

Experiences in Using a Smartphone as a Virtual Reality Interaction Device



Joon Hao Chuah¹ and Benjamin Lok²

Virtual Experiences Research Group, University of Florida¹
Virtual Experiences Research Group, University of Florida²

Abstract— Input devices such as Nintendo Wiimotes are often used to select and manipulate virtual objects. While simple to use and easily available, these devices have some limitations. When used with common large displays such as televisions, they support only indirect manipulation. These devices also require the user to learn and remember which buttons map to which functions. We propose overcoming these limitations by using a smartphone as an interaction device. Smartphones, like Wiimotes, are readily available and easy to operate. Unlike the Wiimote, the smartphone has a touchscreen that can display the selected object, allowing the user to directly manipulate the object. Further, the touchscreen can customize the interface and provide buttons with clearly labeled functions specific to the object. We report on the lessons learned in integrating and using a smartphone as the interaction device for two applications. The first is a mixed reality game focused on general object selection and pose manipulation. We used this game in a pilot study evaluating usability. The second is an adaptation of an existing virtual reality application. This application demonstrated the ease of adaptation as well as improvements from using a smartphone.

Index Terms—Interaction device, mixed reality, object manipulation, smartphone

I. INTRODUCTION

Virtual reality (VR) interactions make use of a wide variety of input devices, including joysticks, data gloves, and instrumented objects. In recent years, devices such as the Nintendo Wiimote and PlayStation Move have grown in popularity as input devices because of their ease of availability and high adaptability [1][2]. Wiimotes and Moves are often combined with commonly available displays such as monitors, televisions, and projectors. The input device is used to select and manipulate virtual objects. Virtual objects are rendered on screen, and the user indirectly manipulates the virtual object's pose (position and orientation) using the input device. Indirect manipulation places two burdens on the user. One, the user must map changes in the device's pose in real space to changes in the virtual object's pose in virtual space. Two, the user must learn and remember which buttons on the input device map to which application and object functions.

This work was supported in part by a University of Florida Alumni Fellowship and a grant from the National Science Foundation (IIS-0803652).

We propose alleviating these burdens by using a smartphone as an input device for selecting and manipulating objects. Smartphones are more than just input devices; they contain a display and are interaction devices, a combination of an input and output device. The display allows the smartphone to render the selected object in the user's hand instead of on the primary display (e.g., television). By placing the virtual object in the user's real hand, we merge the real and virtual spaces. This merging is especially important for mixed reality (MR) applications. Merging also benefits VR applications that require the user to accurately move objects in 3D space. The virtual object's position in 3D space maps exactly to the phone's position in real space. This reduces the user's mental load. Additionally, the smartphone's touchscreen can present an interface customized to the application and even to the object. Buttons can be clearly labeled with their functions. This customization bypasses much of the need for training and on-screen reminders of button mappings. At the same time, smartphones are readily available and familiar to most users.

Because smartphones are multi-purpose devices, not 3D input devices like the Wiimote, using them as VR interaction devices requires special considerations. In particular, attention must be paid to the form factor and affordances of a touchscreen. In this paper, we report on the lessons learned in integrating and using a smartphone as a VR interaction device so others will be able to successfully integrate a smartphone in their own applications. We used a smartphone as an interaction device in two applications. The first was a mixed reality (MR) game in which users selected objects and manipulated them to place them on a game board. We used this game in a pilot study evaluating the smartphone's usability as an interaction device. The second was an adaptation of an existing VR eye exam used to train medical students [3]. We replaced a Wiimote as an input device with a smartphone. This demonstrated the ease of adapting an existing application and improvements from using a smartphone.

II. RELATED WORK

2.1 Virtual Multi-Tools

Kotranza et al. [3] introduced the idea of virtual multi-tools, a variety of tools controlled by a single input device. They implemented a virtual eye exam where multiple exam tools were controlled by a tracked Wiimote. One button cycled between tools, and other buttons mapped to functions specific to the tool such as turning a light on and off. Using a general purpose input device in this fashion avoided the need to instrument several real objects with tracking devices and sensors. Our approach similarly consolidates controls for several objects into one input device. We improve this idea by leveraging the smartphone's touchscreen to adapt the interface to the specific object. The adaptive interface removes the need to memorize mappings between buttons and functions.

