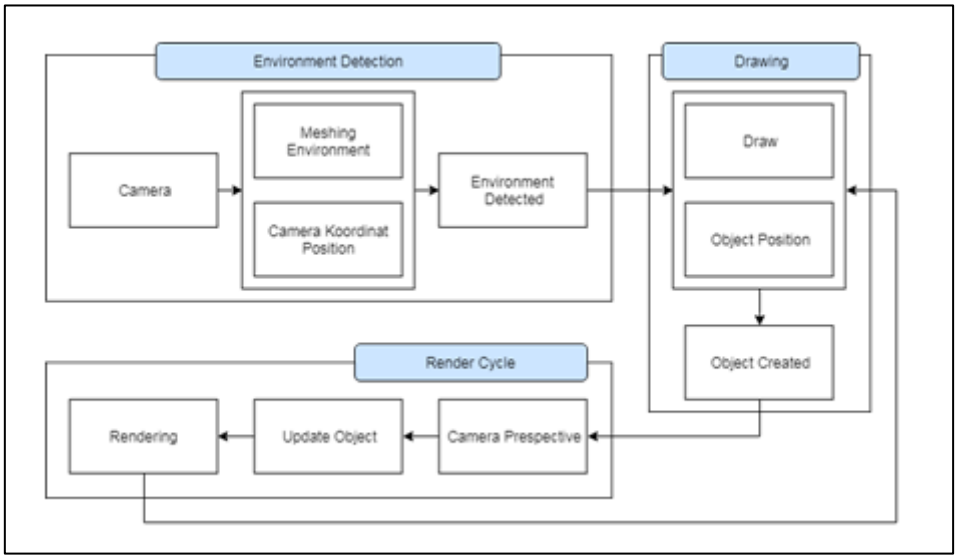


Figure 6. Diagram of system design.

The system starts from environment detection from the camera from the motion tracking sensor (Google)/Orientation Tracking (Apple) which is used to order real-world objects to be the point units that are based on the 3D and Environmental Understanding (Google)/World Tracking (Apple). The objects are used to record the position coordinates of the camera to detect camera movements to set the position of the object so that it is not moving. Then this application detects the input draw of the user that will be recorded position in the environment that has been detected in the first phase. At the last stage, the object will be displayed on the screen and continuously updated when the user enters a new image.

3.2. Drawing and render cycle

Environment Detection is a stage where ARKit detects fields around it and makes it a marker. To detect fields in ARKit we use two main concepts, i.e. Feature Point and Horizontal Field. In short, Augmented Reality on iPhone works with a process called Visual Inertial Odometry (VIO), which retrieves data from the camera and internal sensors to get the 3D unsurpassed in a scene using the feature point. Feature Point helps ARKit get depth from the real world, making the reality much more accurate. Feature Point is also used to help place real-world virtual objects, helping them know where to stay if the device is moving. To display the feature point on the device we need to take one line of code under viewDidLoad ():

```
sceneView.debugOptions=[ARSCNDebugOptions.showFeaturePoints]
```

Figure 7. Feature point.

This application is a 3D Virtual drawing, the most important component of 3D virtual drawing is a drawing feature on this application. The drawing feature in this application uses the concept of the brush as it does in other drawing applications, but the difference of brush on this application using 3D object so that the volume of the Brust itself can be seen and not impressed 2D.To start drawing using this app the user simply presses the draw button on the button located in the middle position at the bottom. Then the user can start to move the device like moving the brush when drawing. This is because the pointer in the middle of the device is placed on the camera position so that when we move the device then the pointer will follow the device.The placement of the adjust to the camera position has a purpose to provide freedom in the drawing. The drawing process starts when the user presses the Draw button. To be able to move the pointer and line according to the camera required matrix to represent the transformation space coordinates, which can represent position, rotation or orientation on the scale of the 3-dimensional object.

```
guard let pointOfView = sceneView.pontOfView else {reurn}
let transform = pointOfView.transform
let cameraOrientation = SCNVector3 (-transform.m31,-transform.m32,-transform.m33)
let cameraLocation = SCNVector3 (-transform.m41,-transform.m42,-transform.m43)
let cameraCurrentPosition = cameraOrientation + cameraLocation
```

The code above serves to determine the camera position by combining the camera orientation with its location. After locating the camera position, the position point is placed on the sceneView so that when the camera is moving the location point will follow where the camera is located.After obtaining the camera position, the next process is inserting a spherenode using the geometry SCNSphere in Swift on the camera.This code also required to move the pointer and line based on the camera required matrix that represents the transformation of space coordinates, where they can represent position, rotation or orientation of the 3D object scale.

4. Experiment and analysis

System trials intend to know whether the built-in system can be well-developed and meet predefined targets.

4.1. Marker detection

The following tests aim to determine if the distance affects the number of feature points appearing on the camera.

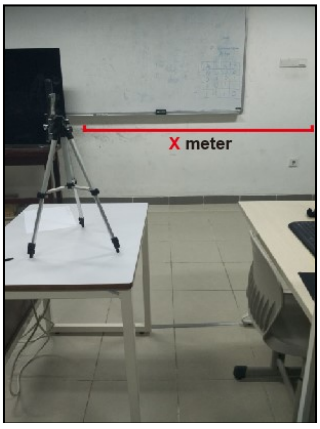


Figure 8. Distance device with marker.

