Freeze View Touch and Finger Gesture based Interaction Methods for Handheld Augmented Reality Interfaces

Huidong Bai
The Human Interface Technology Laboratory New Zealand
University of Canterbury
Christchurch, New Zealand
huidong.bai@pg.canterbury.ac.nz

Gun A. Lee
The Human Interface Technology Laboratory New Zealand
University of Canterbury
Christchurch, New Zealand
gun.lee@hitlabnz.org

Mark Billinghurst
The Human Interface Technology Laboratory New Zealand
University of Canterbury
Christchurch, New Zealand
mark.billinghurst@hitlabnz.org

ABSTRACT

Interaction techniques for handheld mobile Augmented Reality (AR) often focus on device-centric methods based around touch input. However, users may not be able to easily interact with virtual objects in mobile AR scenes if they are holding the handheld device with one hand and touching the screen with the other, while at the same time trying to maintain visual tracking of an AR marker. In this paper we explore novel interaction methods for handheld mobile AR that overcomes this problem. We investigate two different approaches; (1) freeze view touch and (2) finger gesture based interaction. We describe how each method is implemented and present findings from a user experiment comparing virtual object manipulation with these techniques to more traditional touch methods.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Artificial, augmented, and virtual realities*

General Terms

Experimentation

Keywords

handheld mobile augmented reality, freeze view touch, finger gesture based interaction

1. INTRODUCTION

In recent years advances in smart phone technology has made handheld Augmented Reality (AR) very popular. Using a handheld AR interface on a mobile phone, a person can see virtual objects superimposed on live video of the real world using visual tracking or location-based sensing and graphics rendering. A variety of mobile AR interaction techniques can also be used to manipulate these virtual objects such as performing translation, rotation and scaling to alter the location, pose and size of virtual objects in space.

In general, handheld AR interaction techniques can be classified into two main categories; (1) tangible interaction methods and (2) intangible interaction methods. Tangible interaction methods are those that capture the user's commands via physical interaction with the device itself. For example, people scroll their fingers on the touch screen to translate a 3D cube, or rotate their mobile phone with the built-in gyroscope to spin a virtual cylinder. In contrast, intangible interaction methods interpret the user's actions separated from the device (e.g., midair gesture, gaze or speech) and map them onto input parameters to control virtual content. For instance, a user puts his/her hand in front of the mobile phone's camera and drags to select a virtual cube, or pinches to scale it.

Using a handheld mobile AR system is very different from using a more traditional head mounted display (HMD) based AR system because the handheld device combines both display and input technologies. Since the virtual content is overlaid on a view of the 3D physical world, 3D user interface techniques should be used to interact with the content, while conventional 2D interaction techniques may not be as useful. Unfortunately, many handheld AR interaction methods are limited to the pure 2D screen touch input, which has many problems. For example, pressing the touch screen while pointing the device at an AR tracking image can result in camera motion that has a serious negative effect on the AR tracking accuracy, and overall user experience [8].

In our research we are exploring interaction methods for handheld mobile AR that does not rely on the user touching the screen. In this paper we report on two handheld AR interaction methods; (1) freeze view touch and (2) finger gesture based interaction.

The freeze view touch based interaction method was developed to avoid the problem of the handheld device shaking due to a user touching its screen while trying to maintain visual tracking. This method simulates a fixed view position by freezing the current camera frame and allows the user to move the device without updating the visual tracking. The user can use the touch screen to interact with virtual content while the camera view is frozen. After a successful completion of the selected task, the user can unfreeze the view to

IVCNZ '12, November 26 - 28 2012, Dunedin, New Zealand
Copyright 2012 ACM 978-1-4503-1473-2/12/11 ...$15.00.

restart the tracking thread.

The finger gesture interaction helps users to interact with handheld mobile AR applications using their bare hands. The camera on the back of the handheld device can track the user's fingertips and allows them to reach out and directly interact with virtual content in the AR scene. The system will segment the hand region and identify the fingertip, which will be treated as the key input point like the cursor on the desktop computer. This is an intangible approach and so avoids the need for the user to interact with the touch screen.

