An Implementation Review of Occlusion-Based Interaction in Augmented Reality Environment

Mohamad Shahrul Shahidan, Nazrita Ibrahim, Mohd Hazli Mohamed Zabil, Azlan Yusof

College of Information Technology,
Universiti Tenaga Nasional
{shahrul, nazrita, hazli, azlany}@uniten.edu.my

ABSTRACT

Augmented Reality (AR) technology shows some potential in providing new approach of interaction with computer. It shares similar potential in Virtual Reality (VR) but at lower cost. In this paper, an AR application is developed to explore the capability of the interaction approach called Occlusion Based Interaction using low cost device. The implementation of the application is utilizing the ARToolKit library as the main library to handle the AR part while OpenGL and GLUT to handle the graphics manipulation and windows management respectively.

Keywords
Augmented Reality, Occlusion Based Interaction

1.0 INTRODUCTION

The term AR was coined by Ivan Sutherland in 1960's. Since the development of the first AR application, the field has become interesting and challenging. Augmented Reality (AR) has evolved quite rapidly since its introduction more than a decade ago. Many researchers have been looking into finding and perfecting the interface of the AR application.

AR technology may enhance the user perception and interaction with the real world. The technology shows a lot of potential in providing experience to the user to increase the quality and the productivity of task executed in real world. This paper described our experience in developing an AR application utilizing occlusion based interaction approach, and our finding on the approach effectiveness when used with low cost imaging device.

2.0 OCCLUSION-BASED INTERACTION

Occlusion based interaction technique allows the user to interact with the virtual object by totally or partially blocking the marker. Each marker is assigned with an action. By blocking the marker or making the marker invisible, we can identify which action is requested by the user as each marker has been assigned with particular actions. For the project, we are testing the interaction technique by developing house interior environment. The placement and orientation of the virtual object in the environment can be manipulated using a set of marker that is acting as an interaction button. To interact with the virtual object, the user is required to block the interaction button marker. This will allow the object to be reposition or reorientation.

3. 0 IMPLEMENTATION

A simple environment to explore the occlusion-based interaction is developed. The following sub-sections will further elaborate the implementation process of the environment.

3.1 Marker pattern

In deciding the type of marker to use, we have browse for sample marker used in the AR environment. Figure 1 shows some of marker used in AR application as explain in Zhang, 2002.

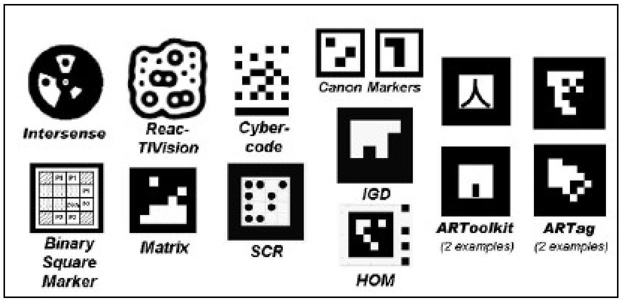


Figure 1: Sample of markers used in AR application (Fiala, 2005)

For our project, we have decided to use the pattern that is optimized by ARToolKit library (HTLab, 2007). From our observation we believe that this is the most reliable pattern in term of providing us with the tutorial, samples and documentation to implement the AR environment. It helps us a lot in exploring the technology as we are very new to such technology.

The implementation of the application involves two sets of marker. The first set is what we call as the *base marker* while the second set is known as the *interaction marker*. Each marker is assigned either an alphabet or a numerical character as its pattern as shown in Figure 2. We simply use alphabet and numerical characters because it is a pattern that is much simpler to produce.

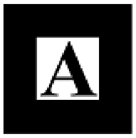


Figure 2: An alphabet character used as the pattern for the marker

This is however leading us to some problem. We observe that the system perceives some of the alphabets and numerical characters to be similar in appearance, for example, numerical 6 and 9. The similarity produce inconsistent pattern identification making the tracking capability becomes unstable. We will further elaborate this problem when we discuss on the interaction marker set.

