

# An Exploratory Study of 3D Interaction Techniques in Augmented Reality Environments

Iyad AlAgha\*, Raed Rasheed

Department of Software Development, The Islamic University of Gaza, Palestine.

\* Corresponding author. Tel.:00972592542727; email: ialagha@iugaza.edu.ps

Manuscript submitted October 10, 2014; accepted December 15, 2014.

doi: 10.17706/jsw.10.4.427-440

---

**Abstract:** Augmented Reality (AR) interfaces typically involve the overlay of virtual imagery onto the real world. 3D interaction with the virtual scene is a key feature to explore the full potential of AR. Despite the large number of interaction techniques that have been proposed to enable for intuitive 3D interaction in AR environments, little effort has been done to compare these techniques. In addition, many techniques have been arbitrarily used without formative evaluation or without considering the requirements of different AR applications. This paper discusses three interaction techniques with 3D content in AR environments and reports on an experiment that we conducted to compare them. These techniques are: 1) manipulation of hand-held fiducial markers, 2) free hand interaction, and 3) a keypad controller. 18 participants were recruited and asked to perform a series of tasks which involved the manipulation of a 3D object. The study sheds the light on the strengths and limitations of the studied techniques and when it is appropriate to use each of them. We believe that the reported results will help inform the design and customization of 3D interaction techniques for AR applications.

**Key words:** Augmented reality, 3D interaction, hand gestures, handheld fiducial markers.

---

## 1. Introduction

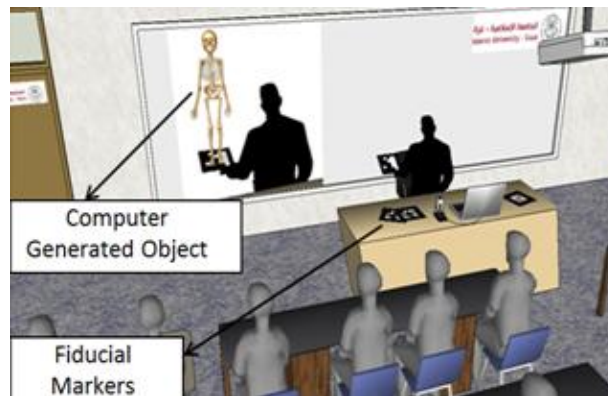
Augmented Reality (AR) is an emerging technology that enables to augment the real world with computer generated content. In simple terms, AR allows digital content (e.g. 2D and 3D objects) to be overlaid and integrated into our perceptions of the real world. In AR, users look at the live image of the video camera (i.e. the reality) and the scene that they see is augmented by computer-generated objects. Individuals can seamlessly perceive the real-world scene, along with the added objects, as a single environment. Although the concept of AR technology has been introduced many years ago, only recently has AR become accessible to the masses. The popularity of hand-held devices and tablets allowed AR applications to be common in many aspects of our lives.

Hand-held fiducial markers have been widely used in AR systems (see Fig. 1.a). The performance of the AR system depends on the tracking of the visual marker and its pose information (e.g. position and orientation). Once the marker is detected by the vision system, the virtual object will be embedded inside the scene and overlaid over the marker. Subsequently, any transformations applied on the marker will be applied on the associated object. Knowing that, the user can manipulate the virtual object by simply manipulating its marker. It is assumed that such an AR setting offers tremendous potential for applications in areas such as cultural heritage, entertainment, education and tourism. For example, Fig. 1.b depicts our perception of an AR system as integrated into lecture rooms to support interactive learning: While a teacher is holding and manipulating a fiducial marker, a special tracking system augments the LCD projected view to integrate a 3D learning model. As the teacher

manipulates the fiducial marker, the pose of the 3D model will be updated respectively.



(a). Fiducial markers.



(b). Our perception of augmented Reality as integrated into lecture rooms.

Fig. 1. Augmented Reality by using fiducial markers.

With the great advances that have been achieved in AR technologies, there has been a considerable interest in facilitating user interaction with the AR content. AR applications should enable users to interact with virtual objects in an intuitive and feasible fashion. Previous research on AR has presented various interaction techniques including the use of tracked objects [1], pen and tablet [2], and mouse and keyboard. Despite the large number of interaction methods that have been proposed, little effort has been done to compare these techniques. In addition, many manipulation techniques have been arbitrarily implemented without formative evaluation or guidelines. Considering the growing attention in AR technology in our daily life, it is necessary to investigate interaction metaphors that offer the best support and ease of use for end users.

This paper reports on an experiment that we conducted to compare three different techniques to interact with 3D content in AR environments. These techniques are: 1) manipulation of hand-held fiducial markers, 2) free hand interaction, and 3) the keypad controller. These techniques were chosen because they are commonly used for interaction in AR applications [3]-[5], and thus should be carefully investigated to reveal their strengths and limitations. In addition, it should be noted that the settings of AR systems can significantly vary based on the supported display technology (e.g. handheld devices, tabletop display, head mounted displays), whereas each of these technologies has different interaction modalities. Our investigation is limited to AR systems that are based on hand-held fiducial markers and where the AR scene is projected on walls or screens such as the system shown in Fig. 1.b. This particular setting was used because it is easily-configurable and widely adopted by the common AR applications in education and entertainment, and thus may be demanding to explore the best techniques to interact with 3D content in this setting. In our experiment, users were recruited and asked to perform a series of

tasks which involved the manipulation of 3D content in an AR environment. Based on the results obtained from our study, we discuss a set of design recommendations for the developers of AR applications. We believe that these recommendations will help to inform the design and customization of interaction methods for AR applications.