Both of these offer interesting alternatives to traditional touch-screen based interaction methods. In this paper, we start by describing the context of our research and discussing recent related work in Section2. Sections 3 and 4 introduce the concept and implementation of our freeze view touch and finger gesture based interaction methods respectively. Section 5 presents a user study evaluating both interaction techniques as well as a traditional free view touch method. We compare them from different aspects (operation time, accuracy, and user experience) in order to investigate their usefulness for handheld mobile AR input. Relevant discussion and findings are summarized in Section 6, and final conclusions and future work are presented in Section 7.

2. BACKGROUND AND RELATED WORK

2.1 Tangible Interaction Approaches

With integration of various sensors into a mobile phone, such as a touch screen, camera, accelerometer, compass, and gyroscope, the mobile device itself has gradually become the chief means of interaction between users and 3D virtual objects in mobile AR applications. The user can tangibly manipulate the mobile device to operate the virtual content. For example, the motion sensors in the phone can be used to estimate the movement of the device itself, and provide 6 degree of freedom input for controlling the virtual content. As a result, tangible interaction techniques for mobile AR have been widely used.

Early efforts at handheld mobile AR interaction utilized both the camera motion of the mobile phone and the keypad interface to manipulate graphical objects and create virtual scenes. Henrysson and Billinghurst [4] developed a mesh editing system that used a button on the keypad of a mobile phone to select vertices in a mesh and then use the device motion to move them. Simon's study [12] presented an in-situ 3D modeling interaction method using a video camera of a mobile phone as both an interaction and a tracking device to draw a virtual line while using a key press to select the start and end points of the line. These mobile AR interaction examples combined camera image processing and tangible button input on the mobile phone.

In addition to the camera and the keypad, other powerful embedded sensors on the mobile phone have also been used for mobile AR interaction. For example, Ha and Woo [3] demonstrated a phone-based indirect 3D object manipulation method that used motion sensor information to control 3D virtual objects by touching the screen as well as rotating the phone itself.

More recently, researchers have been looking into the way of freezing the camera frame to stabilize the screen touch input while using camera tracking. For example, Langlotz et al. [7] developed a mobile in-situ content augmentation system using stylus input and camera tracking. Manipulation and authoring of virtual content are done in a view-freeze mode that allowed users to manipulate the virtual content on a still camera image. Guven et al. [2] presented a set of mobile AR interaction techniques to embed multimedia content in the physical environment. Their technique relied on "freezing" the frame for later editing, which allowed users to capture a snapshot (frame) of the environment to work on, and later maps the results back into the physical world. Lee et al. [8] investigated the benefits of using the freezing method in annotation tasks through a user study.

2.2 Intangible Interaction Techniques

The intangible interface accepts a user's off-screen input, like gesture, gaze or even speech, as input signals for manipulating virtual contents in mobile AR environment. Usually a user's hand holds a mobile phone while the other hand is free, so researchers have mainly studied gesture-based interactions using finger or hand tracking.

Regarding the gesture input, an evaluation of several different techniques for 3D object manipulation in a mobile AR environment was conducted by Henrysson et al. [5]. The authors compared marker-based finger tracking input captured by a phone's front camera to tangible input, keypad interaction and phone tilting in increasingly complex translation and rotation tasks. A gesture-based interaction approach on the mobile phone was explored by Hürst and Wezel [13], who used a single color marker-based fingertip tracking to explore alternatives to screen touch interaction in mobile AR. They investigated the potential for finger-based interaction in a further step by extending the single color tracker to two markers, and conducted the evaluation of two finger interaction approaches in a mobile AR board game scenario.

These early methods required the user to have a trackable marker on the fingers. More recently Baldauf et al. [1] developed a visual markerless fingertip detection engine that can be used to manipulate virtual 3D objects for mobile AR applications. This can help users to select one virtual object by pointing at it, or grab, drag and drop it with a similar gesture or a pinch gesture with thumb and index finger. In contrast, Seo et al. [11] proposed a different method that supported the whole hand input for mobile AR interaction. Using a palm pose estimation method, a virtual object can be placed on the palm of the free hand without needing visual markers, and the user can interact with the virtual object by opening or closing their hand.

3. FREEZE VIEW TOUCH INTERACTION

3.1 Motivation and Goal

Our main motivation is to develop more intuitive interaction methods for handheld mobile AR, avoiding the screen shaking that occurs when using current touch-screen input techniques. We first attempt to tackle with this problem by utilizing the "freeze-set-go" approach [8], which offers users an ability to manipulate an AR scene while the viewpoint stands still.