3.1.1. The base marker set

The base marker is a set of marker used to station the virtual object. We basically used a template marker set provided by ARToolKit (HTLab, 2007) with an additional pattern on it. Figure 3 shows the improvise marker set that we use for the base marker. The alphabets markers come from the template marker and the numeric marker is the additional one added into the base marker.

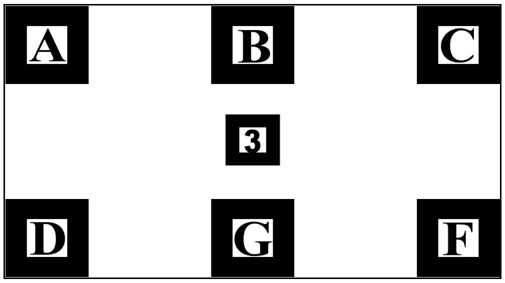
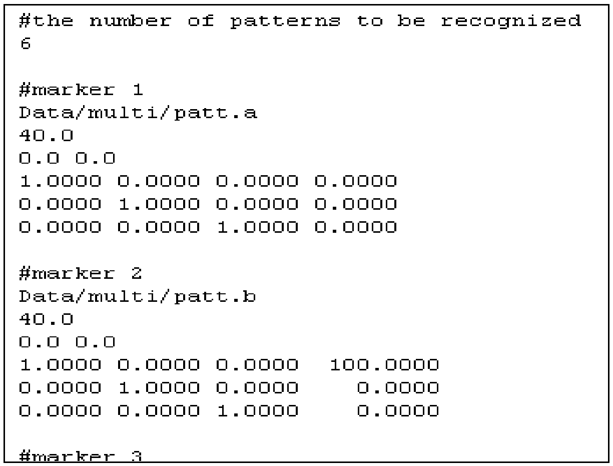


Figure 3 : The base marker set (Shahidan, 2007)

For the base marker, we use a multiple set marker to achieve a much more stable detection capability of the marker. This will help in stabilizing the visibility of the virtual object in the environment. We identify the pattern using multiple approach tracking. This is done by defining the pattern in one external text file rather than calling each pattern directly in the programming code. The content snapshot of the file defining the information about the pattern is shown in Figure 4.

The marker with number '3' as its pattern is an additional marker added in the base marker set. This is a single marker that is used to enable/disable the interaction marker. This marker must always be made visible to allow the interaction to happen.

The reason of using a set of pattern as the base marker is to enhance the stability in tracking the marker by the system. This is because the system is depending on multiple markers to make the object visible on top of this base marker set. This simply means that when one of the markers on the base marker set is occluded, the other 5 markers will still be available to anchor the object. This would allow the virtual object to be visually stabilized on top of the base marker set.



```
#the number of patterns to be recognized
6

#marker 1
Data/multi/patt.a
40.0
0.0 0.0
1.0000 0.0000 0.0000 0.0000
0.0000 1.0000 0.0000 0.0000
0.0000 0.0000 1.0000 0.0000

#marker 2
Data/multi/patt.b
40.0
0.0 0.0
1.0000 0.0000 0.0000  100.0000
0.0000 1.0000 0.0000    0.0000
0.0000 0.0000 1.0000    0.0000

#marker 3
```

Figure 4 : Multiple marker information in external text file

Interaction markers are used to manipulate object projected on this base marker set. Both marker sets must work together to make the whole environment works. The next section will elaborate more on the interaction marker set.

3.1.2. The interaction marker set

The interaction marker set consists of eight markers. Again we are using the pattern type provided by ARToolKit library. However, we customize the patterns so that it meets our preferences. We treat each of these markers as a single marker. Each of these markers is assigned with an action. The pattern for these markers must be unique. Figure 5 shows the pattern that we use for the interaction markers and they are arranged in the following manner.