## 2. Related Word

This work comes at the intersection of two research areas which are discussed in the following sub-sections:

### 2.1. Interaction Techniques for AR environments

A sizeable amount of research has explored the manipulation of virtual objects in AR environments. Volda *et al.* [6] presented a study on the manipulation of 2D objects in AR environments. They compared different interaction techniques including voice commands, touching, pointing and grasping gestures. Other efforts have presented and evaluated a variety of techniques to interact with 3D objects in AR applications [7-9]. Most of these efforts, however, focused on the selection or grasping of objects without considering 3D transformations (e.g. rotation, translation and scaling). In addition, the proposed techniques were applied to AR environments based on head mounted displays where the user's hand is often represented by a virtual hand floating in 3D space. However, it is not clear whether these techniques are appropriate for the projector/camera-based AR systems where the AR scene can only be overlaid on walls or surfaces of furniture.

To ease the user interaction with AR content, some studies proposed the use of physical objects or props such as paddle-, cup-, or box-shaped props, to interact with 3D objects [10], [11]. The markers, which are associated with the pose of the 3D objects, are fixed on these props or physical objects.

With the enormous use of touch devices such as smart phones, tablets, and tabletop displays in the past few years, there has been an increasing interest in developing AR applications that are tailored for touch devices [12]-[14]. Since the characteristics of touch devices provide only 2D data for manipulation, researchers have been exploring a wide range of hand-based interaction metaphors to adapt touch gestures for 3D object manipulations [15].

While the focus in the previous studies was on the design of interaction techniques, which are often customized for the hardware or device characteristics, our work reports on an empirical study that compares different interaction techniques using hand-held markers. The purpose is to explore how users can most naturally interact with 3D projected objects using hand-held fiducial markers.

### 2.2. Interaction Techniques in 3D Space

In the field of 3D user interfaces, a number of research projects have studied and compared handed selection and interaction techniques in 3D space. For example, Ulinski *et al.* [16] proposed two-handed selection techniques for volumetric data visualizations. They compared between bimanual symmetric interaction, in which both hands perform the same action, and bimanual asymmetric interaction in which hands perform different actions. Stellmach *et al.* [17] described an investigation of basic hand gestures for the exploration of large information spaces. A set of pan-and-zoom alternatives using two-handed gestural controls has been implemented and compared using Google Earth. Malik and Laszlo [18] presented a vision-based input device that allows for two-handed interactions with desktop PCs, laptops, or large wall displays. Lee *et al.* [19] presented a 3D vision-based natural hand interaction method based on skin color segmentation, feature point finding and hand direction. They also described how their hand tracking approach was applied in various AR applications.

Despite the various studies that explored the handed-interaction in 3D space, AR environments have some special characteristics that pose additional requirements for user interaction: In AR environments, the position and orientation of the virtual object is often associated with the marker tracked by a vision system, and the user interacts with the virtual object by manipulating its associated marker. It is unclear whether the existing interaction techniques in 3D space are still applicable in an AR environment with hand-held markers. This work explores how the user can manage the manipulation of markers to interact with 3D objects using

handed-interaction.

Hand gesture recognition has been an active research field for a long time, and many different approaches can be found in the literature. Many different types of visual features have been proposed for hand gesture recognition. One type of features is based on the skeleton structure of the fingers [20], [21]. Other approaches are based on the segmentation of hand regions [22], [23]. In this work, we adopt the second approach where a depth camera was used to identify and segment the hand region.

### 3. Interaction Techniques with 3D Content in AR Environments

This section presents the three interaction techniques we propose for interaction with 3D content in AR environments and briefly explains their implementations. The potential limitations of these techniques are discussed from our perspective. Finally, we report on the experiment conducted to compare these techniques.

The prototype AR system used in this work was implemented using NyARToolkit<sup>1</sup>, a Java-based software library for prototyping AR applications. For the imaging processing tasks and interaction with the Kinect sensor, we used javaCV<sup>2</sup> and Java wrappers for the Kinect libraries<sup>3</sup> respectively.

#### 3.1. Manipulation of Hand-Held Markers

This is the simplest form of interaction in AR environments in which a user can inspect a virtual object by manipulating it to a reference fiducial marker in the real world using bare hands. The role of the AR system is to establish the camera pose relative to this marker in order to render the object correctly in 3D space (see Fig. 2).

However, this interaction mode has some restrictions: the pattern of the marker should be constantly visible by the AR system so that it can render the virtual object correctly. If the marker is occluded or obscured while being manipulated, the virtual object will not be rendered correctly. This adds cognitive load on the user as she needs to continually think of the visibility of the marker. The pattern of the marker may also become invisible if it is manipulated to certain positions or angles. For example, if the marker is positioned horizontally relative to the camera, the process of pose estimation will be impossible. Another limitation of manipulating hand-held markers is the difficulty of handling some geometric transformations. For example, scaling the virtual object up and down (e.g. stretching or shrinking the object) can be done by moving the marker towards or away of the camera view. However, this may require the user to move along distance away or close to the camera so that the marker's size, and hence the object's size, changes significantly. Similarly, it may be difficult to rotate the object 360 degrees due to the nature of hand wrist.

It is clear from the above discussion is that the direct manipulation of markers, while being intuitive and commonly used in AR applications, poses certain restrictions for the free interaction with objects in the 3D space. Thus, it is necessary to explore other interaction methods to overcome these restrictions.