3.2 Interaction Design

Our method works as follows. When a trackable target is identified and tracked correctly, users keep the mobile phone in an appropriate position with a clear view of the virtual object, and then click on the "Freeze" button to stop the AR

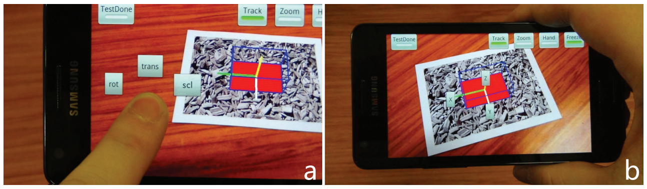

Figure 1: The Marking menu (a) and coordinate buttons (b).

view – the freeze button will not be activated if the visual tracking is failing. Once the freeze button has been pushed, the system will stop the tracking thread immediately to save the computing resources, and users no longer have to point the mobile phone at the target to see the virtual scene.

In the freeze-screen mode, when the user performs a long press on the touch screen, a context menu appears around the pressed point (Figure 1a), implemented in a Marking menu [6] . The menu has three child buttons: Trans (translation), Rot (rotation) and Scl (scaling). We choose this menu style to reduce the user's finger movement and performance time. The user can only have one menu button selected at a time, and once pressing the button the system will be configured in a specific operation mode in which only the selected manipulation has an impact on the virtual object state. The world coordinates will appear in the center of the virtual object to indicate the object's position and pose (Figure 1b). The user can choose which axis he/she wants to operate along by clicking the button at the end of the axis. There will be no influence on the other axis, and multi axis's can be selected at one time.

Any screen touch input will be converted into a change of the virtual object's position or pose. If the user wants to change the manipulation mode or axis, they just need to perform another long press on the screen to call out the main menu and retry the previous steps mentioned above. Specifically, we map the translation and rotation values onto the position change of a single fingertip scrolling on the touch screen, while the scale value is controlled by the distance change between two fingertips pinching on the touch screen. After completing all the manipulations, the user can click the ”Freeze” button again to unfreeze the AR scene, remapping the edited virtual objects into the tracking image. The main benefit of this approach is that it supports precise editing of virtual content.

Our traditional touch interface for handheld AR has the similar operation design except working in the live AR scene instead of the freeze view.

3.3 Implementation Details

Our mobile AR system consists of a tracking module to identify the target and an interaction module to convert the screen touch to virtual object rendering parameters. In addition, the system offers a basic user interface (UI) for human-computer interaction.

We implemented the mobile AR system with freeze view touch interface on a Samsung Galaxy S2 (1.2GHz Dual-Core, 1GB RAM, Android2.3 OS). We split the implementation into two pieces: computer vision tracking, and interaction. The AR tracking was coded in C++ with Android NDK[1], which speeds up the performance of image processing significantly, while the touch interaction and UI layout are built in Java with Android SDK[2], which provides convenient and powerful UI functions for touch gestures. The two parts share the state variables of the virtual object during the application lifecycle.

The system was implemented on top of a natural feature based tracking library we have developed. To build this, we ported the BRISK algorithm [10] to Android using OpenCV4Android[3], and optimized performance by using FastCV[4] functions to implement several time consuming calculation steps. Finally, we combined the detection and description parts with optical-flow tracking to create a full-featured robust AR tracking system. Our tracking software can run smoothly and robustly on the target phone at an interactive frame rate of around 15 to 20 fps with 800*480 pixel camera resolution.

4. FINGER GESTURE INTERACTION

4.1 Motivation and Goal

Instead of physically touching the screen of mobile devices, natural gesture interaction can be an alternative input method for handheld AR. Midair gestures can be captured by the mobile camera and analyzed by computer vision algorithms to indirectly interact with an AR scene. In this section we present a handheld AR interaction method using midair finger gesture input.

4.2 Interaction Design

The finger gesture based interaction uses a similar procedure as the freeze view touch method to begin operation. The user makes a long press on the touch screen and selects an input mode from three options, and then chooses one axis to start the real-time interaction. The only difference between gesture and touch interaction is the way that they acquire the valid input points. Unlike the touch interaction, the gesture interaction gets the input point information from the fingertips detected in the camera frame.

