

# Virtual Object Manipulation using a Mobile Phone

Anders Henrysson<sup>1</sup>, Mark Billinghurst<sup>2</sup> and Mark Ollila<sup>1</sup>

<sup>1</sup>NVIS, Linköping University, Sweden {andhe, marol}@itn.liu.se

<sup>2</sup>HIT Lab NZ, University of Canterbury, New Zealand mark.billinghurst@hitlabnz.org

## Abstract

Augmented Reality (AR) on mobile phones has reached a level of maturity where it can be used as a tool for 3D object manipulation. In this paper we look at user interface issues where an AR enabled mobile phone acts as an interaction device. We discuss how traditional 3D manipulation techniques apply to this new platform. The high tangibility of the device and its button interface makes it interesting to compare manipulation techniques. We describe AR manipulation techniques we have implemented on a mobile phone and present a small pilot study evaluating these methods.

**Key words:** Augmented Reality, Mobile Phone, Manipulation

## 1. Introduction

Augmented Reality (AR) is a technology that allows a user to see virtual imagery overlaid and registered with the real world. Traditionally the AR content was viewed through a head mounted display (HMD). Wearing an HMD leaves the users hands free to interact with the virtual content, either directly or using an input device such as a mouse or digital glove.

In recent years AR applications have migrated to other platforms, including Tablet PCs [27], PDAs [28] and mobile phones [18]. The mobile phone is an ideal platform for augmented reality (AR). The current generation of phones has full color displays, integrated cameras, fast processors and even dedicated 3D graphics chips. Henrysson [10] and Moehring [18] have shown how mobile phones can be used for simple single user AR applications. In their work they create custom computer vision libraries that allows developers to build video see through AR applications that run on a mobile phone.

For handheld and mobile phone based AR the user looks through the screen of the device to view the AR scene and needs at least one hand to hold the device. The user interface for these applications is very different than those for HMD based AR applications. Thus there is a need to conduct research on interaction techniques for handheld AR displays, and to produce formal user studies to evaluate these techniques. This is important because the widespread adoption of mobile phones means that this platform could be one of the dominant platforms for AR applications in the near future.

In this paper we present one of the first mobile phone AR applications in which the user can manipulate the virtual objects being shown. We explore several possible manipulation techniques and conduct a user study to identify which of these techniques is the most usable.

The work that we present here is not specifically for mobile phone AR applications. It is also useful for other AR applications that run in a handheld form factor, and for general 3D graphics applications on a phone or handheld devices. The need for effective manipulation techniques is common across a wide variety of application areas.

In the next section we review related work in the area of virtual object manipulation, especially on a handheld platform. We then describe several approaches that we have implemented, and a user study to test between approaches. Finally we provide some design recommendations and directions for future research.

## 2. Related Work

The need for methods to select and manipulate virtual objects is basic to many types of graphics applications. Bowman et. al. [2] identify three basic object manipulation tasks: selection, positioning, and rotation. They also provide a taxonomy for classifying manipulation techniques.

On a desktop user interface, selection is commonly performed by mouse input and once selected, objects are also manipulated with mouse input. For three-dimensional manipulation, the challenge is using 2D mouse input to control 3D virtual object translation and orientation. In a conventional interface slider, controls can be used with each slider controlling one degree of positional or rotational freedom [4]. Alternatively, 3D widgets can be used which place virtual handles on objects which can then be translated or rotated by clicking and dragging on the handles [12].

The most challenging interaction on the desktop is setting the orientation of a virtual object. Virtual Sphere [4] and Virtual Trackball [13] techniques can be used to map 2D mouse motions onto the surface of a virtual sphere that surrounds the object being rotated.

Three-dimensional object manipulation is more natural in Virtual Reality (VR) environments. In this case users can reach out grab objects with their hands [23] or

employ a variety of ray-casting [3] and interaction at a distance techniques. Some of these techniques can also be employed in AR environments, particularly in HMD based systems. In AR interfaces there is also a close relationship between the virtual imagery and real world and so some metaphors, such as Tangible AR interaction methods [14] use real object manipulation to interact with virtual content.