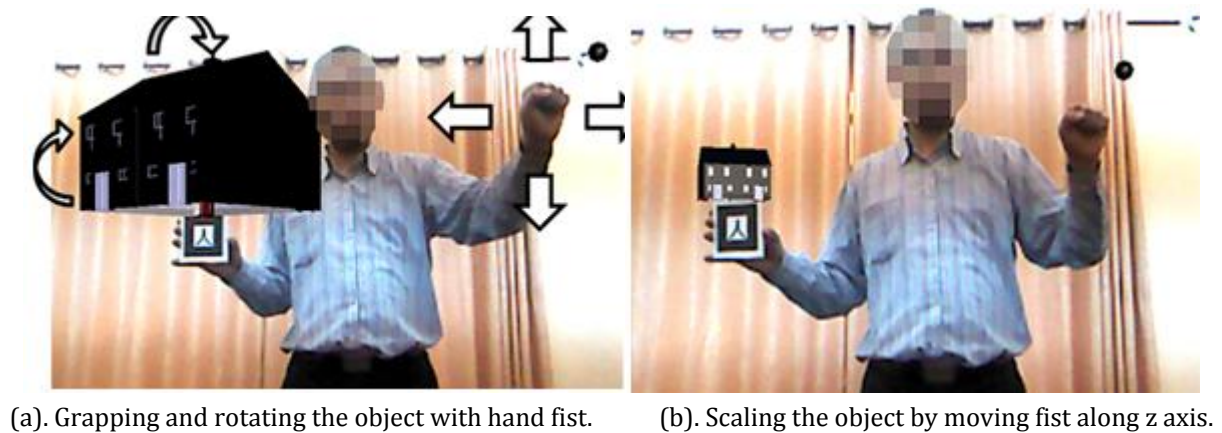


Fig. 2. Interaction with AR content using a hand-held marker

<sup>1</sup> <http://www.artoolworks.com/products/open-source-software/nyartoolkit/> [Accessed on 8/20/2014]

<sup>2</sup> <https://code.google.com/p/javacv/> [Accessed on 8/20/2014]

<sup>3</sup> <http://fivedots.coe.psu.ac.th/~ad/kinect/> [Accessed on 8/20/2014]



(a). Grasping and rotating the object with hand fist. (b). Scaling the object by moving fist along z axis.



(c). Releasing the object by opening the hand fist.

Fig. 3. Free hand interaction with 3D content.

### 3.2. Free Hand Interaction

To overcome the limitations of direct manipulation of markers, we explore an approach that is based on two-handed interaction: while one hand is used to hold the marker and keep it steady, the other hand is used to apply 3D transformations on the object. The user grabs the marker with one hand. This is where the virtual object will be rendered and positioned. Meanwhile, the other hand can be used to rotate/scale the virtual object as the following (see Fig. 3):

- Close/Open hand fist to start or stop manipulating the object.
- Move hand left or right to rotate the object in yaw (around the y axis).
- Move hand up and down to rotate object in pitch (around the x axis).
- Move hand towards and away from the camera to scale the object up and down respectively (move along the z axis).
- Move the marker in any direction to reposition the object.

The proposed hand-based interaction imposes a separation between two tasks: the visibility of the marker's pattern, which is essential to construct the AR scene, and the 3D interaction with the object. This separation is achieved by assigning each task to a different hand. We oversee that this separation allows the user to focus on the interaction with the object without worrying about the visibility of the marker. The marker remains permanently visible as the user does not need to rotate or scale the marker. Thus, the rendering of the virtual object will not be disrupted. Meanwhile, the virtual object can be fully rotated and scaled by moving the other hand along x, y and z axes, without worrying about the visibility of the marker. In addition, unlike the direct manipulation of markers, the 3D transformations are not constrained by the limitation of the human wrist.

To avoid making the virtual object responsive to unintentional hand movements, the above hand actions will take effect only if the hand is closed. Performing the above gestures while closing the hand will cause the virtual object to respond respectively, while opening the hand will release the object. Using the "grab" and "release"

gestures to enable or disable object manipulation is essential to avoid the ambiguity resulting from moving hands for other purposes.

It is obvious that the proposed free-hand interaction requires efficient and accurate tracking of hand movement as well as detection of hand posture (open/closed hand). While the literature in computer vision proposed different techniques for hand detection and tracking (see [24] for a survey of these techniques), most of these techniques lack the desired ease of use and robustness. To address the drawbacks of previous approaches, this work benefited from the recent advances in camera technologies by using the depth camera of the Kinect device, which can achieve robust and accurate tracking of the human skeleton and the joints of human body in real time. Once the skeleton is detected, we can identify the hand joint and its location within the camera scene.