After the AR tracking starts, the user can click on the ”Hand” button on the screen to enable the natural gesture interaction mode. From here on, if the detected hand area is bigger than a threshold value, the hand region inside the camera image will be segmented from the background. The most prominent fingertip will be identified and marked by a small white circle in the ”Translation” and ”Rotation” modes, while a second fingertip will also be recognized and marked by a small red circle in the ”Scaling” mode. The centers of these circles on both fingertips provide the input points for AR interaction. In addition, a big white circle will initially be drawn at the same time to highlight the main input fingertip (Figure 2). This is to indicate whether the real-time gesture input is activated or not. If the big circle appears, it means that all the gesture inputs will be ignored and will have no practical influence on the virtual object's state. On the other hand, when the big circle disappears, it indicates that the user can now make practical operations on the virtual objects with their finger gestures. This approach is

[1] `http://developer.android.com/tools/sdk/ndk/`

[2] `http://developer.android.com/sdk/`

[3] `http://code.opencv.org/projects/opencv/wiki/OpenCV4Android/`

[4] `https://developer.qualcomm.com/mobile-development/mobile-technologies/computer-vision-fastcv`

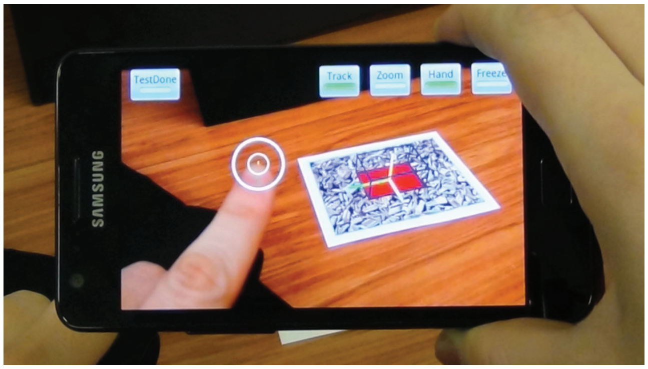

Figure 2: The chief fingertip with two circles.

based on the idea that there is a ”virtual tangible surface” in the air, and the big circle demonstrates whether the fingertip is touching this ”surface” or not.

We change between locked and unlocked states for the main fingertip using a countdown timer: keeping the chief fingertip relatively still in a tiny region (10*10 pixels in our case) around the current position longer than a certain time (2.5 seconds in our case), and the state of the chief fingertip will be reversed, causing the appearance or disappearance of the big circle. While waiting for the locking state change of the fingertip, there is a small green line growing from the center to the edge of the small circle to visually indicate the waiting process.

The gesture interaction has the same implementation of translation, rotation and scaling input as the freeze view touch method, but it is based on fingertip tracking instead. The user can choose to scroll or pinch their fingers in midair in front of the mobile camera for natural input.

4.3 Implementation

We built the natural gesture based interaction method based on the same target mobile platform, tracking system, and UI framework mentioned in Section 3.3, and integrated our finger tracking system (C++ in Android NDK) into the previous system.

We used a skin color detector working in HSV color space to detect possible skin color pixels and segment the hand region from background images, and then removed image noise by using the distance transform. The fingertips were identified based on the curvature-based contour point sampling and elliptical fitting method used in Handy AR [9]. The skin color detection is easily affected by the change of the environment light, so our evaluation performance was tested under stable light condition.

We only consider the condition that the hand is placed in front of the user's face, and during the gesture interaction, the finger is always visible. This assumption is feasible in most handheld mobile AR situations. The final prototype can offer a sufficient interactive performance at 12 to 15 fps.

5. USER STUDY

In order to investigate the usability of our proposed freeze view touch and finger gesture based interaction techniques, we conducted a formal user study comparing these two methods with each other, and with a traditional free view touch approach.

5.1 Experimental Setup and Procedure

We set up a within-group study with one independent variable: the type of interaction method. Ten participants at the university were recruited for the experiment (6 males and 4 females, 6 users at ages 21-30, 3 at ages 31-35, 1 at ages 41-45) (M = 28.7, SD = 5.7). All of them used touch screen devices frequently, and six of them had some experience of using three-dimensional interfaces, mainly from the game consoles like Microsoft Kinect or Nintendo Wii. However, all of them have no previous experience with using AR interfaces. All participants were right handed. During the experiment tests, eight participants held the device in their right hand and used the left hand for interaction, while for the other two it was the other way around. All of them used the index finger for input. No significant differences could be observed regarding handedness.