Researchers have begun conducting formal user studies with HMD based AR systems, but there has been far less research in handheld interfaces. There are several examples of handheld AR interfaces where the user interacts with the content rather than just viewing it. For example, in Rekimoto's Transvision interface [25] two users sit across the table and see shared AR content shown on handheld LCDs panels. They can select objects by ray casting and once selected objects are fixed related to the LCD and can be moved. The ARPAD interface [17] is similar, but it adds a handheld controller to the LCD panel. Selection is performed by positioning virtual cross hairs over the object and hitting a controller button. Once selected the object is fixed in space relative to the LCD panel and so can be moved by moving the panel. The object can also be rotated using a trackball input device, thus ARPAD decouples translation and rotation. More recently the Invisible train [28] uses a PDA to view AR content and users can select virtual models directly by clicking on the model with a stylus. Similar stylus based selection has been implemented in AR interfaces that run on tablet PCs [27].

Despite these examples of selection and manipulation techniques on handheld displays there have been no formal usability studies to see what are the best methods to use on a handheld device. As we show in the next section, handheld devices are different enough from desktop and immersive VR interfaces that the research results gathered from user studies in non-handheld environments may not be applicable.

On the mobile phone there are very few examples of AR applications and none of them support more than simple object selection and manipulation. For example, Mohering [18] has developed a simple AR viewing application, however there is no selection or manipulation of virtual objects possible. Our early work with AR tennis [10] is a mobile phone based collaborative AR application that allows users to hit a virtual ball over a net, but this also does not support more complex interaction.

There are also simpler examples of mobile phone games that feature graphics overlaid on video of the real world, although without 3D registration of the graphics as is normal in an AR application. The Siemen's Mosquito game [19] shows virtual mosquitos that can be killed with a simple "point and shoot" metaphor. The virtual soccer game of KickReal [15] allows people to see a virtual ball superimposed over video of the real world

and kick it with their feet, but again there is no 3D object manipulation. Like AR Tennis, the CamBall application [8] allows users to hit balls at each other, although with limited 3D tracking. In none of these cases has there been a formal evaluation of their usability.

There have been some efforts to implement non-AR 3D graphics applications on mobile phones. There are a range of games that provide joystick type control of vehicles and objects in 3D environments. Most of the control techniques are adopted from console interaction metaphors. Larsen et. al. [16] describe one of the first 3D applications for the mobile with more complex object manipulation. This is a brick-modeling program where the user selects and moves virtual bricks using the arrow keys on the phone. Once again, there is no evaluation of the usability of the technique.

In contrast, in our work we have developed an AR application that runs on the mobile phone and supports selection, translation and rotation of 3D virtual objects using a variety of techniques adopted from desktop and AR user interfaces. We also evaluate these interaction techniques using a formal user study.

### 3. Interaction Methods

In order to explore methods for manipulation in AR applications on a mobile phone we need to consider the appropriate interaction metaphor. There are a number of important differences between using a mobile phone AR interface and a traditional desktop interface, including:

- Limited input options (no mouse/keyboard)
- Limited screen resolution
- Little graphics support
- Reduced processing power

Similarly, compared to a traditional HMD based AR system, in an AR application on a phone the display is handheld rather than headworn, and the display and input device are connected. Finally, compared to a PDA the mobile phone is operated using a one-handed button interface in contrast to a two-hand stylus interaction.

These differences mean that interface metaphors developed for Desktop and HMD based systems may not be appropriate for handheld phone based systems. For example, applications developed with a Tangible AR metaphor [14] often assume that the user has both hands free to manipulate physical input devices which will not be the case with mobile phones.

We need to develop input techniques that can be used one handed and only rely on a joystick and keypad input. Since the phone is handheld we can use the motion of the phone itself to interact with the virtual object. For example, as in ARPAD, we can fix the virtual object relative to the phone and then position objects by moving the phone relative to the real world. Two handed interaction techniques [11] can also be explored; one hand holding the phone and the second a real object on which AR graphics are overlaid. This approach assumes

that phone is like a handheld lens giving a small view into the AR scene. In this case the user may be more likely move the phone-display than change their viewpoint relative to the phone. The small form factor of the phone lets us explore more object-based interaction techniques based around motion of the phone itself.