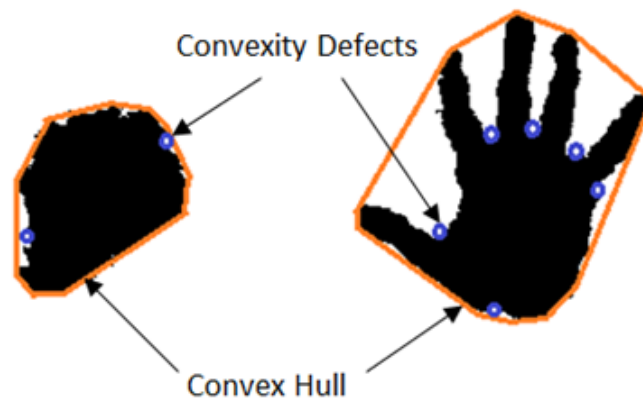
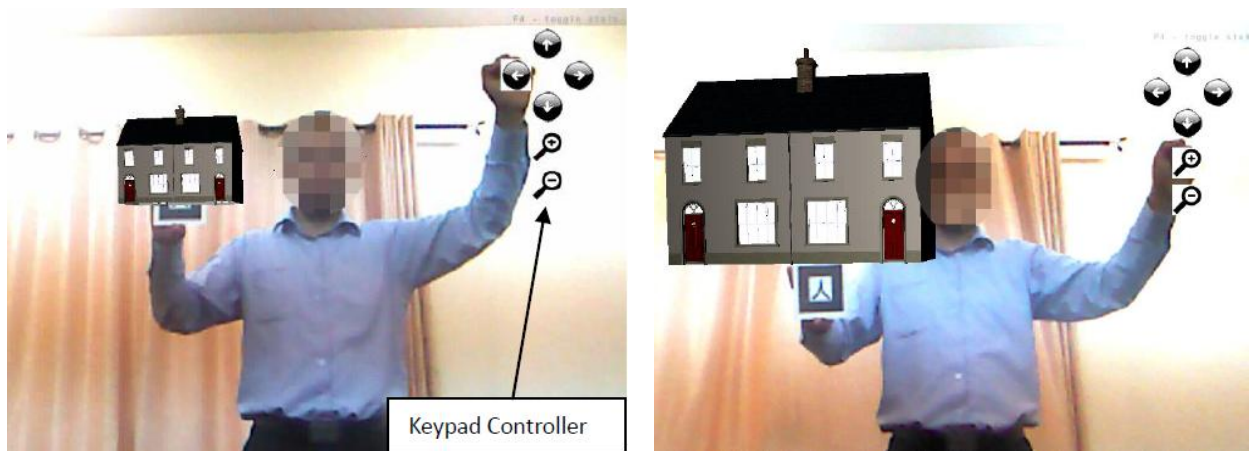


Fig. 4. Hand regions as detected by our depth camera (fist and open hand).



(a). Arrow buttons to rotate object around x and y axes. (b). Zoom buttons to scale object up and down.

Fig. 5. Interaction with AR content using a keypad controller.

Our approach also utilizes the capability of the Kinect camera in detecting 3D depth information to enable the movement of the object along the z axis. The camera provides information of the location of hand joints in the 3D space. Therefore, the virtual object can be translated along the z axis by moving the hand joint close or away from the camera.

Until the time of writing this paper, the available Kinect interface does not support finger detection or open/closed hand. Therefore, we used the following unsupervised technique which utilizes the depth information and the detected skeleton joints to recognize open/closed hands as the following:

- Hand segmentation: Hand segmentation deals with separating the user’s hand from the background in the image. We first used the depth information obtained from the depth camera to plot the depth histogram. A

depth histogram typically shows the pixels' distances from the camera. By identifying the depth at which hand joint is located using the Kinect sensor, we can then segment the hand region by extracting points located at that depth. Fig. 4 shows the result of depth-based hand segmentation. Hand segmentation using depth information provides better results than other techniques that are based on color or intensity-level thresholding [25].

- Perform connected-component labeling: regions extracted from Step 1 may contain, beside the hand segments, other objects that are located at the same depth. To filter these objects out, we perform connected-component labeling, which is an operation that classifies connected pixels as disjoint objects with unique identifiers. Hand segments are the connected regions whose centroids are nearby the hand joints of the skeleton. Other connected regions whose centroids are far from the hand joints by a predefined threshold distance are ignored.
- Apply convex hull and convexity defects methods: The convex hull of a set of points is the smallest convex set that contains all the sets of the given points. Convex hull is drawn around the contour of the hand, such that all contour points are within the convex hull. Subsequently, we detect convexity defects. A convexity defect is present wherever the contour of the object is away from the convex hull drawn around the same contour (see Fig. 4). Convexity defect gives the set of values for every defect in the form of vector. This vector contains the start and end point of the line of defect in the convex hull. The hand status (closed/open) can then be determined by counting the number of large defects that result from the gap between fingers. A closed hand contains less convexity defects than an open hand.

### **3.3. Keypad Controller**

Another approach we present to interact with 3D objects in AR environments is by using a keypad controller along with the hand-held fiducial marker. A keypad, which also resembles the use of a standard keyboard, is shown in Fig. 5, and consists of a set of virtual buttons rendered on the scene: hovering the hand fist over the left or right buttons causes the object to rotate about the y axis while hovering the hand fist over the top or bottom buttons causes the object to rotate about the x axis. The zooming buttons are used to scale the object up and down by moving it along z axis. Similar to the free hand interaction, the rotate and scale transformations are independent of the marker which should be permanently visible. The interaction with the keypad can be done using a desktop mouse, which is useful if the user interacts with the AR scene using a standard PC. Alternatively, the user can use the keypad from a distance by pressing/unpressing buttons using hand actions. Approaching the hand fist to any button and closing it will apply the corresponding action on the object. Hand-based interaction with the keypad releases the user from the restrictions of desktop environments and allows for remote control of the virtual object. The user can use one hand to hold the marker while using the other hand to interact with the keypad.

The intention of proposing the keypad- based interaction is to resemble the use of traditional keyboard and mouse controls, and to compare this interaction paradigm with other techniques. Our hypothesis is that while the keypad approach offer less natural interaction, it allows for more fine-grained and precise manipulation of the controlled object. For example, a simple button action may cause the object to rotate by a 1-degree angle. Such a precise rotation is difficult to perform when hand is used as a rotation device.

## **4. Experiment**

In the following sections, we report on an empirical study that compares the user experiences when using the three techniques discussed above. We conduct the study with the following questions in mind:

- What are the advantages and limitations of each technique? and what are the user preferences?
- Are there particular situations where some techniques may be preferred over the others?
- How can these different techniques inform the design of AR applications?