Each participant was instructed to perform several experimental tasks using the three interaction interfaces (traditional view touch, freeze view touch, and finger gesture based interaction) respectively. Interfaces were presented in different order to the participants to exclude potential learning effects. Three types of manipulations (translation, rotation, and scaling) were included in tasks for each interface. The experimental task was to manipulate a virtual cube in a mobile AR application and match it to the indicated target pose and position. For all tasks, both the indicated target and manipulated cube were clearly displayed on the same screen (Figure 3). Subjects could decide themselves when to start the task by clicking the ”Test Start” button and when to end the task by clicking the ”Test Done” button. The timing of the task was started after the mode and axis selection were done.

In the translation task, subjects had to move a square on the target XY plane to four target positions in different directions with various distances (Figure 3a). In the rotation task, subjects had to rotate the square around the Z-axis clockwise at angles of 25, 120, 225, 330 degrees between the target and initial square pose (Figure 3b). In the scaling task, subjects had to change the size of the square to 0.4, 0.8 times smaller and 1.6, 2.3 times larger at the same proportion (Figure 3c). The square would be reset to initial pose and position (Figure 3d) after each sub-test was completed. Subjects were told to perform the task as fast and accurate as possible. The order of tasks and sub-tests was randomized for each participant to avoid any order-related influences on the results. Before doing each experimental

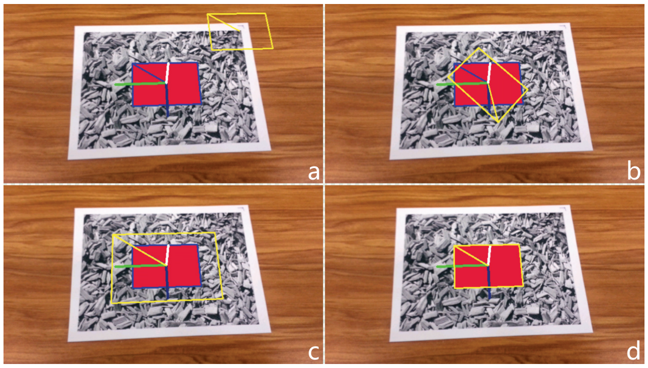

Figure 3: Screenshots of experimental task samples.

trial, participants were given an introduction to the interaction method and they were able to practice the technique. After completing the task by using each interaction method, participants completed a usability questionnaire and gave further comments at the end of the evaluation.

For the evaluation, we configured the system to automatically measure the performance of participants in terms of task completion time and operation error. Furthermore, we evaluated the preference of participants with nine questions (Table 1) related to user performance. We used a nine point Likert-scale (1 to 9 with 1 indicating strongly disagree while 9 indicating strongly agree) for each subjective questionnaire item.

5.2 Experimental Result

To evaluate the performance records and the results of the user questionnaire, we performed the Friedman test with alpha level of 0.05 to verify the significance of difference among the three interaction approaches.

Analyzing the data from the performance measurements (Table 2), we found a significant difference among the interaction methods in terms of task completion time (p <0.001). Participants took more than twice the amount of time to finish all tasks with the gesture based interaction (M = 22.8 sec., SD = 8.8, SE = 0.8) compared to the freeze view touch based interaction (M = 6.4 sec., SD = 2.2, SE =0.2) and the free view touch based interaction (M = 8.7 sec., SD = 3.5, SE = 0.32). Errors from translation, rotation and scaling tasks all showed a significant difference among those interfaces (p = 0.002, 0.0001, and 0.005, respectively). The freeze view touch method produced significantly fewer errors than the other two interaction methods especially for rotation and translation.

In questionnaire terms of easiness of using the interaction method, usefulness of using the method for task completion, intuitiveness, mental stress, and physical stress, as well as fun and engagement, the Friedman test showed that there were some statistically significant differences (p <0.04 for all) between interaction methods. Compared with the traditional free view touch, the freeze view touch received higher ratings, while the finger gesture based interaction obtained lower rating in first five aspects of usability questionnaires mentioned above. This discovery is consistent with the result of the post experiment questionnaire for overall rank of method preference for handheld mobile AR interfaces. Table 1 presents the detailed p value for each question.