Given these requirements there are several possible manipulation methods that could be tried. Table 1 shows the techniques we have implemented.

Positioning	Rotation
A/ Tangible 1: The object is fixed relative to the phone and moves when the user moves the phone. When released the object position is set to the final translated position while its orientation is reset to its original orientation.	A/ ArcBall [4]: When the phone moves the relative motion of the phone is used as input into the ArcBall technique to rotate the currently selected object.
B/ Keypad/Joypad: The selected object is continuously translated in the X, Y or Z directions depending on the buttons currently held down.	B/ Keypad/Joypad: The object rotates about its own axis according to joypad and keypad input. Left and right joypad input causes rotation left and right about the vertical axis etc.
C/ Tangible 2: The same as tangible 1, but the user can use bimanual input, moving both the phone and the object that the phone is tracked relative to.	C/ Tangible 1: The object is fixed relative to the phone and moves when the user moves the phone. When released the object orientation is set to the final phone orientation and position reset to its original position.
	D/ Tangible 2: The same as tangible 1, but the user can use bimanual input, moving both the phone and the object that the phone is being tracked relative to.

Table 1: Methods for Translation and Rotation

In the Tangible Input cases the translation and rotation techniques are applied to objects that are selected by positioning virtual cross hairs over them and clicking and holding down the joypad controller. Objects are deselected by releasing the joypad controller. For the Keypad and ArcBall methods the user just has to click on the keypad to start the motion.

In our initial study we wanted to consider positioning and rotation separately. So all of the techniques except

for the Tangible Input cases separate positioning and rotation. In the Tangible Input cases the virtual model is fixed in space relative to the phone and so can be positioned and translated at the same time. However once the person de-selects the model the rotation or position of the model is reset back to its original depending on if we are conducting a positioning or rotation user study.

In the keypad/joypad method the objects continuously rotate or translate a fixed amount for each fraction of a second while the buttons are pressed. In contrast when the virtual object is fixed relative to the phone (Tangible Input), the user can move the object as fast as they can move the phone. So the user should be able to translate or rotate the objects faster using tangible input techniques than with keypad input.

### 3. Platform

In order to implement an AR application on the phone and these various manipulation techniques it was necessary to develop a custom low level computer vision library for Symbian based mobile phones. This is described in complete detail in an earlier paper [9]. In this section we provide a brief overview of the computer vision work we have done and then focus more on the new code we have developed for the manipulation techniques.

Our mobile phone AR platform is based on our earlier custom port of the ARToolKit computer vision tracking library [1] to the Symbian operating system [10]. ARToolKit can be used to calculate the 3D pose of a camera relative to a single square tracking marker. Although designed for the PC platform, our Symbian port of ARToolKit is able to run on current mobile phones at 6 – 7 frames per second. Creating the ARToolKit port involved creating an optimized fixed point library for image processing on the phone. Figure 1 shows our AR application running on the mobile phone. When the square marker is in view a virtual image appears overlaid on it in the camera view.



Figure 1: Our AR Application on the Mobile Phone

In addition to the tracking software we needed graphic application code. The OpenGL library is a powerful

graphics API that was the natural starting point for the development of a graphics API for mobile devices. Our graphics application was developed using OpenGL ES, which is a reduced subset of OpenGL 1.3, suitable for low-power, embedded devices. The phone we were developing for, the Nokia 6630, ships with a software implementation of OpenGL ES [21].

All of the manipulation techniques mentioned in section 3 were coded in OpenGL ES and can run on the Nokia series 60 phones with an integrated camera. For our experiment we used two Nokia 6630 phones. The Nokia 6630 has a 220Mhz processor and an integrated 1.3 megapixel camera. The screen size is 178 x 208 pixels. On these phones the manipulation applications typically ran at 6-7 frames per second with a video capture resolution of 160x120 pixels.

The object being manipulated is a virtual box with the dimension of 80x64x32 units. We implemented two different techniques for translating the object. In the first case the object remains at a fixed transformation relative to the camera while selected. Selection is performed by pressing the joypad. Each object has a unique alpha value and the selection is accomplished by sampling the alpha value of the central pixel, indicated by a crosshair. When the object is released a new transform is calculated and its rotation component set to the unit matrix.