### **4.1. Participants**

18 volunteers (16 males, 2 females, aged between 19 and 27) participated in the study. All of them were recruited from the local university and were right-hand dominated. They were all frequent-computer users but no one had prior experience with AR applications.

## 4.2. Apparatus

Equipment used for our experiment is shown in Fig. 6. We used a 42 inch TV, which was connected to a PC, to view the AR generated scene. For object tracking, we used a Kinect camera that was mounted in front of the TV. The virtual object we used was a 3D model of a house (see Fig. 2). The object is associated with a fiducial marker with a specific pattern. During tasks, each participant was asked to stand 2.5 meters away from the TV (the practical ranging limit of the Kinect sensor is 1.2-3.5 meters) and raise the marker until the virtual object is detected and rendered on the AR scene.

We have chosen this experimental setting because it resembles the commonly-used AR setting for many applications. In this setting, the user stands in front of a camera which continuously tracks predefined patterns and renders virtual content inside the scene. The augmented scene is then projected on a wall or displayed on a screen.

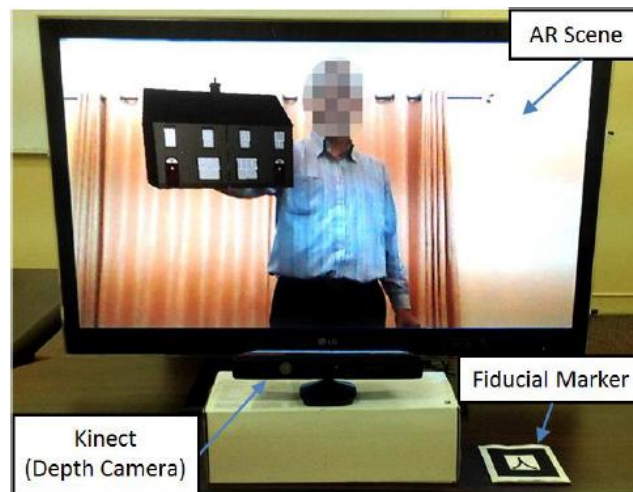


Fig. 6. Experimental apparatus.

## 4.3. Tasks

The tasks were planned to explore the degree to which users of a marker-based AR system can effectively manipulate virtual objects in the 3D space. The virtual object used for our experiment is the house model shown in Fig. 6. Each participant should hold a fiducial marker until the house model is successfully integrated into the AR scene.

Each participant was instructed to perform two different tasks in sequence, each of which required manipulating the house model in a different way: the first task required the participant to rotate the house model 360 degrees around the y axis in counter-clockwise direction. The second task involved a combination of rotate and scale operations (scale the model to twice its original size, rotate it 45 degrees around the x-axis in clockwise direction, followed by rotating it 90 degrees around the y axis in clockwise direction). When the model reached its target position, orientation and scale, it was frozen and a message indicating the end of the task was displayed.

Note that the first task required executing a single 3D transformation while the latter required the execution of multiple transformations. The intention was to observe and analyze the execution of both single as well as multiple 3D transformations. Previous research [26] indicated that while users may perform a single transformation (e.g. rotation) easily, they may encounter difficulty when combining multiple transformations (e.g. a sequence of rotate, scale and translate actions).



At the beginning of each task, participants were shown the target status of the house that they should achieve, but without being told the operations they had to do. The goal was to explore the extent to which participants can find out the sequence of actions they should perform.

#### 4.4. Experimental Design

We used a within-subject, full-factorial repeated measure design for the experiment. For all tasks there was a single independent variable which is the interaction mode (manipulation of markers, free hand interaction and keypad controller). Each participant performed every task using the three interaction modes. There were 3 trials arranged for each task. Thus, a total of 3 (tasks)  $\times$  3 (interaction modes)  $\times$  3 (repetitions) = 27 test trials were collected from each participant.

The order of interaction modes was counterbalanced across participants. For the three interaction modes, there were 6 possible orders. Therefore, participants were split into 6 groups of three. Each group performed the tasks using a different order (for example, one group performed the tasks starting with the marker mode, followed by the free hand interaction, ending with the keypad controller). The aim of creating a group for each possible order is to avoid the order effect. After completing each of the three tasks, each participant rated the three interaction modes according to their overall feelings. Before the experiment started, the three interaction modes were demonstrated to all participants. Participants were also given time to practice with the interaction modes.

#### 4.5. Metrics

We used the following metrics to compare between the three modes:

- **Completion Time:** this is the time elapsed between the beginning and the end of interaction with the 3D object. The trial does not end until it has been successfully completed.
- **Number of Deviations:** A “deviation” occurs when a participant performs a wrong 3D manipulation that causes the object to miss the desired transformation or disappear from the scene. For example, a participant may rotate the object about a wrong axis, scale the object down rather than up or apply useless manipulations that do not contribute towards the task completion. A deviation also includes any action that hinders the vision system from continually tracking the marker. For example, if the marker is positioned so that it becomes invisible by the camera, this will cause the attached virtual object to disappear. The number of deviations provides indication of the difficulty of the interaction mode and the dispersion it causes.

### 5. Results

We performed a one-way, within-subjects ANOVA for each task.

Table 1. Completion times in tasks 1 and 2

Mode	Task 1 Completion Time (in sec)		Task 2 Completion Time (in sec)	
	Mean	SD	Mean	SD
Manipulation of Marker	3.6	0.7	9.72	1.7
Free Hand Interaction	5.2	1.31	10.47	1.81
Keypad Controller	8.04	1.15	15.7	1.83

#### 5.1. Completion Time

Table 1 and Fig. 7 summarizes the average completion times for the two tasks. For the first task, ANOVA showed that the interaction mode had a significant effect on the average completion time [ $F(1,23)=102.95$ ,  $p<.0001$ ]. Post-hoc pairwise comparisons also showed significant differences between every pair of interaction modes. These results collectively suggest that participants completed the first task fastest with hand-manipulated markers, slower with the two-handed interaction, and slowest when using the keypad controller.