2.2 Smartphones as Interaction Devices

Watsen et al. [4] used a Palm Pilot for 2D interaction tasks within a 3D virtual environment. Two-dimensional tasks include selecting menu items, clicking buttons, and other common GUI tasks. They used a flexible interface that adapted to match the current task. We also use a flexible interface that leverages standard GUI elements, but we use the smartphone for 3D interaction tasks such as pose manipulation.

Henrysson et al. [5] used a phone in a handheld AR application. The phone served as both the view into the AR environment (there were no other displays) and a tool for selecting and manipulating the virtual objects. They explored a variety of methods for manipulating the objects. The most similar method to ours was a ray-casting selection technique. The selected object moved with the phone but remained attached to the ray at a fixed distance away. Similar techniques were also used in a later study where the phone was used to select and edit vertices of a 3D mesh [6].

We expand upon this work by making use of a touchscreen and adding manipulations other than pose. Capacitive touchscreen smartphones were not common when Henrysson et al. performed their work [5][6], and touchscreens afford new interaction styles. In particular, touchscreens afford rotating virtual objects by touching the object on the phone's screen and sliding the finger around. One problem Henrysson et al. noted was rotating virtual objects required large physical movements when the object was attached to the phone at a fixed distance away. We also use the touchscreen to display buttons for object-specific manipulations other than pose such as turning the object on and off.

2.3 Mixed Reality Using Embedded Displays

In many VR interactions, the user only interacts with small portions of the environment. For example, if the user is talking to an avatar in an office, only the avatar is interactive and dynamic. The rest of the environment, such as the office furniture, remains static. In such cases, the virtual environment can be replaced by an equivalent real environment (e.g. the office) augmented with displays in key places (e.g. where the avatar is standing). The environment then becomes a mixed

reality (MR) environment. This approach was used by Pair et al. [7] and Raskar et al. [8]. Using a smartphone as an interaction device expands the interactive area of the MR environment. The smartphone adds a handheld display that can be moved around the environment. This display can be used to render objects the user can hold and use to interact with the environment. Object interactions enrich the experience and expand the potential applications for this style of MR environment.

III. MIXED REALITY GAME

We created a game to evaluate the smartphone's usability in an MR interaction. The game involved placing objects on colored spaces on a game board while working with a virtual human partner. We describe below the MR environment, a pilot study evaluating usability, details of how we integrated the smartphone into our MR system, and lessons learned from the pilot study.

3.1 Environment

The MR system displayed virtual elements on two displays embedded within the real environment (Fig. 1). One embedded display was a 22" LCD monitor placed horizontally on a table. This monitor displayed the game board, a C# application that displayed colored spaces and images of the objects.

The other embedded display was a 40" LCD TV mounted in portrait orientation to an office chair. This TV displayed the virtual human partner from the waist up. A pair of pants stuffed with pillow filling was placed in the chair. This combination provided the appearance of a virtual human sitting in a chair.

The remainder of the MR environment was physical. Two nearby physical shelves contained the physical game pieces the participants selected and manipulated. The participant sat in a physical chair placed on one side of the game board.



Fig. 1. Participant using the smartphone to select an object. The objects and most of the room are real. Only the game board and upper body of the virtual human are virtual.

3.2 Pilot Study

We used this game in a pilot study ($n = 17$) evaluating the smartphone's usability as an interaction device. We describe below the procedure, smartphone interface, and results.

3.2.1 Procedure

The MR game had two phases, a solo phase and a partner phase. The solo phase focused solely on object selection and position manipulation. In this phase, a series of audio prompts directed users to select objects and place them on colored game board spaces. Participants performed a practice sequence to become comfortable with the interface and prompts. Participants then performed three sequences of selecting objects and placing them on spaces. Excluding practice, each participant performed a total of 18 selections.