Table 1: Friedman Test Result

Question	p
Q1. I was performing well	0.211
The given interface was:	
Q2. Easy to learn	0.174
Q3. Easy to use	0.029
Q4. Useful to complete the task	0.025
Q5. Intuitive	0.035
Q6. Natural	0.223
Q7. NOT mentally stressful	0.037
Q8. NOT physically stressful	0.024
Q9. With fun and engagement	0.009

* N = 10, df = 2.

In contrast, in terms of fun and engagement, the finger gesture based interaction was considered as the most interesting method, while the freeze view touch was ranked last. However, there was no significance found in ratings on the performance confidence, easiness of learning the interaction methods, naturalness. Figure 4 indicates the detailed results of each item.

6. DISCUSSION

Although the freeze view touch performed fastest and most accurately, users felt that its static AR view considerably reduced the real-time engagement and so was not attractive enough for them.

The gesture-based interaction appeared to have less usability compared to the other two touch-based interaction methods. This seems to be partly due to the immatureness of our gesture recognition software, as well as the extra time consumed on locking and unlocking the target's editable state, which was included in the task completion time. If a target is located far away from the initial position in translation tests, users may repeat the unlock-manipulate-lock cycle multiple times. In this case, the gesture-based interaction method may require more completion time than the touch-based interaction technique, which may significantly affect the results. If taking these time out of the consideration, the completion time of our gesture-based interaction method would be reduced approximately 22.0% on average.

However, it is interesting that three interaction methods did not show significant differences in terms of performance confidence, easiness of learning the interaction method, and

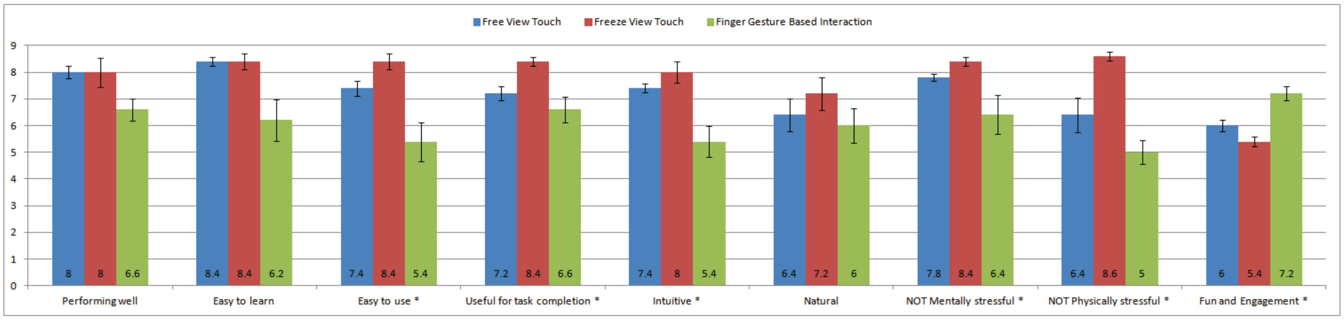

Figure 4: Usability questions and results: Performing well, Easy to Learn, Easy to Use, Useful, Intuitive, Natural, NOT Mentally Stressful, NOT Physically Stressful, Fun and Engagement (Error bar: +/- SE). The items with significant differences are indicated by using an asterisk.

Table 2: Completion Time and Operation Error of Each Task

	Time			Error		
Task	Free touch	Freeze touch	Finger gesture	Free touch	Freeze touch	Finger gesture
Translate	8.0(1.9/0.3)	6.3(0.8/0.1)	24.9(9.7/1.5)	8.2(4.5/0.7)	5.3(2.9/0.5)	14.8(11.0/1.7)
Rotate	11.0(4.1/0.6)	7.5(3.2/0.5)	22.0(12.1/1.9)	1.7(1.2/0.2)	0.8(0.8/0.1)	6.9(7.2/1.1)
Scale	7.1(2.9/0.5)	5.3(1.3/0.2)	21.4(11.3/1.8)	10.7(6.0/1.0)	12.9(7.7/1.2)	20.1(7.2/1.1)

* Mean(SD/SE). Time values in second; Error values in pixel for translate and scale, degree for rotate.

naturalness. This gives an indication that the gesture-based interaction is as easy to learn as traditional methods, and it has potential in terms of naturalness by comparing it to how we interact with the real world.