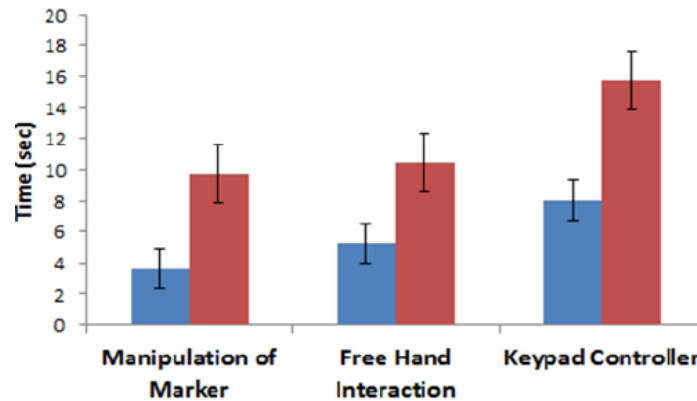


Fig. 7. Completion times in tasks 1 and 2.

In the second task, which required doing multiple 3D operations, ANOVA also revealed that the main effect of the interaction mode was significant on the completion time [ $F(1,17)=79.86$ ,  $p<.0001$ ]. However, pairwise mean comparison showed that there was no significant difference between the marker mode and the free hand mode. Similar to the first task, participants took more time to complete the task with the keypad mode than with the other two modes. However, the difference in completion time between the marker mode and free hand mode was marginal.

Among the three modes, the keypad mode took the highest completion time. This is due to the fact that, with the keypad, only one button can be activated at a single time. Thus, participants were forced to perform operations one by one. As revealed from the recordings, they had to complete one operation (e.g. rotation) before starting the following operation (e.g. scaling). In contrast, participants could generally perform more than one operation at a time when using the other two modes (e.g. rolling the marker as well as moving hand forward and backward). Besides, the speed of manipulating the object in the keypad mode is limited by the precision of the controller: the smaller the change resulting from a button press on the keypad, the more time it takes to complete the task.

## 5.2. Number of Deviations

Table 2 and Fig. 8 show the average number of deviations made across the two tasks. In the first task, which involved a single rotate action, ANOVA showed that the main effect of the interaction mode on the number of deviations approached statistical significance [ $F(1,17)=5.61$ ,  $p=0.0055$ ]. Results showed that participants deviated least when using the keypad controller, followed by the free-hand mode and then the marker mode. Post-hoc pairwise comparisons also showed significant differences between all possible pair of modes.

Looking at this result in conjunction with the average completion time, we found that participants spent less time using both the marker and the free hand modes but they made more deviations. In contrast, the keypad controller took more time but less number of deviations.

To explore the rationale behind this result, we traced video recordings in order to identify the deviations and their causes in each interaction mode. We identified the following issues that often hindered the user interaction:

**Loss of object tracking:** in the case of the marker mode, rotating the marker by 360 degree cannot be directly done using a single hand due to the limitation of the hand wrist. Thus, participants had to move the marker from one hand to another to do the full rotation. Due to this transition, participants sometimes failed to keep the

marker constantly visible, causing the vision system to temporarily lose the tracked marker, and hence to withhold the virtual object from the scene. This observation supports our hypothesis about the limitation of the marker mode for interaction in 3D space.

Table 2. Number of Deviations in Tasks 1 and 2

Mode	Task 1 Completion Time (in sec)		Task 2 Completion Time (in sec)	
	Mean	SD	Mean	SD
Manipulation of Marker	3.6	0.7	9.72	1.7
Free Hand Interaction	5.2	1.31	10.47	1.81
Keypad Controller	8.04	1.15	15.7	1.83

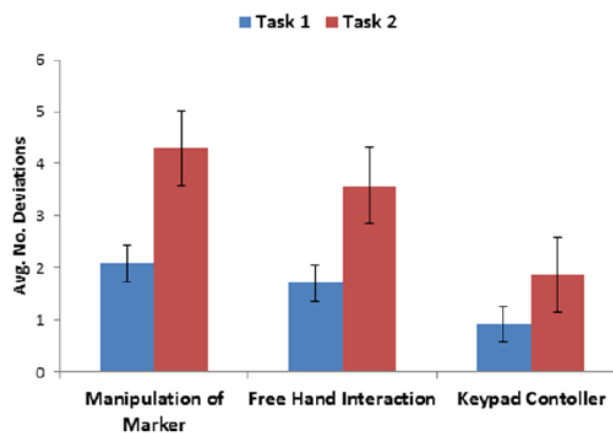


Fig. 8. Number of deviations in tasks 1 and 2.

Inaccurate object grasping and release: Looking at the user behavior when using the free hand mode, most erroneous actions happened due to the participants missing to open or close hand at the right time. As explained earlier, the user should close the hand fist to grab the virtual object before moving it. At the end of the task, the hand fist should be open to release the object being manipulated, and thus stop it from responding to hand movements. Most deviations in this mode resulted from the participants forgetting to close or open hand at the right time, causing the virtual object not to respond to their hand movements. Similarly, many participants also forgot to open hand when the virtual object reached its target transformation, causing the object to keep responding to unintentional hand movements.