During this phase, we collected timing and error rate data. Time began when the audio prompt started and ended when the participant had selected an object or placed the object on a destination. The error rate was the percentage of selection attempts (presses of the "Pickup Object" button) that failed to select an object. A selection attempt would fail if the crosshair did not intersect an object.

After this phase, we asked participants to fill out a survey evaluating object selection and position manipulation. The survey contained free-response questions asking if they encountered any problems or had any suggestions. It also contained an item asking participants to rank their agreement with the statements "I feel that I performed quickly when selecting objects" and "I feel that I performed quickly when placing objects on colored spaces." This statement used a 7-point Likert-type scale ranging from "Strongly Disagree" to "Strongly Agree."

The partner phase added orientation manipulation and the need to work with a virtual human partner. In this phase, the game board briefly displayed one object in each of the four colored spaces. The participant and virtual human partner worked together to memorize which objects were in which spaces and how they were oriented. The virtual human was, unbeknownst to the participants, remotely controlled by an operator observing the interaction through a video camera. After the partner phase, we asked participants to fill out a survey asking if they encountered any problems or had any suggestions.

3.2.2 Interface

Participants used a Samsung Galaxy S smartphone running Android 2.1. This phone had a 4-inch capacitive touchscreen display running at a resolution of 480x800 and a 5-megapixel rear-facing camera.

Rather than manipulating the real objects directly, participants selected objects using a ray-casting technique and manipulated virtual copies of the objects. Ray-casting techniques in standard VR environments typically provide feedback using a visible light ray. We replaced the visible light ray in our MR version with a crosshair overlaid on live images from the phone's camera. Participants lined up the crosshair on an object, and the phone displayed the words "Object Found" (Fig. 2). The participant then tapped the "Pickup Object" button on the screen to select the object. We used this game in a pilot study (n = 17) evaluating the smartphone's usability as an

interaction device. We describe below the procedure, smartphone interface, and results. When the participant selected an object, the phone displayed an image (virtual copy) of the selected object (Fig. 3). The phone's tracked position was mapped directly to the virtual copy's position. Participants returned objects to the shelf by tapping the "Place Object Back on Shelf" button. To place objects on the game board, participants held the phone over a colored space on the game board and tapped the "Place Object on Game Board" button.

Rotations were performed in 90 degree increments around the horizontal, forward, and up axes. Rotating the object rotated or swapped the image appropriately. To rotate about the current horizontal or forward axes, the participant swiped a finger across the screen. To rotate clockwise around the current up axis, the participant tapped a button on the touchscreen.

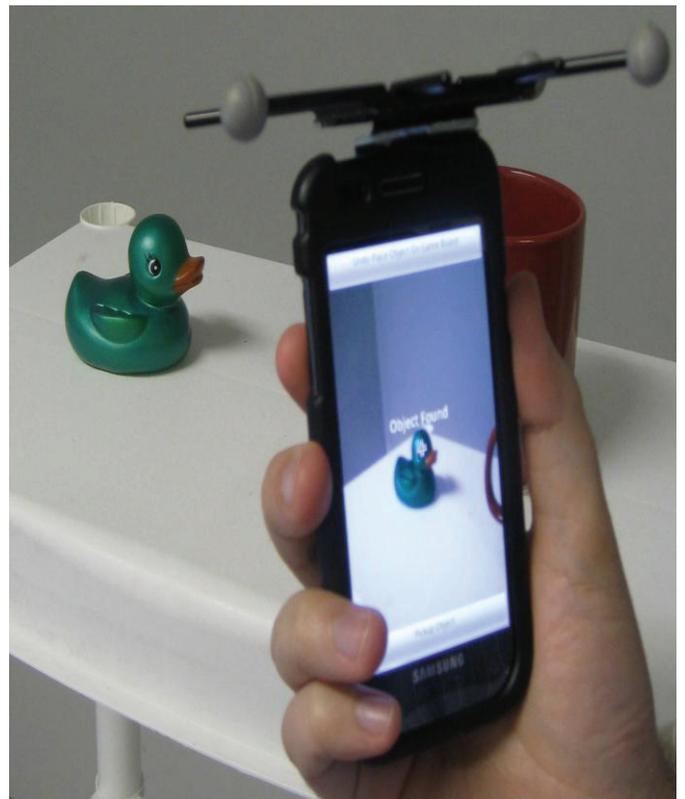


Fig. 2. Interface for selecting objects using a ray-casting technique. The words "Object Found" provide feedback that the crosshair is over an object.