In the second case the box is translated by the pressing of keys – two for each dimension. To translate the object in the x-z plane we use the four directions of the joypad and complement it with the 2 and 5 keys for translation along the y-axis. The translation speed is 4 units/frame yielding a speed of about 30 units per second.

At each update an error vector is calculated by taking the current position minus the goal position. The block is regarded to have been placed correctly if the length of the error vector less than 8 units.

For the rotation we added a second block since a single block is rotation invariant. We implemented the two most important rotation techniques found in 3D applications – the ArcBall and rotation around the object axis– along with the isomorphic case where the object is fixed relative to the phone.

The ArcBall allows the user to perform large 3DOF rotations using small movements. Our ArcBall was implemented using the code provided by NeHe Productions[20]. In its original use, the mouse pointer is used to manipulating an invisible ball that contains the object to be rotated. The resulting rotation depends on where on the ball the user clicked and in which direction the pointer was dragged. In our case the center of the bottom block is projected into screen coordinated and the crosshair act as a mouse pointer around which the screen is centered. Our implementation has some limitations. In particular the ArcBall is assumed to be manipulated from roughly the same viewpoint at all

times since its internal rotation is not updated when the camera moves unless clicked. There is also a jumping artifact at the very first clicking.

For rotation using the keypad we use the joypad to rotate around the x and z-axis, while the 2 and 5 buttons rotate the object around the y-axis. The speed of rotation is 4 degrees per update i.e. around 30 degrees per second.

The Tangible Input condition attaches the virtual object to the camera just like in the translation case. However, with the difference that in this case when the object is released it is the translation part that is set to zero.

The error metric for the rotation conditions is the sum of the absolute values of the difference between the current rotation matrix and the goal matrix. The rotation is regarded to be correct if the sum is less than 0.4.

#### 4. User Study

In order to test the usability of the manipulation techniques described earlier we conducted a study in which users tried to position and orient blocks. The subject sits at a table, which has a piece of paper with a number of ARToolKit tracking markers printed on it. When the user looks through the phone display at the tracking marker they will see a virtual ground plane with a virtual block on it and a wireframe image of the block.

The study was done in two parts. In the first we tested the following three positioning conditions:

- A: Object fixed to the phone (one handed)
- B: Button and keypad input
- C: Object fixed to the phone (bimanual)

In each case the goal was to select and move the block until it was inside the target wireframe block (see Figure 2). In the bimanual case the user is able to move both the phone and the piece of paper that the virtual model appears attached to. In all other cases the subject wasn't allowed to move the tracking marker, although they could stand and walk around the table.

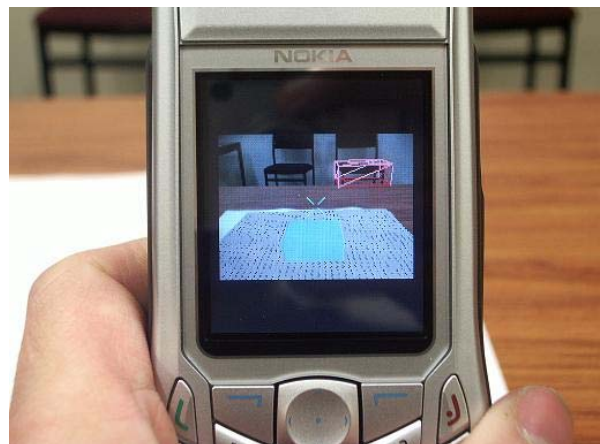


Fig. 2 A virtual block and translation target

In the second part of the experiment we tested the following rotation techniques:

- A: ArcBall
- D: Keypad input for rotation about the object axis
- B: Object fixed to the phone (one handed)
- C: Object fixed to the phone (bimanual)

For each condition the virtual block was shown inside a wireframe copy and the goal was the rotate the block until it matched the orientation of the wireframe copy (see Figure 3). In the bimanual case the user was able to rotate the tracking paper in one hand while moving the phone in the other, while in the other conditions the user wasn't able to move the tracking marker.

In both the translation and rotation case the user was able to practice in each condition before trying the experimental task. Once they felt comfortable with the technique they also performed the task three times for each condition with virtual blocks at different positions and orientations. The order of trying the conditions was counterbalanced to remove order effects.