On the other hand, controlling the 3D object using the keypad resulted in more robust interaction. This is because the visibility of the marker's pattern remains intact during interaction. It also does not involve actions for object grasping and release as in the free hand mode. In addition, the keypad mode resulted in precise positioning of the object. This precision can be explained by the fixed and rigorous changes that results from pressing the controller buttons. Hand moves, on the other hand, are often less accurate, as a small hand move can cause a big change in the position of controlled object.

### 5.3. Subjective Comments

Participants were asked about their preference of the interaction modes after they completed the tasks. In general, results showed that they were divided: 11 out of 18 participants preferred the manipulate marker mode while 7 participants preferred the free hand mode. No participant gave preference to the keypad mode. This result indicates that participants, in general, preferred interaction modes that offer more "natural"

interaction. Three participants who preferred the marker mode explained that it resembles the real interaction as if the real object was physically present in their hands.

Participants who preferred the hand mode agreed that they had no problem to mirror their hand actions to the target object without being distracted or overwhelmed. They also indicated that the open/close hand action gave them more control to halt and resume contact with the object during the task. This allowed them to fix the target object on a specific orientation/position by simply opening the hand fist to release the target object. Regarding the keypad mode, most participants found it more “artificial” and less natural than the other modes. Two participants indicated that it is more close to the traditional keyboard/mouse interaction.

#### **5.4. Implications of Results**

Looking at results from both tasks, we can highlight the following differences between the three interaction modes. In what follows, we present these differences and discuss how they inform the design of AR applications:

On average, the direct manipulation of markers achieved the best completion time and ease of use. However, there are many circumstances where this mode becomes impractical: The fact that the user directly manipulates the physical marker, which should be permanently trackable by the vision system, can increase the difficulty of interaction and restrict the degree of freedom. This was evident as participants were attempting to keep the marker constantly visible to the camera during interaction, a thing that can increase the cognitive load on them. In addition, some 3D manipulations can never be applied at all because it will cause the marker to be invisible. For example, it is impossible to position the house model in the previous experiment to be perfectly horizontal as this will cause the marker’s pattern to be invisible by the camera. The above result also suggests that the manipulated marker mode, which is a prevalent interaction paradigm in AR applications, can very often be less efficient than other modes, and may be inadequate for AR applications that demand flexible positioning or viewing of 3D object.

The free hand interaction and keypad controller modes are different in that they impose separation between the tracked object (e.g. the marker) and the interaction with the virtual object: participants hold the marker in one hand while manipulating the virtual object with the other hand. Therefore, the interaction was less error-nous because participants can focus on the interaction without worrying about the continuous visibility of the marker.

The keypad controller mode was not a preferable choice for participants. In general, participants preferred the other two modes of interaction as they were more intuitive and resembled the interaction with real objects. However, we cannot ignore the fact that using the keypad offered the highest precision and less-deviated manipulation. Therefore, designers should consider using the controller mode for AR applications in which the precision of interaction is a priority. Examples of these applications include medical/surgical training or flight simulations.

We also suggest that the separation between the object tracking and 3D interaction has the potential for further functionalities that can be useful for AR technology. Using both hands in AR environment (one hand for holding the marker and one hand for interacting with content) gives the opportunity to extend interaction in various ways. For example, a user can apply a hand gesture (e.g. swipe or point) with one hand while holding the virtual object with the other hand. The applied gesture can change the virtual object in different ways such as splitting, annotating, replacing or transforming it. Furthermore, free hand interaction offers the ability to perform two different independent transformations on the same object, such as changing its position with one hand while inspecting it from all sides by rotating with the other hand. In contrast, these capabilities are difficult to achieve in the marker mode where the whole interaction is restricted to a single hand that holds the marker and manipulates the object at the same time.

The above results suggest that there is no ideal mode of interaction for AR applications, and that what was thought to be common and easy to use in some situations can be impractical in other situations. Therefore, it is necessary for the designers of AR applications to consider the specific requirements of their applications in order to choose the best interaction technique.

## 6. Conclusion and Future Work

In this paper, we presented different techniques to interact with 3D content in AR environments. We also reported on an experiment that systematically examined the use of these techniques in practice. Guided by the results of our experiment, we discussed the strengths and limitations of these techniques as well as the circumstances in which each technique can be advantageous over the others.

While this work focuses primarily on object manipulation in 3D space, we will explore other interaction techniques in AR environments such as object annotation and 3D navigation. We also plan to focus on the design of AR applications for educational purposes, similar to the one shown in Fig. 1.b, and to examine the potential of the proposed interaction techniques for instruction in the classroom. We believe that the outcomes of this work will guide us to design effective interaction techniques to demonstrate and manipulate 3D learning material in AR environments.

## Acknowledgement

This work is funded by the Deanship of Scientific Research at the Islamic University of Gaza, Palestine.