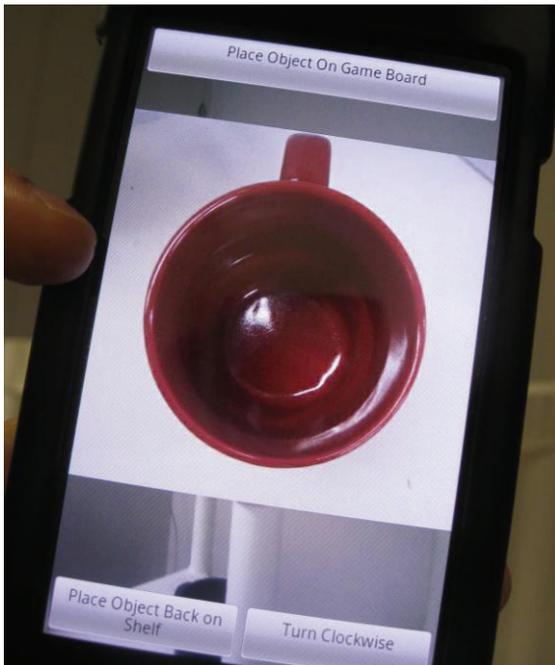


Fig. 3. Interface for manipulating objects. The buttons are clearly labeled with their functions. These functions are specific to object manipulation and distinct from the selection interface.

3.2.3 Results and Discussion

In terms of time, selecting an object took on average 5.17 ± 2.18 seconds. Positioning an object on a space took on average 5.16 ± 1.25 seconds. For the “I felt like I performed quickly” survey item, 100% of the participants responded “Slightly Agree” or better for both selecting and positioning objects. Seventy-five percent of those were “Agree” or better.

The selection time is higher than other ray-casting selection technique implementations. In an experiment focused on ray-casting selection techniques, Wingrave and Bowman measured average selection times around one second [9]. However, the survey results show participants did not feel hampered by the smartphone as an interaction device. Additionally, selection and positioning are low-level components of many applications, not the high-level goal. For example, during the partner phase of the study, selection and positioning were secondary to the memorization task and discussions with the partner dominated the total task time.

The average error rate was $6.16 \pm 17.54\%$. The error rate is relatively low, which suggests the phone works well as an interaction device for ray-casting selection techniques. The high standard deviation can be attributed to the form factor of the phone, which will be discussed in Section 3.4.3.

3.3 Implementation Details

Integrating a smartphone into a VR/MR system required solving two main problems: tracking the phone's pose (position and orientation) and communicating between the phone and the other components.

3.3.1 Tracking

Smartphone-based optical tracking is an advancing field. Henrysson et al. [5][6] used a custom port of ARToolKit for tracking. Klein demonstrated smartphone-based markerless tracking using the phone's camera [10]. However, the accuracy and robustness were reduced compared to desktop-based optical tracking. This performance loss was caused by the low refresh rate and narrow-angle lens of the phone's camera.

Current smartphone-based optical tracking systems did not have the necessary tracking volume, accuracy, refresh rate, and low latency for a ray-casting selection technique to work well. As a result, we used a NaturalPoint OptiTrack system with four ceiling-mounted infrared cameras that tracked a fiducial attached to the phone.

3.3.2 Communication

We created a C# bridge between the phone and the other components of our system. The bridge and other components communicated using VRPN, a standard VR communications protocol. Other components sent commands (e.g., display an image) to the bridge, which relayed them to the phone. The phone sent input events (e.g. button pressed) to the bridge, which broadcasted them to the other components. The bridge and phone communicated over an 802.11g network using a TCP/IP socket and a custom message protocol. Messages consisted of three parts:

- Code - A 32-bit integer indicating what the message is, e.g., a rotation input event. These codes were implemented as identical enums in both the C# bridge code and Android code.
- Payload length - A 32-bit integer indicating the length of the payload in bytes, possibly zero.
- (Optional) Payload - A string containing JSON serialized data, e.g., the rotation direction.