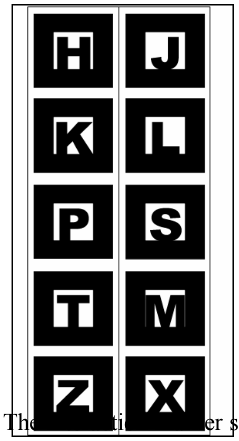


Figure 5 : Th er set (Shahidan, 2007)

We assigned different transformation action on each marker. The assignments are shown in Table 1.

Table 1: *marker pattern and its action*

Marker pattern	Action
H	Rotate clockwise
J	Rotate anti-clockwise
K	Translate left
L	Translate right
P	Translate up
S	Translate down
T	Scale up
M	Scale down

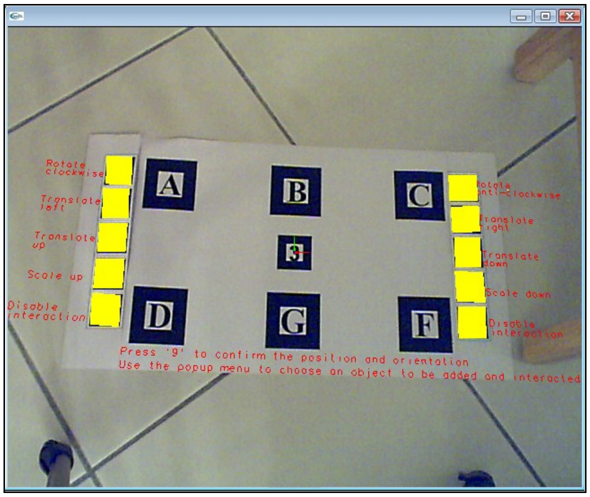
The “Z” and “X” marker are two special markers that activate and deactivate all the interaction markers above them. In other words, to allow the marker “H”, “K”, “P”, and “T” being enabled, the “Z” marker must be visible.

We developed the environment in such a way that marker “3” on the base marker set must be visible all the time to ensure the interaction can be executed. For this purpose we use marker “3” with marker “Z” and “X” to enable or disable the interaction. If we hide marker “3”, all the interaction marker will be disabled, however if we hide either marker “Z” or “X” only the interaction marker above them will become disable while the rest remain enable.

The reason of using the special marker to test the activation and deactivation is because of the visually instability for the interaction marker. The system had a difficulty to maintain the visibility of the pattern in the system making the interaction marker active automatically even though we do not want it to be active. By introducing the “Z” and “X” marker we can basically control the marker activation and deactivation.

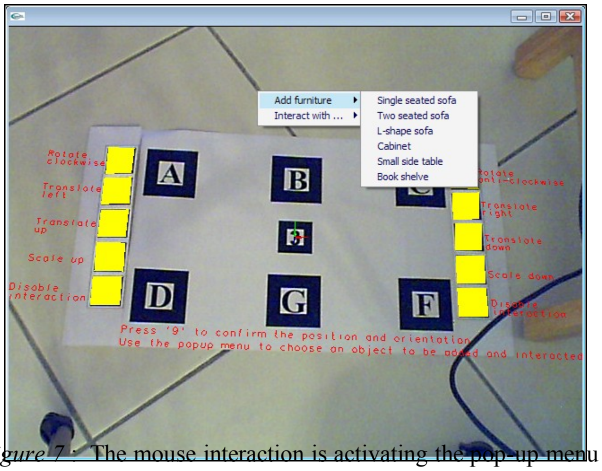
4.0 AR ENVIRONMENT MANIPULATION

This section illustrates the snapshots of the AR environment manipulation. It shows all the interaction approaches in the application which are the mouse interaction, the keyboard interaction as well as the marker interaction. Figure 6 shows the initial environment setting. It shows the environment before we add any object to be manipulated. This is the first visual that the user will see when executing this AR application. In the environment the user can see the label for each interaction marker as well as the instruction to execute command in the environment.