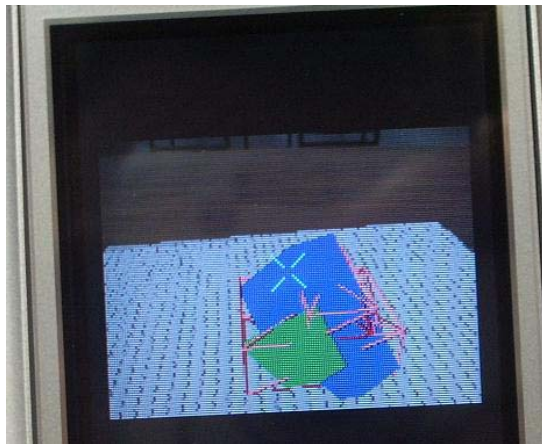


Fig. 3 A virtual block and rotation target

When the block was positioned or rotated correctly inside the target wire-frame it changed color to yellow showing the subject that the trial was over. This was determined by measuring the error in position or orientation and stopping the trial once this error value dropped below a certain threshold.

For each trial we measured the amount of time it took the user to complete the trial and also continuously logged the position or rotation of the block relative to the target. After three trials in one condition we asked the subject to subjectively rate his or her performance and how easy was it for them to use the manipulation technique. Finally after all the positioning or orientation conditions were completed we asked the users to rank all them in order of ease of use and asked them some interview questions.

## 4.1 Results

We recruited a total of 9 subjects for the user studies, 7 male and 2 female, aged between 22 and 32. None of the subjects had experience with 3D object manipulation on mobile phones but all of them had used mobile phones before and some of them had played games on their mobile phone.

### Positioning

There was a significant difference in the time it took users to position objects depending on the positioning technique they used. Figure 4 shows the average time it took the users to position the virtual block in the wireframe target.

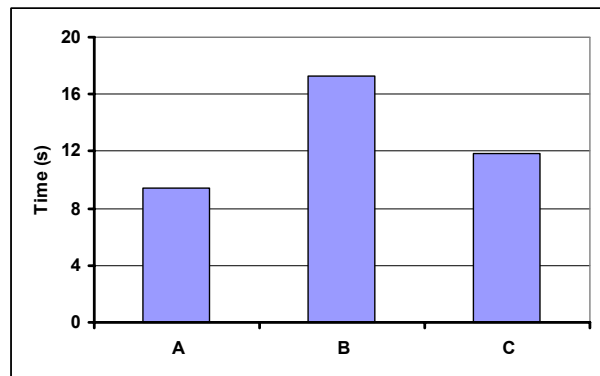


Fig. 4 Average Positioning Times

As can be see conditions A and C take less time that the keypad condition (condition B). Using a one factor ANOVA ( $F(2,24) = 3.65, P < 0.05$ ) we find a significant difference in task completion times. The users also subjectively preferred condition A. Subjects were asked to answer the following questions:

- Q1: How easy was it for you to position the object?
- Q2: How accurately did you think you placed the block?
- Q3: How quickly did you think you placed the block?
- Q4: How enjoyable was the experience?

Using a scale of 1 to 7 where 1= not very easy, 7 = very easy, etc. Table 2 shows the average results.

	A	B	C
<b>Q1</b>	5.8	5.0	5.8
<b>Q2</b>	5.4	6.0	5.1
<b>Q3</b>	5.9	4.4	5.1
<b>Q4</b>	5.7	4.6	5.3

Table 2. Subjective Results

As can be seen, the users thought that when the object was fixed to the phone (conditions A and C) it was easier to position the object correctly (Q1) but it they could position the model more accurately (Q2) with the keypad input. A one factor ANOVA finds a near significant difference in the results for Q1 ( $F(2,24) = 2.88, P = 0.076$ ) and Q2 ( $F(2,24) = 3.32, P = 0.053$ ).



There is a significant difference in the other conditions. The users thought they could place the objects more quickly when they were attached to the phone (Q3) and the tangible interfaces were more enjoyable (Q4). A one factor ANOVA finds a significant difference in the results for Q3 ( $F(2,24) = 5.13, P < 0.05$ ) and Q4 ( $F(2,24) = 3.47, P < 0.05$ ).