The phone did not demonstrate an unacceptable power draw despite sending or receiving at least 20 messages a second. We were able to run the pilot study for several hours a day while only charging the phone during a lunch break and overnight.

3.4 Lessons Learned

3.4.1 Metaphor

To give users the correct mental model, the correct metaphor must be chosen.

In an initial test group (before the pilot study) we used a camera metaphor for selecting objects. We told participants to select an object by taking a picture of it. This metaphor caused participants to believe the phone was doing image recognition. As a result, some participants tried to get a close up photo of the object. In doing so, they stood up and walked over to the object. This caused them to either leave the tracking volume or occlude one of the tracking cameras. As a result, the ray-casting selection technique repeatedly failed, causing much frustration.

In the pilot study, we switched to a targeting metaphor. We told participants to select an object by centering it on the screen.

This gave participants the correct mental model of aiming from a distance. To reinforce this, we included crosshairs and displayed the words “Object Found” when the ray-casting selection would succeed. This feedback is similar to displaying “Target Locke” or similar in many games. With this feedback and mental model, no participants stood up from the chair.

3.4.2 Direct Orientation Manipulation

The combination of a touchscreen with a tracked device affords two equally important means of directly manipulating orientation, both of which need to be accommodated.

The first means is manipulating the virtual object on the phone's screen. Most users are accustomed to the multi-touch gestures on current smartphones. Swiping on a smartphone's home screen typically moves the page of icons and widgets with the user's finger. However, in our implementation the image did not move with the user's finger and only changed after the gesture was finished. Similarly, users expected the object to rotate with their finger as they swiped left, right, up, and down. A two-finger rotate gesture, common in image manipulation programs, could have replaced the "Turn clockwise" button.

The second means is directly manipulating the orientation of the phone. A few participants expected rotating the phone would also rotate the virtual object. We chose not to implement this because rotating the phone could turn the screen away from the user or place the buttons in awkward positions. However, mapping the phone's orientation directly to the object's orientation is more natural. Direct mapping is also more practical in applications that require only limited rotations rather than full 360 degree rotations about all axes.

3.4.3 Form Factor

The unsteadiness of operating a phone with one hand should be compensated for.

If a participant operated the phone with only one hand, tapping the “Pickup Object” could tilt the phone. The ray-casting-based selection would then fail because the ray no longer intersected the object. This problem of button presses disturbing position (known as the “Heisenberg effect”) is present in many input devices [11]. In response, participants all either switched to two-handed operation or were more careful with one-handed operation. However, this placed an additional burden on the user and possibly impacted speed. We could have compensated for the tilt by selecting the most recent object the ray intersected in the half second prior to the button press.

IV. VIRTUAL REALITY EYE EXAM

As an example of adapting an existing real-world application to use a smartphone as the interaction device, we chose Kotranza et al.'s virtual eye exam [3]. This example highlights

the advantages of a smartphone as an interaction device as well as the ease of adapting existing applications. We describe here an overview of the eye exam, the original interface, the smartphone interface, and the ease of adaptation. We plan to run a study formally comparing the interaction devices.

4.1 Eye Exam Overview

Medical students use the virtual eye exam to learn how to recognize the symptoms and properly diagnose cranial nerve damage. Cranial nerve damage results in symptoms including double vision, the inability to move an eye past a certain point, and retinal damage. The symptoms are assessed with tests including examining the patient's eyes and asking the patient to follow instructions. Instructions included reading a line on an eye chart, counting the number of fingers held up, and following a finger with their eyes. These tests are performed in the virtual eye exam using a set of three tools, an ophthalmoscope, an eye chart, and a virtual hand.

It is important to note that the purpose of the virtual eye exam is to teach which tests to perform and how to interpret the results. The exam is not meant to teach motor skills level use of the tools. As a result, the input device for the tools does not need to perfectly match the real tool.