## References

- [1] Buchmann, V., Violich, S., Billingham, M., & Cockburn, A. (2004). Fingertips: gesture based direct manipulation in augmented reality. *Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia* (pp. 212-221).
- [2] Schmalstieg, D., Fuhrmann, A., & Hesina, G. (2000). Bridging multiple user interface dimensions with augmented reality. *Proceedings of IEEE and ACM International Symposium on Augmented Reality* IEEE(pp. 20-29).
- [3] Carmigniani, J., Furht, B., Anisetti, M., Ceravolo, P., Damiani, E., & Ivkovic, M. (2011). Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, 51(1), 341-377.
- [4] Shen, Y., Ong, S. K., & Nee, A. Y. (2011). Vision-based hand interaction in augmented reality environment. *International Journal of Human Computer Interaction*, 27(6), 523-544.
- [5] Billingham, M. (2013). Hands and speech in Space: multimodal interaction with augmented reality interfaces. *Proceedings of the 15th ACM on International Conference on Multimodal Interaction* (pp. 379-380).
- [6] Volda, S., Podlaseck, M., Kjeldsen, R., & Pinhanez, C. (2005). A Study on the manipulation of 2d objects in a projector/camera-based augmented reality environment. *Proceedings of the Sigchi Conference on Human Factors in Computing Systems* (pp. 611-620).
- [7] Looser, J., Billingham, M., Grasset, R. L., & Cockburn, A. (2007). An evaluation of virtual lenses for object selection in augmented reality. *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia* (pp. 203-210).
- [8] Benko, H., & Feiner, S. (2007). Balloon selection: A multi-finger technique for accurate low-fatigue 3D selection. *Proceedings of the IEEE Symposium on 3D User Interface* (pp. 79-86).
- [9] Myojin, S., Kato, H., & Nishida, S. (2009). Evaluation of a cup-shaped interface in tabletop air environments. *Proceedings of the IEEE Symposium on 3D User Interfaces* (pp. 137-138).
- [10] Ha, T., & Woo, W. (2010). An empirical evaluation of virtual hand techniques for 3D object manipulation in a tangible augmented reality environment. *Proceedings of the IEEE Symposium on 3D User Interfaces* (pp. 91-98).
- [11] Kato, H., Tachibana, K., Tanabe, M., Nakajima, T., & Fukuda, Y. (2003). Magiccup: A tangible interface for virtual objects manipulation in table-top augmented reality. *Proceedings of the IEEE International Augmented Reality Toolkit Workshop* (pp. 75-76).
- [12] Mossel, A., Venditti, B., & Kaufmann, H. (2013). 3D touch and homer-s: intuitive manipulation techniques for one-handed handheld augmented reality. *Proceedings of the Virtual Reality International Conference: Laval Virtual* (pp. 12).

- [13] Wilson, A. D., & Benko, H. (2010). Combining multiple depth cameras and projectors for interactions on, above and between surfaces. *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology* (pp. 273-282).
- [14] Bai, H., Gao, L., Sana, J., & Billinghurst, M. (2013). Markerless 3D gesture-based interaction for handheld augmented reality interfaces. *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality* (pp. 1-6).
- [15] Hancock, M., Ten, C. T., & Carpendale, S. (2009). Sticky tools: Full 6dof force-based interaction for multi-touch tables. *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (pp. 133-140).
- [16] Ulinski, A. C., Zambaka, C. A., Wartell, Z., Goolkasian, P., & Hodges, L. F. (2007). Two handed selection techniques for volumetric data, 107-114.
- [17] Stellmach, S., Juttner, M., Nywelt, C., Schneider, J., & Dachselt, R. (2012). Investigating freehand pan and zoom, *Mensch and Computer* (pp. 303-312).
- [18] Malik, S. & Laszlo, J. (2004). Visual touchpad: A two-handed gestural input device. *Proceedings of the 6th International Conference on Multimodal Interfaces* (pp. 289-296).
- [19] Lee, M., Green, R., & Billinghurst, M. (2008). 3D natural hand interaction for AR applications. *Proceedings of the 23rd International Conference on Image and Vision Computing, New Zealand* (pp. 1-6).
- [20] Kurakin, A., Zhang, Z., & Liu, Z. (2012). A real time system for dynamic hand gesture recognition with a depth sensor. *Proceedings of the 20th IEEE European Signal Processing Conference (EUSIPCO)* (pp. 1975-1979).
- [21] Guomundsson, S. A., Sveinsson, J. R., Pardas, M., Aanaes, H., & Larsen, R. (2010). Model-based hand gesture tracking in TOF image sequences. *Articulated Motion and Deformable Objects*, 118-127.
- [22] Suryanarayan, P., Subramanian, A., & Mandalapu, D. (2010). Dynamic hand pose recognition using depth data. *Proceedings of 20th IEEE International Conference on Pattern Recognition (ICPR)* (pp. 3105-3108).
- [23] Van, D. B. M., & Van, G. L. (2011). Combining RGB and TOF cameras for real-time 3D hand gesture interaction. *Proceedings of IEEE Workshop on Applications of Computer Vision (WACV)* (pp. 66-72).
- [24] Chaudhary, A., Raheja, J., Das, K., & Raheja, S. (2011). A survey on hand gesture recognition in context of soft computing, *Advanced Computing*, 46-55.
- [25] Ren, Z., Yuan, J., Meng, J., & Zhang, Z. (2013). Robust part-based hand gesture recognition using kinect sensor. *Multimedia*, 15(5), 1110-1120.
- [26] Martinet, A., Casiez, G. R., & Grisoni, L. (2010). The effect of DOF separation in 3D manipulation tasks with multi-touch displays. *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology* (pp. 111-118).



**Iyad M. AlAgha** received his MSc and PhD in computer science from the University of Durham, the UK. He worked as a research associate in the Center of Technology Enhanced Learning at the University of Durham, investigating the use of Multi-touch devices for learning and teaching. He is currently working as an assistant professor at the Faculty of Information technology at the Islamic University of Gaza, Palestine. His research interests are semantic web technology, adaptive hypermedia, human-computer interaction and technology enhanced learning.



**Raed S. Rasheed** is an instructor of multimedia at the Islamic University Gaza, Palestine. He obtained his M.Sc. from IUG-GAZA, his B.Sc. in computer science from Applied Science University, Amman, Jordan. His recent research includes projects on the multimedia, 3D Web and computer vision.