The users were asked to rank the conditions in order of ease of use (1 = easiest, 3 = most difficult). Table 3 shows the average ranking. Condition A and C are the best ranked conditions. A one factor ANOVA gives a significant difference between conditions ( $F(2,24) = 5.36, P < 0.05$ ).

	A	B	C
<b>Rank</b>	1.44	2.56	2.0

Table 3. Average Positioning Rank (1= highest)

### Orientation

There was also a significant difference in the time it took users to orient objects depending on the technique they used. Figure 5 shows the average time it took the users to rotate the virtual block to match the wireframe target.

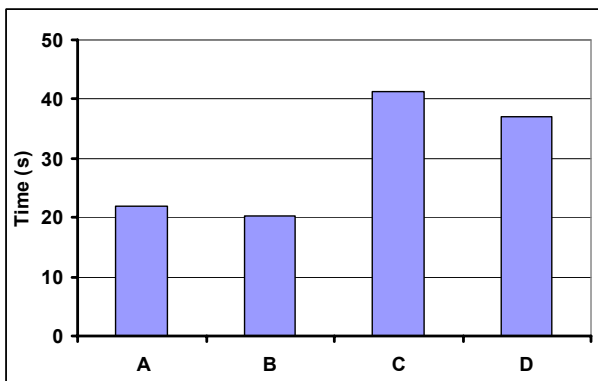


Fig. 5 Average Rotation Times

As can be seen, conditions A (ArcBall) and B (keypad input) are on average twice as fast as the Tangible Input rotation conditions (C and D). A one-factor ANOVA finds a significant difference between these times ( $F(3,32) = 4.60, P < 0.01$ ).

Subjects were also asked to answer the same survey questions as in the translation task, except Q1 was changed to:

Q1: How easy was it for you to rotate the virtual object?

Table 4 shows the average results.

	A	B	C	D
<b>Q1</b>	4.7	5.4	4.6	5.1
<b>Q2</b>	4.8	5.2	4.8	5.0
<b>Q3</b>	5.0	4.8	4.1	4.6
<b>Q4</b>	5.1	4.9	4.9	5.6

Table 4. Subjective Rotation Results

There were no significant differences between these survey responses. The subjects thought that the conditions were equally easy to use and enjoyable.

The users were asked to rank the conditions in order of ease of use (1 = easiest, 5 = most difficult). Table 5 shows the average ranking. There is no significant difference between the results, a one factor ANOVA finding ( $F(3,32) = 0.82, P = 0.49$ ).

	A	B	C	D
<b>Rank</b>	3.0	2.3	2.4	2.2

Table 5. Ranking of Orientation Results

### Observations

Although the interfaces were designed to be used one handed it was interesting to observe how they were actually used. When subjects used the keypad or ArcBall conditions they would typically hold or steady the phone with their non-dominant hand and push the keys with their dominant hand (see Figure 6). This was to provide support for when they pressed the keys. They also typically remained seated since they didn't need to move the phone much to translate or rotate the model.

In contrast, with the conditions where the virtual object appeared attached to the phone the user would hold the phone with one hand (clicking the joystick) and stand up and move around the tracking pattern to get a better view. This was especially true in the rotation case where they often had to rotate the phone to extreme angle to get the object rotation they wanted. In the bi-manual case, users would typically sit and use their non-dominant hand to rotate the target tracking pattern, while moving the phone in the other hand (See Figure 7). About half of the subjects in the translation bi-manual case chose not to move the paper with their free hand, but almost all did in the rotation case.



Figure 6 Holding the phone with both hands

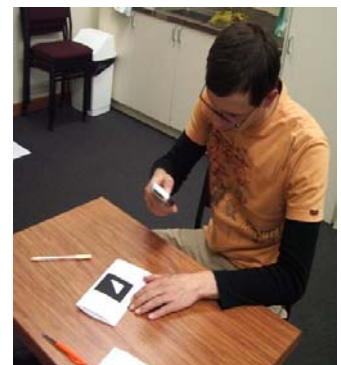


Fig. 7 A subject using bimanual input

### User Feedback

In addition to survey responses many users gave additional comments about the experience. Several commented that when the virtual object was attached to

the phone they felt like they were holding it, compared to the case where the keypad was used and they felt that they were looking at a screen. One user said “when the object was attached to the phone, the phone felt more like a tool.” They felt like they were more in control and they could use their spatial abilities when manipulating the virtual object. In contrast those that preferred the keypad liked how it could be used for precise movements and also how you didn’t need to physically move themselves to rotate the object about its axis.