4.2 Original Interface

The original interface used a television as the output device and a Wiimote as the input device. The Wiimote was tracked by a combination of a NaturalPoint OptiTrack camera and the Wiimote's own tracking capabilities. Tracking allowed the user one-to-one 6-DOF pose control of a virtual tool rendered on the television.

The Wiimote's buttons were mapped to specific functions. The A button cycled between the three tools. Each tool contained its own specific mappings as well. For the ophthalmoscope, the B button turned the light on and off. With the virtual hand, the up and down buttons controlled the number of fingers held up. Finally, the eye chart mapped the Wiimote's position on the y-axis to the selected line number.

4.3 Smartphone Interface

In adapting the eye exam to use a smartphone as an interaction device, we sought to improve on the original interface in two ways:

- Reducing the need for training and memorizing button mappings for each tool.
- More seamlessly combine the real and virtual worlds.

To reduce the need for training and memorizing button mappings, we leveraged the smartphone's touchscreen. The touchscreen displayed a row of buttons with icons for each tool (Fig. 4). These icons immediately inform a new user of the available tools. When the user selected a tool, the touchscreen displayed a large image of the tool in the center. This image reflected the current state of the tool, such as the selected line number on the eye chart. The touchscreen also adapted the interface to present buttons specific to the tool. The buttons are

labeled with their functions, eliminating the need to memorize button mappings. In order to reduce confusion, we mapped all tool-specific functions to two buttons on the bottom of the screen. These buttons only changed their labels; they did not change size or position.

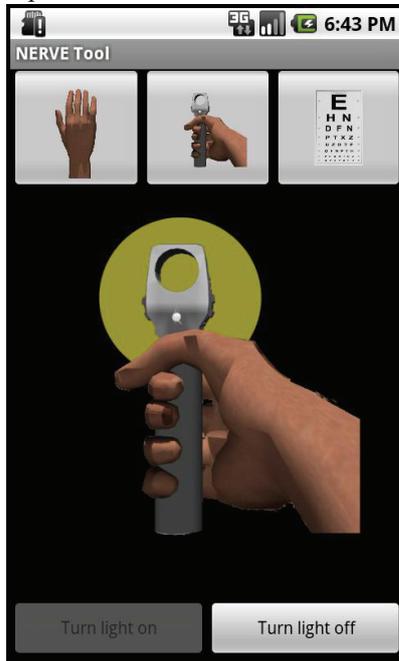


Fig. 4. Closeup of the smartphone eye exam interface.

The real and virtual worlds are more seamlessly combined by having a secondary display (the phone) directly in the user's hand. This display allows the user to directly manipulate the tool rendered on the phone rather than indirectly manipulate a virtual tool rendered on the primary display (e.g. television). Direct manipulation is especially important when the user asks the patient to follow the user's (virtual) finger (Fig. 5). The user must be able to correctly judge the finger's position in 3D space to determine if the patient's eye is following the tool correctly. When both the finger and the patient are rendered on the primary display, as with the original interface, this can be difficult.



Fig. 5. User interact with a life-size virtual human. The virtual human can follow the smartphone with its eyes.

4.4 Ease of Adaptation

Several characteristics of smartphones made it possible to quickly integrate it into an existing VR application. Smartphones have significant on-board computational capabilities and can run custom applications. These applications can also communicate using standard wireless protocols such as 802.11b/g and Bluetooth. The applications can also utilize the phone's touchscreen to display graphics and provide intuitive touch interfaces customized to the VR interaction's needs. Because of all these characteristics, we were able to create and integrate a smartphone interface in less than eight hours using a team of only two programmers.

V. CONCLUSION AND FUTURE WORK

Our pilot study with the MR game demonstrated the smartphone's viability as an interaction device, especially in MR environments using real environments augmented with displays in key places. The pilot study also taught us important lessons regarding direct orientation manipulation, selection metaphors, and form factor.