Figure 8 is a testing way to determine the distance of the device to the marker. The distance will determine whether the feature point that appears will be affected by the distance from the device and whether it will affect the stability of the object when inserted into the marker.

4.1.1. Marker test against distance. This test uses the constant parameters used in each experiment. The following Table 1 are the constant parameters used in this test.

Table 1. Constant parameter of distance testing.

Constant parameter	
Device altitude	120 cm
Device position	Vertical
Environment	Door

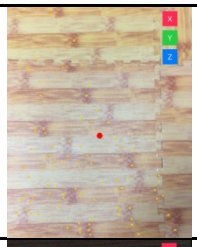
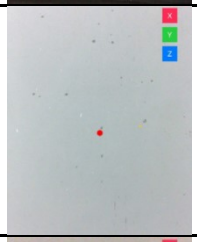
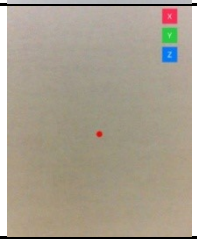
This test uses a door environment. At this test the author took 6 data which were taken based on a multiples distance of 0.5 meters. Once the feature point is detected, the experiment is to put the virtual 3D object in the feature point and analyze whether the object remains stable at that distance. Based on Table 2, there is a result of 6 times of distance testing on the door environment. It can be seen that the feature point was detected when the textures were captured in the object. In the first and second tests, the feature point is still stable at number 10 and above. Feature points start down at a distance of 2 meters and start to disappear at a distance of 2.5 meters. The conclusion of the experiment above is that the distance also affects the number of feature points. The farther distance the device is with the environment then the device will distress to detect the unique point of the object and will look for a unique point nearby other than the object. Objects will also move or be unstable when the feature points are slightly or no. Maximum distance for the device can detect an environment with a shape that is not very unique is 2 meters, but if the environment has a unique shape maximum distance is 2.5 to 3 meters.

Table 2. The result of distance testing.

Distance (meter)	Number of featured points	Object position
1	10	Stable
1.5	12	Stable
2	6	Stable
2.5	0	Not Stable
3	0	Not Stable
3.5	0	Not Stable

4.1.2. Marker test against texture object. The following Table 3 is the result of the test to find out if the pattern of the texture of the objects affects the marker of 3D virtual drawing this application. From Table 3, it can be concluded that the texture of the environment is very influential related to the appearing of many feature points. An environment that has a striking texture will bring up the feature point that many otherwise if the environment does not have textures then the feature point will be difficult to be detected. Because on its basic feature point detects a unique pattern or unique point of each object.

Table 3. Result of texture testing.

No	Result	Description
1.		The texture from Evamat visible feature point detected very much follow the pattern of the texture from Evamat. Each unique point of the Evamat is detected and transformed into a feature point.
2.		The texture of the wooden table the visible feature point detected appears only on the circle of the year and there is little that is detected outside of it.
3.		The texture of iron visible feature point detected very little, feature point detected a number of 1 piece.
4.		The texture from the floor appears to feature the detected point is not present, ceramic does not have a striking the texture like the previous texture, therefore the feature point can not appear in this ceramic the texture

4.2. Drawing test

This test intends to know whether the X, Y, Z position of the line can be stable at a certain value in some constant testing.

Table 4. Constant parameter testing position.

Constant Parameter	
Device Altitude	120 cm
Device Position	Vertical
Sphere Number	10

In the method of testing, the user draws 5 lines with the same device, shape and interval position. Each finish creates a position line to be stored in a text form, the position value taken by a number of 20 spheres that appear on the screen. The result will be compared and look for the average value whether the image created with the constant parameter will be stable at a certain value.

The increase in value due to the created image leads to a X-axis, so the coordinates of each point also increase as the player adds a new point as a line. The X-axis coordinates values are shown as the following graph.

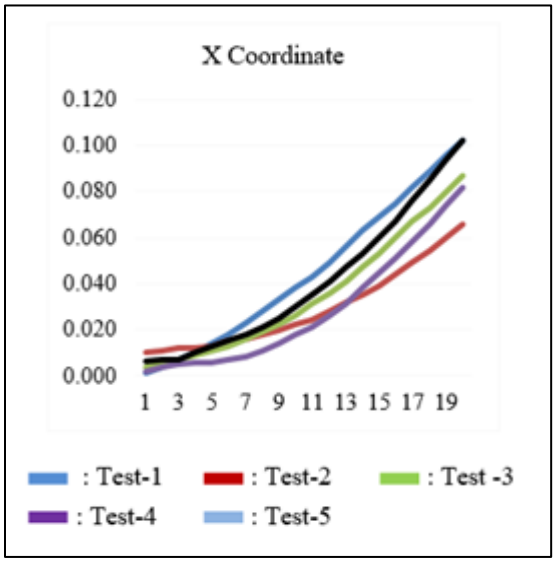


Figure 9. X-Coordinate graph.