Some users also commented on a lack of visual feedback about the rotation axis. The block changed color when it was released inside the target but subjects thought it would have been good to change before it was released. They also felt visual cues showing the axis of rotation would be helpful, especially in the case of the ArcBall.

Several of the users also had trouble with the computer vision tracking failing. When part of the marker was covered up then the virtual objects would disappear. However this was very temporary and most subjects adapted their behavior to prevent this.

Those subjects that did use two handed input said that they felt they had more control because they could make gross movements with the camera and then fine tune the block position with small movements of the marker.

### 4.3 Discussion

In this pilot study we have explored a variety of methods for rotation and translation of virtual objects. The results show that using a tangible interface metaphor provides a fast way to position AR objects in a mobile phone interface because the user just has to move the real phone where the block is to go. The subjects also felt that it was more enjoyable.

However, there seems to be little advantage in using our implementation of a tangible interface metaphor for virtual object rotation. When the virtual object is fixed to the phone then the user often has to move themselves and the phone to rotate the object to the orientation they want, which takes time. Even when the person can use a second hand to rotate the tracking marker, this is still more time consuming than using the ArcBall or keypad input. One of the main advantages of the keypad is that just rotates the object around one axis at a time and so makes it easy for the user understand what the rotation axis is and how to undo any mistakes.

There is also a compromise between speed and accuracy that may also affect performance. Tangible input techniques may be fast, but because they provide full six degree of freedom input, they may not be the best methods for precise input. This was shown in the rotation study where more precise input was needed to correctly align the models.

### 5. Conclusion

In this paper we have reviewed some of the issues that must be considered when designing AR interfaces for

mobile phones and have presented a pilot study evaluating different types of virtual object manipulation techniques.

Our results suggest that virtual object positioning based on physical phone motion could be a valuable technique but rotation may be better performed through keypad input about constrained axes.

However this is just a pilot study. In the future, we will need to conduct more rigorous studies with different tasks. In particular we need to explore manipulation when object position is not decoupled from rotation, such as 3D path following. In this case the need for rapid six degree of freedom input may mean that a tangible interface metaphor for both position and rotation may have a significant advantage over other techniques.

### 6. Acknowledgement

The first author is funded by the Department of Science and Technology at the University of Linköping as well receiving supervisory support from the Swedish National Graduate School in Computer Science.

### References

- [1] ARToolKit site: [www.hitl.washington.edu/artoolkit/](http://www.hitl.washington.edu/artoolkit/).
- [2] Bowman, D., Kruijff, E., LaVoila, J., Poupyrev, I. 3D User Interfaces: Theory and Practice. Addison-Wesley 2005.
- [3] Bowman, D., Hodges, L. (1997) An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. Proceedings of the 1997 ACM Symposium on Interactive 3D Graphics (I3D '97), ACM Press, pp. 35-38.
- [4] Chen, M., Mountford, S., Sellen, A. (1988) A Study in Interactive 3-D Rotation Using 2-D Control Devices. Computer Graphics 22(4): 121-129.
- [5] Cutting D. J. C. D., Assad M. and Hudson A. *AR phone: Accessible augmented reality in the intelligent environment*. In OZCHI2003, Brisbane, 2003.
- [6] Feiner T. H. S., MacIntyre B. and Webster T. *A touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment*. In Proc. ISWC '97 (First IEEE Int. Symp. On Wearable Computers), Cambridge, MA, 1997.
- [7] Geiger C, Kleinjohann B, Reimann C, Stichling D. *Mobile AR4ALL*, ISAR 2001, The Second IEEE and ACM International Symposium on Augmented Reality, New York, (2001).
- [8] Hakkarainen, M., Woodward, C., “SymBall - Camera driven table tennis for mobile phones”, submitted to *ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE 2005)*, Valencia, Spain, 15-17 June, 2005 .
- [9] Henrysson, A., Billinghurst, M., Ollila, M. Face to Face Collaborative AR on Mobile Phones. In Proceedings of ISMAR 2005, 2005 (To Appear) Vienna, Austria.
- [10] Henrysson A. and Ollila M. *UMAR - Ubiquitous Mobile Augmented Reality* In Proc. Third International Conference on Mobile and Ubiquitous Multimedia