The virtual reality eye exam showed the ease of adapting an existing application to use a smartphone as the interaction device. Additionally, the smartphone can reduce the burden on the user in terms of training, memorization, and blending of the real and virtual worlds. We plan to replace the OptiTrack system with a system based on a Microsoft Kinect. This system will combine position data from the Kinect with orientation data from the phone. We will use this system in a study comparing our smartphone-controlled version of the eye exam to the Wiimote-controlled version.

Smartphones as interaction devices can improve a variety of VR and MR applications. In particular, computer supported collaborative work applications could benefit from giving each user a flexible interaction device with a private display. Virtual-human-based training applications could be enhanced by incorporating more non-verbal interactions using objects. We hope our experiences in using a smartphone as an interaction device and adapting an existing application will aide others in creating new VR applications or enhancing interfaces to existing VR applications.

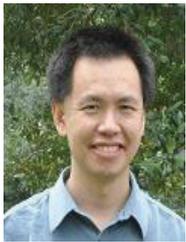
ACKNOWLEDGEMENT

The authors wish to thank Diego Rivera-Gutierrez for his assistance in implementing both applications.

REFERENCES

- [1] J. C. Lee. Hacking the Nintendo Wii remote. *IEEE Pervasive Computing*, vol. 7, no. 3, pp. 39-45, July 2008.
- [2] C. A. Wingrave, B. Willamson, P. D. Varcholik, J. Rose, A. Miller, E. Charbonneau, J. Bott, and J. LaViola. The Wiimote and beyond: spatially convenient devices for 3D user interfaces. *IEEE Computer Graphics and Applications*, vol. 30, no. 2, pp. 71-85, March 2010.
- [3] A. Kotranza, K. Johnsen, J. Cendan, B. Miller, D. S. Lind, and B. Lok. Virtual multi-tools for hand and tool-based interaction with life-size

- virtual human agents. In *2009 IEEE Symposium on 3D User Interfaces*, pp. 23-30.
- [4] K. Watsen and R. Darken. A handheld computer as an interaction device to a virtual environment. In *Third International Immersive Projection Technology Workshop*, Stuttgart, Germany, 1999.
- [5] A. Henrysson, M. Billinghurst, and M. Ollila. Virtual object manipulation using a mobile phone. In *Proceedings of the 2005 international conference on augmented tele-existence*, pp. 164-171.
- [6] A. Henrysson and M. Billinghurst. Using a mobile phone for 6DOF mesh editing. In *Proceedings of the 7th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: design centered HCI*, pp. 9-16.
- [7] J. Pair, U. Neumann, D. Piepol, and B. Swartout. FlatWorld: combining Hollywood set-design techniques with VR. *IEEE Computer Graphics and Applications*, vol. 23, no. 1, pp. 12-15, Jan. 2003.
- [8] R. Raskar, G. Welch, M. Cutts, and A. Lake. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the 25th annual conference on computer graphics and interactive techniques*, pp. 179-188., 1998.
- [9] C. Wingrave and D. Bowman. Baseline factors for raycasting selection. In *HCI International*, 2005.
- [10] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *Proceedings of the 8th IEEE International Symposium on Mixed and Augmented Reality*, pp. 83-86, Oct. 2009.
- [11] D. A. Bowman, C. Wingrave and J. Campbell. Using Pinch Gloves™ for both natural and abstract interaction techniques in virtual environments. In *HCI International*, pp. 629-633, 2001.



Benjamin Lok is an Associate Professor in the Computer and Information Sciences and Engineering Department at the University of Florida an co-founder of Shadow Health, Inc. an educational software company. He is also an Adjunct Associate Professor in the Surgery Department at the Georgia's Health Sciences University. His research focuses on virtual humans and mixed reality in the areas of

computer graphics, virtual environments, and human-computer interaction. Professor Lok received a Ph.D in Computer Science from the University of North Carolina at Chapel Hill in 2002.



Joon Hao Chuah is currently a PhD candidate in the Department of Computer and Information Science and Engineering at the University of Florida. He received a BS in computer science from the University of Texas at Austin in 2006 and an MS computer engineering from the University of Florida in 2012. His research interests include virtual humans, mixed reality, and human-computer interaction