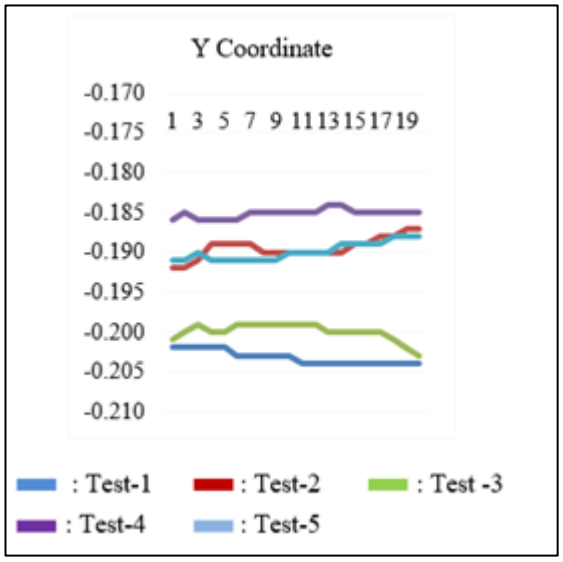


Figure 10. Y-Coordinate graph.

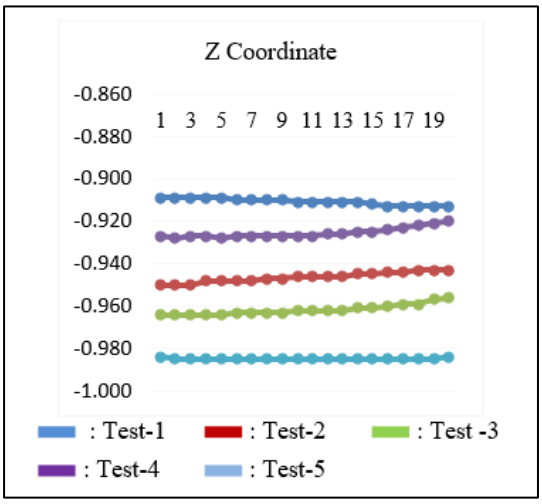


Figure 11. Z-Coordinate graph.

Figure 12. The drawing test.

In the Figure 9, it can be seen a graph of coordinate changes from X-axis through five of tests. From that data it can be concluded that the X coordinate of the created line is incrementally increased, the average line coordinate increment ranges from 0.002 m to 0.004 m and the maximum average of coordinates on the object to 20 is 0.091 m. But there is one trial that a little slow in the second experiment.It is caused by the movement of users who are less similar to other images so that the increase in the second experiment becomes a little slow.

It can be seen from Figure 10 that the value of the Y-axis coordinates is constant and not too much change. The constant value of the Y-axis due to the created image leads to a x-axis, so the coordinates of each Y point do not change. While on Figure 11, it can be seen the value of the Z-axis coordinates are constant and not too much change. The constant value of the Z-axis is because the created image leads to the X-axis, so the coordinates of each z point do not change. From that data, it can be concluded that the z coordinate of the created line has a value change which is not too significant or could be called the average constant change of coordinates of -0.004 m. On the fifth experiment we had the correct true

coordinate value constant, i.e. stable at a value of-0.985, unlike the other four trials that have a slightly rising value. Based on Figure 12, the result of the drawing test was successful. Lines can be formed and attached to the environment around.

5. Conclusions

The application can detect the environment and mark it as a marker for the drawing place. The marker or feature point used is affected by the distance of the device to the marker and is also affected by the texture. This is because the feature point detects the unique point of each object so that the more unique shape or texture of the object will be more and more feature points that appear. The made line has enough stability in the feature point that many changes the coordinate of the position around 0.004 m so as not to affect the line. Further study of this application is personalisation of AR platforms to be based on learners' models using students' learning styles [8].

6. References

[1] Oh Y J, Park K Y and Kim E K 2014 *16th International Conference on Advanced Communication Technology* 1296–1300
[2] Ghozi S and Yuniarti S 2017 *Journal Physics Conference Series* **943**
[3] Nóbrega R, Jacob J, Coelho A, Weber J, Ribeiro J and Ferreira S 2017 *Encontro Português de Computação Gráfica e Interação (EPCGI)* 1–8
[4] Dudley J J, Schuff H and Kristensson P O 2018 *Proceedings of the 2018 Designing Interactive Systems Conference* 241–252
[5] Chen H, Feng K, Mo C, Cheng S, Guo Z and Huang Y 2011 *IEEE International Symposium on IT in Medicine and Education* **2** 362–365
[6] Mota J M, Ruiz I , Dodero J M and Arnedillo S I 2018 *Computer Electronic Engineering* **65** 250–260
[7] Koch C, Neges M, König M and Abramovici M 2014 *Automation in Construction* **48** 18–30
[8] Axon S 2018 *ARC Technican* **8**
[9] Kurilovas E 2016 *Behavior Information Technology* **35** 998–1007
[10] Lanham M 2018 *Learn ARCore - Fundamentals of Google ARCore* (Birmingham – Mumbai: Packt)