- (MUM2004) pp 41-45. College Park, Maryland, U.S.A. October 27 - 29, 2004.
- [11] Hinckley, K., Pausch, R., Proffitt, D., Patten, J., Kassell, N., Cooperative Bimanual Action, ACM CHI'97 Conference on Human Factors in Computing Systems, pp. 27-34.
- [12] Houde, S. (1992) Iterative Design of an Interface for Easy 3-D Direct Manipulation. Proceedings of the 1992 ACM Conference on Human Factors in Computing Systems (CHI '92), ACM Press, pp. 135-142.
- [13] Hultquits, J. (1990) A Virtual Trackball. Graphics Gems I, Academic Press, pp. 462-463.
- [14] Kato H., Billinghurst M., Poupyrev I., Tetsutani N. and Tachibana K. Tangible Augmented Reality for Human Computer Interaction. In *Proc. of Nicograph 2001*. (Nagoya, Japan, 2001).
- [15] KickReal website: <http://www.kickreal.de/>
- [16] Larsen, B., Barentzen, J., Christensen, N. Using cellular phones to interact with virtual environments ACM Siggraph 2002, Conference Abstracts and Applications, pp. 274.
- [17] Mogilev, D., Kiyokawa, K., Billinghurst, M., Pair, J. *AR Pad: An Interface for Face-to-face AR Collaboration*, Proc. of the ACM Conference on Human Factors in Computing Systems 2002 (CHI '02), Minneapolis, pp.654-655, 2002.
- [18] Moehring, M., Lessig, C. and Bimber, O. *Video See-Through AR on Consumer Cell Phones*. In Proc. of International Symposium on Augmented and Mixed Reality (ISMAR'04), pp. 252-253, 2004.
- [19] Mosquito Hunt. [http://w4.siemens.de/en2/html/press/newsdesk\\_archive/2003/foe03111.html](http://w4.siemens.de/en2/html/press/newsdesk_archive/2003/foe03111.html)
- [20] NeHe Productions: <http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=48>
- [21] OpenGL ES Site: <http://www.khronos.org/opengles/>
- [22] Piekarski, W. and Thomas, B. H. Tinmith-Hand: Unified User Interface Technology for Mobile Outdoor Augmented Reality and Indoor Virtual Reality. In IEEE Virtual Reality Conference, Orlando, FL, Mar 2002.
- [23] Poupyrev, I., Billinghurst, M., Weghorst, S., Ichikawa, T. (1996) The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR. Proceedings of the 1996 ACM Symposium on User Interface Software and Technology (UIST '96), ACM Press, pp. 79-80.
- [24] Reitmayr, G., Schmalstieg, D. Mobile Collaborative Augmented Reality In Proc. ISAR 2001, New York, USA, Oct. 2001.
- [25] Rekimoto J., *TransVision: A Hand-held Augmented Reality System for Collaborative Design*. Virtual Systems and Multi-Media (VSMM)'96, 1996.
- [26] Thomas, B., Close, B., Donoghue, J., Squires, J., De Bondi, P., Morris, M., and Piekarski, W. *ARQuake: An Outdoor/Indoor Augmented Reality First Person Application*. Proc. 4th Int'l Symposium on Wearable Computers, pp 139-146, Atlanta, Ga, USA, Oct 2000.
- [27] Träskbäck M., Haller M. Mixed reality training application for an oil refinery: user requirements. In ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry, VRCAI 2004, pp. 324- 327, Singapore.
- [28] Wagner, D., Schmalstieg, D.: *First steps towards handheld augmented reality*. Proc. of the 7th International Symposium on Wearable Computers (ISWC2003), White Plains, NY, USA, IEEE Computer Society (2003) 127–137.