The post experiment questionnaires indicate that interaction based on touching is considered to be much easier to use due to users feeling less physical stress. In comparison, users felt that the gesture based interaction was more physically stressful since it requires the participant to move a lot in a large 3D space instead of a small 2D surface, which could lead to fatigue over time.

7. CONCLUSION AND FUTURE WORK

In this paper, we investigated two different handheld interaction methods; freeze view touch and finger gesture based interaction. We described how each of these methods is implemented and presented results from a user study comparing these two techniques to the traditional free view touch method. The user evaluation showed that the freeze view touch method seems beneficial compared to the traditional touch input, but more work is required on the gesture based interaction method before it is as accurate enough, even though users found the gesture input method very enjoyable.

In the future we will focus on improving the natural gesture recognition and interaction usability. More complicated gestures, such as select, grab, drag, and drop, will be studied and more detailed user studies will be conducted.

8. REFERENCES

[1] M. Baldauf, S. Zambanini, P. Fröhlich, and P. Reichl. Markerless visual fingertip detection for natural mobile device interaction. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 539–544, Stockholm, Sweden, September 2011.

[2] S. Guven, S. Feiner, and O. Oda. Mobile Augmented Reality interaction techniques for authoring situated media on-site. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '06, pages 235–236, Santa Barbara, CA, USA, October 2006.

[3] T. Ha and W. Woo. ARWand: Phone-based 3D object manipulation in Augmented Reality environment. In *Proceedings of the 2011 International Symposium on Ubiquitous Virtual Reality*, ISUVR '11, pages 44–47, Jeju-si, Republic of Korea, July 2011.

[4] A. Henrysson and M. Billinghurst. Using a mobile phone for 6 DOF mesh editing. In *Proceedings of the 8th ACM SIGCHI New Zealand Chapter's International Conference on Computer-human Interaction: design centered HCI*, CHINZ '07, pages 9–16, Hamilton, New Zealand, July 2007.

[5] A. Henrysson, J. Marshall, and M. Billinghurst. Experiments in 3D interaction for mobile phone AR. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, GRAPHITE '07, pages 187–194, Perth, Australia, December 2007.

[6] G. Kurtenbach and W. Buxton. User learning and performance with marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: celebrating interdependence*, CHI '94, pages 258–264, Boston, MA, USA, April 1994.

[7] T. Langlotz, S. Mooslechner, S. Zollmann, C. Degendorfer, G. Reitmayr, and D. Schmalstieg. Sketching up the world: in-situ authoring for mobile Augmented Reality. *Personal and Ubiquitous Computing*, pages 1–8, 2011.

[8] G. A. Lee, U. Yang, Y. Kim, D. Jo, K.-H. Kim, J. H. Kim, and J. S. Choi. Freeze-set-go interaction method for handheld mobile Augmented Reality environments. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, VRST '09, pages 143–146, Kyoto, Japan, November 2009.

[9] T. Lee and T. Höllerer. Handy AR: markerless inspection of Augmented Reality objects using fingertip tracking. In *Proceedings of the 11th IEEE International Symposium on Wearable Computers*, ISWC '07, pages 83–90, Boston, MA, USA, October 2007.

[10] S. Leutenegger, M. Chli, and R. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *Proceedings of the 13th IEEE International Conference on Computer Vision*, ICCV '11, pages 2548–2555, Barcelona, Spain, November 2011.

[11] B.-K. Seo, J. Choi, J.-H. Han, H. Park, and J.-I. Park. One-handed interaction with augmented virtual objects on mobile devices. In *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, VRCAI '08, pages 8:1–8:6, Singapore, December 2008.

[12] G. Simon. In-situ 3D sketching using a video camera as an interaction and tracking device. In *Proceedings of the 31st Annual Conference of the European Association for Computer Graphics*, EGS '10, Norrköping, Sweden, May 2010.

[13] H. Wolfgang and v. W. Casper. Gesture-based interaction via finger tracking for mobile Augmented Reality. *Multimedia Tools and Applications*, pages 1–26, 2012.