Press '9' to confirm the position and orientation
Use the popup menu to choose an object to be added and interacted

Figure 6 : The AR environment before any virtual 3D objects is loaded

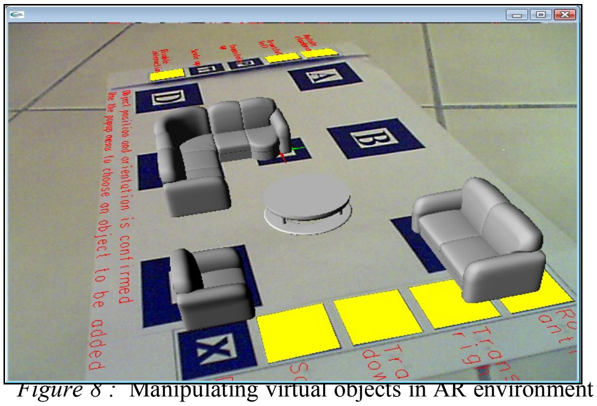


Press '9' to confirm the position and orientation
Use the popup menu to choose an object to be added and interacted

Figure 7 : The mouse interaction is activating the pop-up menu which allows users to add and to select an object for interaction

To interact in the AR environment, the user needs to add in the 3D object first. We use furniture as the objects to be manipulated. To add 3D objects into the environment, the user needs to use their mouse. Right click to activate the pop-up menu and choose an object to be loaded into the environment as shown in Figure 7. Since the system allows more than one object to be uploaded, the users need to right click the mouse again to activate the pop-up menu and choose the object that they want to manipulate. After an option is selected, the user can manipulate the virtual object using the interaction marker.

By blocking one of the markers, the chosen object will be manipulated depending on the action set on that particular marker, as shown in Figure 8.



Object position and orientation is confirmed
Use the popup menu to choose an object to be added

Figure 8 : Manipulating virtual objects in AR environment

5.0 EXPERIMENT

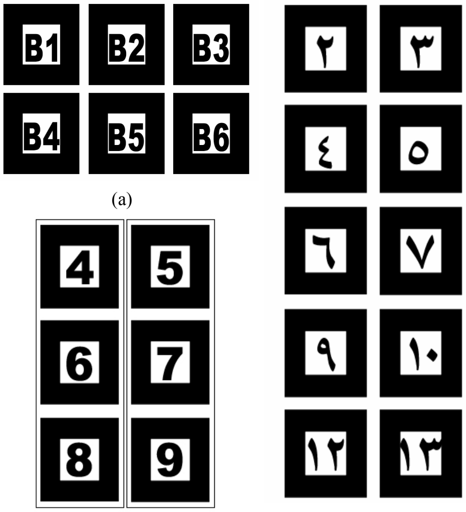
We are using two types of web camera for the experimentation of the project. We categorized the camera into two category; low-end-low-cost web camera, and middle-end-middle-cost web camera. Table 2 shows the differences between the two cameras used in the project.

Table 2 : *Comparison of low cost and middle cost web camera used for the project.*

	Low cost web camera	Middle cost web camera
Name/ Brand	USB PC Camera-168	Logitech QuickCam Pro 5000
Price	RM 45.00	RM 350.00
Features	1. Standard web cam features 2. No auto-lighting capability 3. No auto-focus capability 4. Low quality images	1. High quality VGA sensor with RightLight Technology 2. Support auto-lighting and auto-focus 3. True 640 x 480 pixels 4. 1.3 megapixel still images

The reason we are testing the developed environment with two types of cameras is because we would like to know if the capability of the camera will affect the stability of the application. We would also like to know how well the image processing algorithm in ARToolKit handling the capturing and identification process in different hardware setting. This will be very useful in helping us to determine how to improve our implementation on the project.

Two experiments have been set up with two different groups of user. The first group is working with the low-end-low-cost web camera, while the second group is working with the middle-end-middle-cost web camera. Each group is asked to perform certain function within the AR environment, such as moving the furniture, resize the furniture and rotate the furniture. For each experiment, several type of interaction marker has been used. The testing is done to identify the best marker which can give us the best detection features (detection visibility test) and visually unique (uniqueness test). Figure 9 shows some of the pattern that we have tested for the interaction marker.



(a)

(b)

(c)

Figure 9 : Experimented patterns for the interaction marker **[4]**

From the experiment that we have done, some of the tested patterns fail the detection capability test while most of the patterns fail the uniqueness test.

Detection capability test is conducted to investigate the system ability in detecting the marker. Markers in figure 9(a) are some of the samples that fail the detection capability test. The patterns used for the marker is very small because it consist of two character. The small size pattern makes the system difficult to detect it. Although we can enlarge the marker pattern to solve the problem, for our case this is not practical as we need to fit the entire marker used in a small capture area. The best solution for the problem is to improve the detection algorithm instead of adjusting the marker size.

The uniqueness test is conducted to analyze the system ability to differentiate between two closely identical patterns, such as 6 and 9. Markers in figure 9(b) and 9(c) are some of the sample patterns that fail the uniqueness test.

In figure 9(b), markers with “6” and “9” patterns fail the uniqueness test. These patterns are similar from the system point of view. When detecting a pattern on marker, the system will store four orientations of the pattern in the data file. Figure 10 shows the character orientation as recorded into the data file. The method of recording the pattern does not allow the number “6” and “9” to be identified as a different pattern.

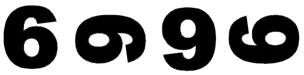


Figure 10 : Four orientation of pattern recorded in the data file

Pattern in Figure 9(c) basically shows the same problem as in Figure 9(b). Pattern and , and look similar from the system point of view.

From the experiments conducted, we observe that the middle-end-middle-cost web camera shows better detection of the pattern compared to the low-end-low-cost web camera. This approach however is still not very effective in terms of solving the problems but it does show us that improvement can be made by using better camera, in terms of visual stability and pattern detection capability of the system.

6.0 CONCLUSION

The occlusion based approach as explain in this paper is referring to an interaction which is optimizing the marker occlusion in executing the task in the environment. The approach works by hiding some part of the marker to allow the user to interact with the object in the environment. Each interaction markers are assigned with specific actions. By hiding a particular marker from the view of the camera, the user will be able to interact with the AR environment.

We observe that the patterns used for the marker must be unique from the system point of view. We also observe that using better quality camera would enhance the system ability to capture better images, hence improve the system stability in detecting pattern.

In the future we will improve the interaction approach by improving the tracking capability as well as removing the mouse and keyboard interface from the environment. The standard in AR interaction is still under intensive research. To produce a standard for interaction in AR environment would require a review on all interaction approaches by other researchers in related field.

7.0 REFERENCES

Fiala, Mark (2005). "ARTag, a Fiducial Marker System Using Digital Techniques", *IEEE Computer Society Conference, CVPR 2005,vol. 2,* Canada, pp 590-596.

HIT Lab NZ. "ARToolkit Online Documentation", Retrieved 18 March 2007 from http://www.hitl.washington.edu/artoolkit/..

Nate, Robin (2007), "Nate Robin OpenGL Tutors", Retrieved 23 March 2007 from http://www.xmission.com/~nate/tutors.html.

Shahidan, M. S. (2007). "Interface and Interaction Technique for 3D Object Exploration in Augmented Reality Environment", in *Master Thesis,* UPM, Serdang, Malaysia.

Zhang, Xiang, Fronz S. and Navab N. (2002). "Visual Marker Detection and Decoding in AR Systems: A Comparative Study", In *Proceedings of IEEE ISMAR'02*, Darmstadt, Germany, pp 97-106.